Gesellschaft für Informatik (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into the fields of
- Seminar
- Proceedings
- Dissertations
- Thematics

current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English

Information: http://www.gi-ev.de/service/publikationen/lni/

ISBN 978-3-88579-422-6

This volume contains papers from the symposium "Informatik in der DDR - Grundlagen und Anwendungen" held in Dresden May, 2008. This conference is the sequel to the symposia in Chemnitz, 2004 and in Erfurt, 2006 that are also issued in this series. The overall aim of the conference was to provide a scientific survey of the history of the computer science in the former German Democratic Republic. The topics covered in the papers range from research in software-related fields, computer science education, data bases, computer applications, to papers that present the development of hardware platforms including operating systems.

Birgit Demuth (Hrsg.): Informatik in der DDR 2008, Dresden

3

GI-Edition

Lecture Notes in Informatics

Birgit Demuth (Hrsg.)

Informatik in der DDR - Grundlagen und Anwendungen

Tagungsband zum Symposium 15. und 16. Mai 2008 in Dresden

Thematics



Birgit Demuth (Hrsg.)

Informatik in der DDR
Grundlagen und Anwendungen

Drittes Symposium "Informatik in der DDR"

15. und 16. Mai 2008
in Dresden, Deutschland

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Thematics
Series of the Gesellschaft für Informatik (GI)

Volume T-3

ISBN 978-3-88579-422-6

Volume Editors
Dr.-Ing. Birgit Demuth
Softwaretechnologie, Institut für Software- und Multimediatechnik,
Fakultät Informatik, Technische Universität Dresden
01062 Dresden, Germany
Email: birgit.demuth@tu-dresden.de

Series Editorial Board
Heinrich C. Mayr, Universität Klagenfurt, Austria (Chairman, mayr@ifit.uni-klu.ac.at)
Jörg Becker, Universität Münster, Germany
Hinrich Bonin, Leuphana-Universität Lüneburg, Germany
Dieter Fellner, Technische Universität Darmstadt, Germany
Ulrich Flegel, SAP Research, Germany
Johann-Christoph Freytag, Humboldt-Universität Berlin, Germany
Ulrich Furbach, Universität Koblenz, Germany
Michael Koch, TU München, Germany
Axel Lehmann, Universität der Bundeswehr München, Germany
Peter Liggesmeyer, TU Kaiserslautern und Fraunhofer IESE, Germany
Ernst W. Mayr, Technische Universität München, Germany
Heinrich Müller, Universität Dortmund, Germany
Sigrid Schubert, Universität Siegen, Germany
Martin Warnke, Leuphana-Universität Lüneburg, Germany

Dissertations
Dorothea Wagner, Universität Karlsruhe, Germany
Seminars
Reinhard Wilhelm, Universität des Saarlandes, Germany
Thematics
Andreas Oberweis, Universität Karlsruhe (TH)

© Gesellschaft für Informatik, Bonn 2008
printed by Köllen Druck+Verlag GmbH, Bonn

Freie Forschung in freiwilliger Achtung der Würde des Menschen
Zum Geleit der ”Informatik in der DDR” 2008

Uwe Aßmann

Lehrstuhl Softwaretechnologie
Fakultät Informatik
Technische Universität Dresden
uwe.assmann@tu-dresden.de
http://st.inf.tu-dresden.de

Die Würde des Menschen ist unantastbar. Sie zu achten und zu schützen ist Verpflichtung aller staatlichen Gewalt.

Art. 1(1) Grundgesetz der Bundesrepublik Deutschland (GG)

Forschung und Lehre sind frei.

Art. 5(3) GG

Die Umstände mögen sich ändern, aber die gleichen Fehler werden von den Nachgeborenen mit Gründlichkeit wiederholt. Wer sie vermeiden möchte, wird Konrad Zuses Lebenserinnerungen ein zweites Mal lesen müssen. Das scheint mir ein Lob zu sein, das man nur selten aussprechen darf.

Prof. H. Zemanek, Wien, im Vorwort zu [Zus08]

1 Erinnerung, Bildung und Freiheit

Dietrich Schwanitz schreibt in seinem Bestseller über Bildung [Sch99], dass die deutsche Bildungskrise hauptsächlich dadurch verursacht ist, dass man den Wert des Wissens über die Geschichte unterschätzt, folglich den Geschichtsunterricht und die Geschichtsbildung vergisst und dann dazu verurteilt ist, alle Fehler der Vergangenheit zu wiederholen. Viele Weise haben versucht, der Menschheit dieses Prinzip zu vermitteln, dennoch begeht sie diesen unseligen Fehler immer wieder (in Bezug auf die Informatik formuliert diese Wahrheit Heinz Zemanek, der Schöpfer des Mailüfterls, im Vorwort der Autobiographie Zuses [Zus08]). Die Absicht, diesen Fehler zu vermeiden, liefert allein schon genügend Motivation, eine Tagung wie die “Informatik in der DDR” (IDDR) zu veranstalten, um aus der Rückschau heraus den Blick in die Zukunft richten zu können. Natürlich sollte ein solcher Rückblick umfassend und vollständig sein, sodass auf dieser, nun dritten Tagung

in einer Reihe, neben der Geschichte der Technologie des Computers auch die Geschichte der Softwaretechnologie eine große Rolle spielen soll.

Neben der Kenntnis der Geschichte als wichtiges Element für die Bildung liegt ein zweiter Grund, um zurückzuschauen, in der Würde des Wissenschaftlers als Mensch. Gleichgültig, unter welchem Regime ein Wissenschaftler gedient hat, hat er das Recht, in würdiger Art und Weise auf sein Lebenswerk zurückzuschauen bzw. es so ausführlich, korrekt und fair zu dokumentieren, damit andere zurückschauen können. Also dient eine Konferenz wie die IDDR dazu, die Leistungen der Menschen würdig darzustellen, die, vielleicht auch gerade unter schwierigen Bedingungen, anständig gearbeitet und Wichtiges für Gesellschaft und Industrie geleistet haben. Daher ist es geradezu eine demokratische Pflicht, eine solche Konferenz zu veranstalten, und es ist Ausdruck unserer Freiheit, auch gerade der Freiheit unserer Forschung, dies tun zu können.

Allerdings sollte man in der Rückschau die Geschichte möglichst neutral darstellen, damit auch wirklich etwas gelernt werden kann. Es geht ja nicht nur um die eigene Würde im Rückblick, sondern auch um die Würde derjenigen Menschen, über die wir reden und erzählen. Daher spielt natürlich der Umgang mit der Spannung zwischen Wissenschaft und Politik eine wichtige Rolle. Wir dürfen nicht vergessen, dass jede wissenschaftliche Tätigkeit in ein politisches und gesellschaftliches Umfeld, das den Wissenschaftler und seine Forschung beeinflusst, eingebettet war und natürlich immer noch ist. Ja, noch mehr, wir dürfen nicht vergessen, wer forschen durfte, wer nicht forschen durfte, und warum er das nicht durfte. Ich denke, etliche Papiere, die auf dieser Tagung präsentiert werden, tragen diesem Anliegen Rechnung, weil sie jenseits der Beschäftigung mit der reinen Technologie auch diesen Aspekt nicht vergessen. Sicher heißt es, nach vorne zu schauen, zu versöhnen statt zu spalten; aber dieses Ansinnen wird nur mit einer fairen und vollständigen Darstellung der Geschichte gelingen. Vergebung und Versöhnung benötigen Ehrlichkeit, und Ehrlichkeit braucht Mut. Und vielleicht lässt sich ja in der Rückschau auch manches unverkrampft diskutieren, was damals nur hinter verschlossenen Türen und vorgehaltener Hand besprochen werden konnte. Wir wollen auch nicht vergessen, dass unser heutiger Rechtsstaat, unsere demokratische Verfassung und unsere freie Wissenschaft ein Resultat vieler Kämpfe sind, die andere Menschen für uns ausgefochten haben. Nicht zuletzt die friedliche Revolution von 1989 ist ja gerade durch den aufrechten Einsatz vieler Bürger der damaligen DDR ermöglicht worden. Gerade dieser Opfer wegen bleibt es unsere Aufgabe für die Zukunft, den Rechtsstaat und die Demokratie zu verteidigen. Die hohen Güter der Menschenwürde, der Forschungsfreiheit, der Meinungsfreiheit und anderer Grundrechte, die ja gerade diese Konferenz ermöglichen, kosten etwas - zumindest Ehrlichkeit und Mut, den ich mir persönlich und Ihnen allen wünsche.

Nun sind wir als Informatiker keine Historiker, sondern veranstalten eine solche Konferenz aus der Sicht der Augenzeugen. Da ich zu jung bin, um aus dieser Perspektive etwas beitragen zu können, aber trotzdem eine Vorliebe für Geschichte pflege, gestatten Sie mir, einen kleinen amateurhistorischen Beitrag zum historischen Hintergrund zu geben, vor dem die beiden o.a. Artikel über die Menschenwürde und die freie Forschung entstanden sind.

2 Freiheit und Menschenwürde

Würde des Menschen und Grundrechte, insbesondere die Freiheit der Forschung, hängen zusammen. Das Grundgesetz betont die Abhängigkeit beider Grundsätze, indem es sie zusammen in den Grundrechtsartikeln verankert; es sagt aber überraschenderweise sonst nichts weiter dazu. Die deutsche Geschichte des 20. Jahrhunderts zeigt an vielen Stellen, dass beide Grundsätze nicht zu trennen sind: Sowohl im ersten Weltkrieg und der Weimarer Republik, im Dritten Reich als auch in den beiden deutschen Staaten nach dem großen Krieg haben sich entweder beide Grundrechte zusammen verwirklichen lassen, oder sie sind beide zusammen missachtet worden. Insbesondere Wissenschaftler geraten, so scheint es, leicht in diese Spannung zwischen Sein oder Nicht-Sein der Freiheit und der Menschenwürde. Wie aber kann eine Gesellschaft beide Prinzipien realisieren? Wie sollen wir diese Grundsätze leben, nachdem im 20. Jahrhundert so vieles misslang?

Ein Rückblick auf die Entstehungsgeschichte der Grundrechtsartikel zeigt, dass das Verhältnis der beiden Grundsätze sehr wohl diskutiert wurde. Zwar wurde der Begriff der Menschenwürde, auf den Vorschlag Theodor Heuss hin [Wet05, S. 8-9], bewusst nicht definiert und seine Bedeutung dem Leser des Grundgesetzes offen gelassen, doch verdeutlicht schon alleine die gewählte Stellung der Artikel, dass man die Grundrechte aus der Würde des Menschen abgeleitet sieht [Wet05, S. 8-9]. Die Existenz der Würde des Menschen schafft sein Recht, insbesondere seine Grundrechte in Staat und Gesellschaft. Staat und Gesellschaft sind um des Menschen willen, der Mensch nicht um des Staates willen. Der Staat hat die Grundrechte zu achten, weil der Mensch eine Würde hat. Achten Staat und Gesellschaft die Grundrechte des Einzelnen nicht, sind die Grundrechte nicht einklagbar, wird in Wirklichkeit die Würde des Menschen angetastet - und das wird durch Artikel 1 zum Tabu erklärt. Wo aber entstand diese Idee, Würde und Freiheit des Menschen mit einander als untrennbar verknüpft zu betrachten?

Historisch gesehen sind die Grundrechtsartikel des Grundgesetzes stark von den Erfahrungen des nationalsozialistischen Regimes und dem zweiten Weltkrieg geprägt. Der Begriff der Menschenwürde entstand nicht aus dem Nichts, sondern bewegte viele Überlebende des Nationalsozialismus, weil sie Argumente gegen seine menschenverachtende Lehre suchten. Was aber lässt sich erfolgreich ins Feld führen? Es kann nicht nur einfach eine oberflächliche politische Ideologie sein, man benötigt eine solide Ethik, eine grundlegende Motivation für die Menschenrechte. Und da lag die Idee nahe, dem Kern des Nationalsozialismus, der Menschenverachtung, als Grundprinzip die Achtung der Würde des Menschen entgegen zu stellen [Wik08, Wet05]. Bereits 1943 entwarf der Kreisauer Kreis unter Helmuth James Graf von Moltke einen Vorschlag für eine Verfassung, der einen wesentlichen Passus zur Menschenwürde enthielt [Wet05, S. 8-9]:

> Die Anerkennung der unverletzlichen Würde der menschlichen Person als Grundlage der zu erstrebenden Rechts- und Friedensordnung wird gefordert.

Auch die Richtlinien für eine deutsche Staatsverfassung der Sozialdemokraten im Exil (1944 London) enthalten einen ähnlichen Passus:

> Die Achtung und der Schutz der Freiheit und der Würde der Persönlichkeit

> sind die unveräußerlichen Grundlagen des staatlichen und gesellschaftlichen Lebens der deutschen Republik.

Als 1946 die ersten Landesverfassungen ausgearbeitet wurden, spielte der Begriff der Menschenwürde eine wesentliche Rolle. Für die Bayerische Verfassung wirkte der Staatsrechtler Prof. Hans Nawiasky, der, von den Nazis verfolgt, in die Schweiz emigriert war, als wissenschaftlicher Berater. Auf Nawiaskys Betreiben hin wurde die Idee der Menschenwürde in die Bayerische Verfassung vom 8.12.1946 aufgenommen, sowohl in die Präambel, also auch in die Definition der Grundrechte [Ver46a]:

> (Präambel) Angesichts des Trümmerfeldes, zu dem eine Staats- und Gesellschaftsordnung ohne Gott, ohne Gewissen und ohne Achtung vor der Würde des Menschen die Überlebenden des zweiten Weltkrieges geführt hat, in dem festen Entschlusse, den kommenden deutschen Geschlechtern die Segnungen des Friedens, der Menschlichkeit und des Rechts dauernd zu sichern, gibt sich das bayerische Volk nachstehende demokratische Verfassung.
>
> Art. 100. Die Würde der menschlichen Persönlichkeit ist in Gesetzgebung, Verwaltung und Rechtspflege zu achten.
>
> Art. 108. Die Kunst, die Wissenschaft und ihre Lehre sind frei.

Man sieht, dass bereits hier die Stellung der Artikel ähnlich zum Grundgesetz erfolgt und somit das Ansinnen deutlich macht, dass die Menschenwürde die Grundlage auch der freiheitlichen Rechte, insbesondere auch der Forschungsfreiheit, ist. Auch die Hessische Verfassung vom 1.12.1946 enthält eine entsprechende Formulierung [Ver46b]:

> Art. 3. Leben und Gesundheit, Ehre und Würde des Menschen sind unantastbar.

Die Formulierungen des Grundgesetzes wurden dann entscheidend vom Herrenchiemseer Verfassungskonvent geprägt (10.-23.8. 1948) [Chi48]. An ihm nahm wieder Hans Nawiasky teil und brachte erneut den Begriff der Menschenwürde ein. Dieses Mal erachtete man ihn als derart wichtig, dass er am Beginn des Verfassungsentwurfes platziert wurde:

> Artikel 1. (1) Der Staat ist um des Menschen willen da, nicht der Mensch um des Staates willen.
>
> (2) Die Würde der menschlichen Persönlichkeit ist unantastbar. Die öffentliche Gewalt ist in allen ihren Erscheinungsformen verpflichtet, die Menschenwürde zu achten und zu schützen.

Auf diesem Entwurf baute dann der sog. Parlamentarische Rat auf, der vom 1. September 1948 an tagte und weitere Einzelheiten festlegte. Hier haben einige andere Persönlichkeiten noch eine Rolle gespielt, wie Carlo Schmid oder Theodor Heuss, aber auch Konrad Adenauer, der den Vorsitz führte, und auf dessen Rolle ich im Folgenden noch näher eingehen möchte.

2.1 Konrad Adenauers Rhöndorfer Programm

Eine weitere Wurzel der Grundrechtsartikel liegt in der Persönlichkeit und den Anschauungen Konrad Adenauers [Sch94]. Adenauer, in der Weimarer Republik bereits Oberbürgermeister von Köln, war unter dem nationalsozialistischen Regime kaltgestellt und pflegte in seinem Haus in Rhöndorf seine Rosen. Als der Krieg zu Ende ging - er war immerhin 69 Jahre alt - stieg er dennoch in die Parteiarbeit der jungen CDU ein. Anfang 1946 wurde er ihr Vorsitzender in der britischen Zone und spielte von da an eine führende Rolle in der Vorbereitung des neuen Staates im Westen, der etwa ab 1948, nach dem Beginn des kalten Krieges, immer konkreter geplant wurde.

Es heißt in einer führenden Adenauer-Biographie [Sch94, S. 496], dass Adenauer eine wichtige vorläufige Formulierung für Art. 1 GG über die Würde des Menschen selbst vorweggenommen hat, und zwar in seinem *Rhöndorfer Programm*, einem Programmentwurf für die CDU von Ende 1945. Von diesem Programmentwurf existiert ein handschriftlicher Aufschrieb, der wie folgt beginnt:

1. Einzelperson und Staat

   (a) Die Grundsätze christlicher Ethik und Kultur, wahrer Demokratie, müssen das staatliche Leben tragen und erfüllen. An der Würde und den unveräußerlichen Rechten der Person findet die Macht des Staates ihre Grenzen.

   (b) Recht auf politische und religiöse Freiheit.

   (c) Gerechtigkeit, gleiches Recht und Rechtssicherheit für jeden.

   (d) Anerkennung der grundlegenden Bedeutung der Familie für Volk und Staat.

   (e) Anerkennung und Schutz der Frau bei ihrer Tätigkeit in Haus und Familie. Betätigung der Frau im beruflichen und öffentlichen Leben.

   (f) Die Mehrheit hat kein willkürliches und uneingeschränktes Recht gegenüber der Minderheit. Auch die Minderheit hat Rechte und Pflichten.

Man sieht, dass aus der Formulierung des Punktes 1.a ebenfalls Artikel 1 GG hervor scheint. Weitere Punkte, zum Beispiel 1.b-d, finden sich ebenso in den Grundrechtsartikeln wieder. Man kann also festhalten, dass Adenauer vom Rhöndorfer Programm an in 1945 bis zur Definition des Grundgesetzes in 1949 an diesen Grundsätzen festgehalten und als Vorsitzender des Parlamentarischen Rates bei ihrer Durchsetzung eine wesentliche Rolle gespielt hat, was sich natürlich gut mit dem Anliegen von Prof. Nowiasky und vieler anderer traf.

2.2 Die Wurzel im Humanismus

Historisch hat die These, dass die Würde des Menschen die Grundrechte hervorruft, mindestens zwei wesentliche Wurzeln, die humanistische und die christliche (wenn man einmal die sozialdemokratische unter der humanistischen subsumiert). Beides lässt sich gut

am Beispiel Adenauer zurückverfolgen. Adenauer war stark von dem Schweizer Sozialethiker, Politiker, und Professor für internationales Recht Carl Hilty geprägt (1833-1909) [Sch94, Seite 109ff]. Es wird berichtet, dass Adenauer Hilty intensiv gelesen hat, weil in seinem Schlafzimmer Bücher Hiltys standen, in denen er viele handschriftliche Anmerkungen gemacht hat [Hil91, Hil99]. Das folgende Zitat aus [Hil46, S. 73-75] stellt sehr gut die humanistische Wurzel des Zusammenhangs zwischen Achtung der Menschenwürde und gesellschaftlicher Freiheit dar. Laut Hilty wird vom Einzelnen gefordert, sich *freiwillig* unter sittliche Grundsätze unterzuordnen, um so eine *freiheitliche* Grundlage von Demokratie und Rechtsstaat zu finden und gleichzeitig die Würde des Menschen zu wahren (Hervorhebungen aus dem Original):

> Die Welt muss in jeder Richtung *durch Freiheit* zur Vollendung gelangen, nicht durch Zwang und Gewalt irgend einer Art. Der *freiwillige* Gehorsam jedes Einzelnen gegenüber der sittlichen Weltordnung ist der Zweck und das Ziel der Weltgeschichte. ... Die freie sittliche Persönlichkeit aber findet ihr volles Wachstum und Gedeihen nur auf dem Boden der *politischen*, religiösen, individuellen und wirtschaftlichen Freiheit, und sie entwickelt sich mangelhaft, wenn auch nur *eine* dieser Freiheiten außer dem Bereich ihres Verständnisses sich befindet.

Dabei ist das Argument Hiltys in merkwürdiger Weise zirkulär: Einerseits muss die Menschenwürde geachtet werden, um Freiheit in Staat und Gesellschaft zu erlangen. Andererseits ist die erlangte Freiheit nicht grenzenlos, sondern findet ihre Grenze in der Beachtung der Würde des Menschen. Und drittens ist diese Entscheidung zur Achtung der Würde des Menschen freiwillig, also eine Ausübung von individueller Freiheit, denn erst im freiwilligen sittlichen Umgang mit dem Nächsten, in der freiwilligen Beachtung seiner Würde, findet der Mensch die gesellschaftliche Freiheit.

2.3 Die Wurzel im christlichen Menschenbild

Die zweite Wurzel des Zusammenhangs der Grundrechte und der Achtung der Menschenwürde liegt im christlichen Menschenbild. Viele der Verfassungsväter waren von dort her geprägt, und es wurde kontrovers diskutiert, einen Bezug zum Gottesbegriff in die Grundrechtsartikel aufzunehmen, was schließlich, bis auf die Formulierung der Präambel *"in der Verantwortung vor Gott"*, unterblieb [Wet05, S. 8-9]. In christlicher Anschauung ist der Mensch als das Bild des allmächtigen Gottes geschaffen und als solcher wertvoll und achtenswürdig. Daraus folgen seine Grundrechte, insbesondere seine Freiheit.

Diese christlich-humanistische These wird auch bei Hilty vertreten. Nach Hilty kann richtig verstandenes Christentum nur humanistisch und freiheitlich sein [Hil46, S. 73-75]:

> Freiwilligkeit, freie Annahme aus Überzeugung, ist der Kern und Sinn des Christentums gegenüber der vorangegangenen, mehr autoritativen und formalen Religionsausübung gewesen. Wir wollen nichts von einem Zwang, auch

> nicht zur Tugend, wissen, sondern die Einsicht und der vernünftige freie Wille der Menschheit zum Guten soll den Staat regieren. ... Dies scheint uns der ursprüngliche Gedanke des Christentums zu sein: die Erziehung aller Menschen zur Freiheit.

Lassen Sie mich also vermuten, dass Adenauers Formulierungen aus dem Rhöndorfer Programm von 1945 ihre Wurzel in der humanistischen Ethik Carl Hiltys haben: in der freiwilligen Unterordnung in eine sittliche Grundordnung, findet der Mensch die einzige Möglichkeit, die Würde des Nächsten und seiner selbst zu achten und seine politische, religiöse und individuelle Freiheit zu leben - und wir können ergänzen, auch seine wissenschaftliche Freiheit.

3 Zum Geleit der Tagung

Der Begriff der *Menschenwürde* ist in Deutschland sprichwörtlich geworden, er bildet unser höchstes Verfassungsprinzip [Wet05], er trägt einen guten Teil unseres Rechtsverständnisses, ja, er bildet auch die Voraussetzung für die Freiheit von Wissenschaft und Forschung. Würde des Menschen und individuelle Freiheit des Forschers gehören zusammen. Beide Blumen können nur zusammen wachsen, gedeihen und blühen. Der Lehrstuhl Softwaretechnologie der Fakultät Informatik der Technischen Universität Dresden, der die Tagung “Informatik in der DDR 2008” ausrichtet, und ich denke, auch Sie, meine Damen und Herren, bekennen sich daher zu folgenden Grundsätzen:

1. Bildung benötigt einen bewussten, neutralen und würdigen Umgang mit der Geschichte, auch der Wissenschaftsgeschichte.
2. Freiheit der Forschung und Würde des Menschen sind untrennbar miteinander verknüpft. Forschung braucht Freiheit, um solche menschenwürdige Forschung sein zu können. Forschung ohne Freiheit wird der Würde von Menschen abträglich sein.
3. Nur auf dem Boden der individuellen Freiheit und der freiwilligen Unterordnung unter sittlich-humanistische Grundsätze können Menschenwürde und Freiheit der Forschung gedeihen.

In diesem Sinne wünsche ich Ihnen eine echte Bildungs-Tagung, getragen von Würde und Freiheit.

Ich danke Frau Dr. Birgit Demuth für die Übernahme der Hauptlast der Organisation vor Ort, Decanus Prof. Dr. Wolfgang E. Nagel für die freundliche Unterstützung, Herrn Prof. Dr. Karl Hantzschmann für die Übernahme des eingeladenen Vortrags, sowie dem Programmkomittee für die Begutachtung der Beiträge.

Literatur

[Chi48] Chiemseer Entwurf eines Grundgesetzes für einen Bund deutscher Länder. http://www.verfassungen.de/de/de49/chiemseerentwurf48.htm, 10.-24. August 1948 1948.

[Hil91] Carl Hilty. *Glück*. Leipzig, 1891.

[Hil99] Carl Hilty. *Was ist Glaube? In: Glück, dritter Teil*. Leipzig, 1899.

[Hil46] Carl Hilty. *Freiheit*. Huber, 1946.

[Sch94] Hans-Peter Schwarz. *Adenauer. Band 1. Der Aufstieg*. dtv, 1994.

[Sch99] Dietrich Schwanitz. *Bildung - alles, was man wissen muß*. Goldmann-Verlag, 1999.

[Ver46a] Verfassung des Freistaates Bayern. http://www.dircost.unito.it/ cs/docs/bayern1946.htm, 2. Dezember 1946.

[Ver46b] Verfassung des Landes Hessen. http://www.hessenrecht.hessen.de/ gesetze/10_1verfassung/10-1-verfass/verfass.htm, 2. Dezember 1946.

[Wet05] Franz J. Wetz. Die Würde des Menschen: antastbar? Heft der niedersächs. Landeszentrale für polit. Bildung, 2005. http://www.politische-bildung.de/niedersachsen/wuerde_menschen.pdf.

[Wik08] Wikipedia. Menschenwürde, Hessische Verfassung, Bayerische Verfassung, Hans Nowiasky, Konrad Adenauer und diverse andere Artikel. http://de.wikipedia.org/wiki, 2008.

[Zus08] Konrad Zuse. *Der Computer, mein Lebenswerk*. Springer-Verlag, Heidelberg, Germany, 2008.

Vorwort

Less visible and not sufficiently noted by the public is the role that software and software development play in this context (Anmerkung des Herausgebers: computer applications). The reasons are obvious: software is almost intangible, generally invisible, complex, vast and difficult to comprehend. Hardly anybody, with the exception of any experts, can truly grasp what software really is and what role it plays.
Manfred Broy auf der sd&m-Tagung „Software Pionieers“, Bonn 2001

Vor Ihnen liegt der zweite Tagungsband zu einer Reihe von Symposien mit dem Titel „Informatik in der DDR“. Die ersten beiden „Informatik in der DDR“-Symposien fanden 2004 in Chemnitz und 2006 in Erfurt statt. Herrn Prof. Friedrich Naumann und Frau Prof. Gabriele Schade gebührt der Verdienst, mit der Gründung dieser Tagungsreihe, der Organisation der Tagungen und als Herausgeber des ersten Tagungsbandes[1], Bilanz zu den wissenschaftlichen und technischen Entwicklungen auf dem Gebiet der Informatik in der DDR zu ziehen und damit das Bewahrenswerte zu dokumentieren. Die gemeinsame Initiative und Arbeit beider Wissenschaftler war eine glückliche Fügung. Herr Naumann war als Professor für Wissenschafts-, Technik- und Hochschulgeschichte in der Lage, die Entwicklungen in der DDR als historische Kategorie exakt zu bewerten. Frau Prof. Schade repräsentierte die Informatiker, die als Akteure und Augenzeugen der DDR-Informatik die überwiegende Mehrheit der Autoren und Teilnehmer der Tagung darstellte.

Das dritte Symposium 2008 in Dresden ist als würdige Fortsetzung der bisher stattgefundenen Tagungen zu verstehen. Dresden hat sich in der sächsischen IT-Industriegeschichte einen Namen sowohl in der Hardware- als auch in der Softwareentwicklung gemacht. Viele dieser Entwicklungen haben bis heute ihre Spuren hinterlassen, insbesondere durch Firmengründungen nach 1990. Nachdem 2004 und 2006 in den Tagungsbeiträgen die Entwicklung der DDR-Rechentechnik im Mittelpunkt des Rückblicks stand, haben sich engagierte Wissenschaftler 2008 im Aufruf zu Beiträgen unter dem Untertitel der Tagung „Grundlagen und Anwendungen“ stärker an die „Softwarepioniere“ der DDR gewandt. Die Resonanz auf den Aufruf zu Beiträgen war erfreulich hoch. Das ist sicher nicht zuletzt das Resultat der erfolgreich stattgefundenen ersten beiden Symposien. Es ist aber zugleich der Wunsch vieler Aktiven der „weichen Seite“ der Informatik gewesen, Forschung und Entwicklung jenseits der Entwicklung der DDR-Rechentechnik darzustellen. Neben Kolleginnen und Kollegen aus Rostock, Berlin, Chemnitz und Ilmenau fühlten sich in Dresden insbesondere ehemalige Vertreter des Kombinates Robotron (Zentrum für Forschung und Technik, Robotron-Projekt Dresden), der Sektionen Informationsverarbeitung sowohl an

[1] Friedrich Naumann, Gabriele Schade (Hrsg.): *Informatik in der DDR - eine Bilanz.* GI-Edition Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn 2006

der Technischen Universität Dresden als auch an der Ingenieurhochschule Dresden, der Pädagogischen Hochschule Dresden und der Medizinischen Akademie „Carl Gustav Carus“ angesprochen, über ihre wissenschaftliche Arbeit in der DDR zu berichten.

Einen breiten Raum nehmen Beiträge zur wissenschaftlichen Arbeit an den klassischen Forschungseinrichtungen der DDR, der Akademie der Wissenschaften, den Universitäten und dem Zentrum für Forschung und Technik Robotron ein. Darin eingeschlossen sind auch Beiträge zur Softwaretechnologie und der Datenbankentwicklung, die im Call for Paper mit eigenen Schwerpunkten gewürdigt worden sind. Wenn auch der Begriff „Software“ in der DDR, wie in verschiedenen Beiträgen immer wieder berichtet wird, offiziell verpönt war, so war Software für die Fachleute und insbesondere die Wissenschaftler ein selbstverständlicher Begriff. Dabei soll nicht verschwiegen werden, dass die DDR auch Software „adoptierte“, wie in dem Beitrag über die PC-Software in der DDR berichtet wird. Die „Datenbankszene“ der DDR errang mit der Entwicklung eigener Systeme, wie z.B. DBS/R, AIDOS und DABA 1600, nicht nur eine große Bedeutung in der Anwendungsforschung, sondern erzielte auch beachtenswerte Erfolge bei den Anwendern. Unter der Überschrift „Anwendungslösungen“ werden interessante IT-Lösungen in der Medizin, dem Autombil- und Maschinenbau vorgestellt. Die Entwicklung der DDR-Rechentechnik findet ihren Platz in Beiträgen zu ESER-Rechnern als auch Kleinrechnern einschließlich der Entwicklung der dazu notwendigen Betriebssysteme.

Ziel der DDR war es stets „den Weltstand mitzubestimmen“. Wenn dies auch im Wesentlichen nicht gelang (über verschiedene Ursachen und kleine Erfolge wird in vielen Beiträgen berichtet), so erkannte die Politik doch die große Rolle der Aus- und Weiterbildung ihrer jungen Leute und ebenso ihrer gestandenen Ingenieure. Das kommt in den Artikeln zu den beiden großen Schwerpunkten der Tagung, Informatikausbildung an Universitäten und Hochschulen als auch Informatikausbildung in der Schule zum Ausdruck. Zum Beispiel ist in der gesamtdeutschen Informatikgeschichtsschreibung kaum bekannt, dass Ende der sechziger Jahre nicht nur in der alten Bundesrepublik Informatikstudiengänge eingerichtet wurden. So kann die Fakultät Informatik der TU Dresden im nächsten Jahr (2009) 40 Jahre Informatikausbildung in Dresden feiern. Ihre Vorläuferin wurde am 1. Januar 1969 als „Sektion Informationsverarbeitung“ an der TU Dresden gegründet.

Informatik als Pflichtunterricht in allen allgemeinbildenden erweiterten Oberschulen wurde zu Beginn des Schuljahres 1988/89 eingeführt. Herrn Prof. Steffen Friedrich ist es zu verdanken, dass dieser Aspekt der DDR-Informatikgeschichte im Tagungsprogramm seine Beachtung findet. Die Traditionen und Erfahrungen in der Informatikausbildung leben heute in den verschiedenen Bildungseinrichtungen sowohl der schulischen Bildung als auch der Hochschulen fort.

Der vorliegende Tagungsband spiegelt die Schwerpunktsetzung in seiner Gliederung wider. Wir haben in Vorbereitung des Dresdner Symposiums ein hochkarätiges Programmkomitee zusammen gerufen. Neben den Gründern der Tagungsreihe (Prof. Friedrich Naumann, Prof. Gabriele Schade) haben darin die Schirmherren Prof. Wilfried Brauer (Vorsitzender des Präsidiumsarbeitskreises „Geschichte der Informatik“ der Gesellschaft Informatik) und Prof. Uwe Aßmann (Professur Softwaretechnologie an der Fakultät Informatik der TU Dresden), ehemalige Führungspersönlichkeiten sowie Professoren mitgewirkt, die die Entwicklungen vor und nach 1990 beurteilen können und heute zum Teil noch im Beruf

aktiv sind. Stellvertretend für alle Mitglieder des Programmkomitees sei Herrn Prof. Gerhard Merkel, Herrn Prof. Ottomar Herrlich, Herrn Prof. Helmut Löffler und Herrn Prof. Reinhold Schönefeld für ihre besonders aktive inhaltliche Mitarbeit gedankt. Einige der Fragen bei der Begutachtung der eingereichten Kurzfassungen von Beiträgen waren

- Ist der Beitrag gegenüber Chemnitz 2004 und Erfurt 2006 neu?
- Ist der Beitrag von historischem Interesse?
- Ist der Beitrag für ein breites Publikum interessant?
- Ist die Betrachtungsweise kritisch?
- Wird etwas zur Nachhaltigkeit der DDR-Arbeiten in der heutigen Zeit gesagt?

Die Langfassung der Tagungsbeiträge liegen in der alleinigen Verantwortung der Autoren. Der Leser sollte bedenken, dass diese Beiträge in der Mehrzahl nicht von Technikhistorikern wie Herrn Prof. Friedrich Naumann und Herrn Detlev Fritsche oder Archivwissenschaftlern wie Frau Buchwald geschrieben worden sind. Die Gefahr der Darstellung der Geschichte durch Wissenschaftler des Faches Informatik und im Abstand von oft mehr als zwei Jahrzehnten liegt darin, dass die Berichte auf Basis von Erinnerungen und damit oftmals nicht ganz objektiv geschrieben sind. Auf der anderen Seite sind es Berichte von Augenzeugen, die sich in ihrer Gesamtheit, wie im Vorwort des ersten Tagungsbandes geschrieben, zu einem wertvollen Ganzen fügen und so den Anspruch erheben können, Bilanz zur Informatik in der DDR zu ziehen. Aus eigener Erfahrung weiß ich, dass sich mit dem größer werdenden zeitlichen Abstand auch die Sicht auf die Dinge verändert. Ich habe inzwischen den wesentlich größeren Teil meines „Forscherlebens“ im geeinten Deutschland verbracht und die neuen Arbeitsbedingungen zu schätzen gelernt. Das wiederum erlaubt bei kritischer Betrachtung der Vergangenheit, wie ich denke, ein realistisches Bild zu vermitteln.

Ich möchte auch nicht vergessen, andere, meist Privatpersonen zu erwähnen, die bei der Aufarbeitung der Geschichte der Informatik in der DDR wertvolle Informationen zusammen getragen haben. An erster Stelle sei der Förderverein der Technischen Sammlungen Dresden (TSD) genannt. Dieser besteht aus ehemaligen Robotron-Mitarbeitern und stellt im WWW[2] ein nützliches Archiv von Fakten und Ereignissen aus der Geschichte des VEB Kombinat Robotron sowie der dort entwickelten Rechentechnik einschließlich der dazugehörigen Software der interessierten Öffentlichkeit zu Verfügung. Der Arbeitsgruppe (AG) Rechentechnik des Fördervereins sei an dieser Stelle für die Unterstützung des Dresdner Symposiums gedankt. Sie hat in Vorbereitung des Symposiums mit viel Liebe zum Detail eine Ausstellung zur Rechentechnik in der DDR in den Räumen der Technischen Sammlungen Dresden vorbereitet und bietet am Begrüßungsabend Führungen dazu an. Weitere Initiativen ranken sich um das Erbe von N.J. Lehmann, der in ganz Deutschland als Pionier der Rechentechnik Anerkennung gefunden hat. Das Dresdner Symposium würdigt das Wirken von N.J.Lehmann mit einem eingeladenen Vortrag von Prof. Karl Hantzschmann, der das vielseitige Wirken von N.J. Lehmann nicht nur auf dem Gebiet

[2] http://robotron.foerderverein-tsd.de/

der Rechentechnik darstellt. Die Fachrichtung Mathematik der TU Dresden als ehemalige Wirkungsstätte von Prof. N.J Lehmann bietet ebenfalls zur Tagung Führungen durch die Sammlung „Historische Rechenmaschinen“ an (Prof. Manfred Ludwig).

Die Vorbereitung des 3. Symposiums zur „Informatik in der DDR“ war ähnlich zu den beiden vorhergenden Veranstaltungen nicht problemlos. Umso mehr bin ich dem Dekan der Fakultät Informatik der TU Dresden, Herrn Prof. Wolfgang E. Nagel, und meinem eigenen Chef, Herrn Prof. Uwe Aßmann, dankbar, die Durchführung der Tagung an der TU Dresden unterstützt zu haben. Die Initiative für ein Dresdner Symposium einschließlich erster Absprachen insbesondere mit dem Förderverein der TSD hatte Herr Prof. Wilfried Krug (DUALIS GmbH IT Solution) ergriffen, dem ich an dieser Stelle ebenfalls herzlich danken möchte. Als Vertreter der Technischen Sammlungen sei Herrn Dr. Helmut Lindner gedankt, der zusammen mit seinen Mitarbeitern und der AG Rechentechnik, vertreten durch Herrn Thomas Falk, für den Erfolg des Begrüßungsabend einschließlich der „anfaßbaren“ Zeugnisse der DDR-Informatik steht. Herr Prof. Friedrich und Frau Prof. Schade haben die diesjährigen Organisatoren mit Tipps und Erfahrungen in der Tagungsvorbereitung unterstützt. Die Hauptlast der Organisation der Tagung tragen in den letzten Monaten vor dem Symposium Frau Silvia Kapplusch und Frau Christiane Wagner. Beiden Kolleginnen möchte ich ganz herzlich für die engagierte und konstruktive Zusammenarbeit danken. Frau Kapplusch kümmert sich um alle Belange der Öffentlichkeitsarbeit wie die Gestaltung der Webseite, das Versenden von Einladungen, die Verwaltung von Anmeldungen und viele andere Tagungsmodalitäten.

Frau Wagner ist in erster Linie als Vertreterin des FFFI (Förderverein Freunde und Förderer der Informatik der TU Dresden e.V.) aktiv und hat in dessen Auftrag die solide Finanzierung der Tagung sichergestellt. Die Bereitstellung der notwendigen finanziellen Mittel konnten wir durch die Unterstützung unser externen Förderer absichern. Dabei seien an erster Stelle Herr Prof. Wilfried Brauer als Vorsitzenden des Präsidiumsarbeitskreises „Geschichte der Informatik“ der Gesellschaft Informatik und Herr Prof. Sigram Schindler von der TELES AG genannt, die zusammen die Herausgabe des Tagungsbandes als GI-Edition „Lecture Notes in Informatics“ finanziell sicherstellten. Weitere finanzielle Unterstützung erhielten wir von den „Datenbankfirmen“ Robotron Datenbank-Software GmbH und der SQL GmbH.

Im Unterschied zum ersten Tagungsband schreibe ich dieses Vorwort zu einem Zeitpunkt, da die Vorbereitungen zur Tagung noch nicht abgeschlossen sind. Deshalb sei an dieser Stelle bereits heute all den Personen gedankt, die die Durchführung des Dresdner Symposiums zu einem Erfolg machen und die ich bisher nicht erwähnt habe. Ich bin zuversichtlich, dass Dresden im Mai 2008 allen Teilnehmern in guter Erinnerung bleiben wird. Nicht zuletzt sorgen dafür auch die vielen interessanten Beiträge der Autoren und die hoffentlich zahlreichen diskussionsfreudigen Teilnehmer des Symposiums. Die Organisatoren der Tagung sind stolz, mit Unterstützung der Arbeit des Programmkomitees und der Autoren, die Herausgabe des Tagungsbandes zum Zeitpunkt der Konferenz zu ermöglichen. In diesem Zusammenhang sei meinem Ehemann Herrn Dr. Frank Demuth, ebenfalls Informatiker, ausdrücklich für seine wertvolle Unterstützung bei der Vorbereitung, insbesondere der aufwendigen Lektorierung des Tagungsbandes, gedankt.

Möge das Dresdner Symposium „Informatik in der DDR“ dazu beitragen, den „weißen Fleck“ auf der Landkarte der gesamtdeutschen Informatikgeschichtsschreibung weiter zu tilgen und zu vielen streitbaren Diskussionen mit dem Blick auf die Zukunft, nicht zuletzt auf die weitere erfolgreiche wirtschaftliche Entwicklung des „Silicon Saxony“, anregen. In diesem Sinne wünsche ich der Tagung einen großen Erfolg!

Dr.-Ing. Birgit Demuth

Technische Universität Dresden
Institut für Software- und Multimediatechnik
Softwaretechnologie

birgit.demuth@tu-dresden.de
http://st.inf.tu-dresden.de

Programmkomitee

Uwe Aßmann
Wilfried Brauer
Steffen Friedrich
Ottomar Herrlich
Klaus Kabitzsch
Helmut Löffler
Gerhard Merkel
Friedrich Naumann
Günter Riedewald
Gabriele Schade
Reinhold Schönefeld
Uwe Wloka

Tagungsleitung und Organisation

Uwe Aßmann
Birgit Demuth
Thomas Falk
Wilfried Krug
Helmut Lindner
Silvia Kapplusch
Christiane Wagner

Förderer

Freunde und Förderer der Informatik der
TU Dresden e.V. (FFFI)



Technische Universität Dresden



Gesellschaft für Informatik e.V.
Präsidiumsarbeitskreis
„Geschichte der Informatik“ der GI



TELES AG Informationstechnologien



Technische Sammlungen Dresden



Robotron Datenbank-Software GmbH



SQL Gesellschaft für
Datenverarbeitung mbH Dresden



VDI-BV Dresden



DUALIS GmbH IT Solution



Inhaltsverzeichnis

Eingeladener Vortrag

Karl Hantzschmann
N. J. Lehmann - sein Wirken für die Informatik in der DDR 25

Forschung I

Gerhard Merkel
Forschungsarbeiten zur Informatik
in der Akademie der Wissenschaften der DDR 29

Horst Reichel
Forschung zu Abstrakten Datentypen im ZFT des Kombinates ROBOTRON 41

Frank Zänker
Künstliche Intelligenz und automatische Sprachverarbeitung -
ausgewählte Forschungen im Kombinat ROBOTRON 48

Forschung II

Helmut Löffler
Rechnernetzforschung und -entwicklung an der TU Dresden 60

Dieter Jungmann
Virtuelle Maschinen für Anwendungssoftware -
Kompatibilität und zukunftsorientierte Lehre 67

Martin Frank
Forschungsarbeiten zur Simulation Diskreter Prozesse und
zu objektorientierter Simulationssoftware an der IHD 80

Forschung III

Joachim Oelschlegel, Horst Kreschnak
Forschungen zum computergestützten Unterricht am Forschungs- und Rechenzentrum der Akademie der Pädagogischen Wissenschaften der DDR (FRZ der APW) und an der TU Dresden aus Sicht einer tätigkeitsorientierten Modellierung 92

Adolf Kotzauer, Karl-Heinz Kutschke
Computergraphik an der Universität Rostock von 1969 bis 1990 104

Günter Riedewald
Von ALGOL 60 zur symbolischen Simulation 117

Ausbildung I

Angela Buchwald
Die Ausbildung von Informatikern in Dresden - frühe Anfänge 129

Erwin Schmidt
Zur Geschichte und Lehre der Informatik an der Technischen Universität Dresden bis zur Gründung des Informatik-Zentrums 1986 142

Thomas Kudraß
Die Studienrichtung Informationsverarbeitung an der TU Dresden - aus studentischer Perspektive 154

Ausbildung II

Winfried Kalfa
Betriebssysteme erfahren - erleben - begreifen - verstehen 164

Hardwin Jungclaussen
Das LEDA-Experiment 176

Mieczysław A. Kłopotek
Strukturiertes Programmieren - strukturelles Denken 185

Ausbildung III

Steffen Friedrich, Bettina Timmermann
Lehrerbildung Informatik - Basis für die Informatik als Allgemeinbildung 197

Holger Rohland, Michael Unger
Gab's gar schon eLearning? 209

Steffi Heinicke, Michael Unger
Das Schülerrechenzentrum Dresden von 1984 bis heute 220

Softwareentwicklung

Hans-Jürgen Lodahl
Das Softwarehaus des VEB Kombinat Robotron 231

Detlev Fritsche
Mit Prototyprekonstruktion zum Welthöchststand?
PC-Software in den letzten Jahren der DDR 243

Reinhold Schönefeld
Von den Anfängen der objektorientierten Softwareentwicklung
an der TU Ilmenau bis zu einem Ansatz für ein
komponentenorientiertes Softwaresystem 255

Datenbanken

Jürgen Bittner
Die Entwicklung von Datenbanksystemen für die Verwaltung
strukturierter Daten im Kombinat Robotron 268

Birgit Demuth, Karl-Heinz Wiggert
Technologietransfer am Beispiel von DABA1600 280

Hanns-Dieter Hartmann, Peter Naumann, Rolf-Dieter Wenzel, Frank Zänker
Das Robotron Informationsrecherchesystem AIDOS:
Software-Technologie von gestern? 292

Anwendungslösungen I

Werner Gumpert
50 Jahre „Computer Aided Engineering (CAE)“ von DDR-Wurzeln bis Airbus 303

Wilfried Krug
Zur Entwicklung der Fertigungsinformatik in der DDR 314

Klaus Fuchs-Kittowski, Hildebrand Kunath
Zur Gestaltung medizinischer Informationssysteme und
zur Entwicklung der medizinischen Systemforschung in der DDR 326

Anwendungslösungen II

Gerhard Bergholz
Forschung zur Leistungsmodellierung von Rechnerinstallationen
in Rechenzentren der DDR 338

Harald Löhr
Die Verarbeitung unformatierter Daten am Beispiel
der Beschreibung von wissenschftlich-technischen Dokumenten 350

Rüdiger Liskowsky, Klaus Panceram
Entwicklung eines verteilten bildschirmorientierten
Buchungs- und Auskunftssystems für die Materialwirtschaft 362

Betriebssysteme und Rechentechnik I

Friedrich Naumann
Adam Rieses Visionen oder: Der lange Weg zum ESER 374

H.-Georg Jungnickel
Zur Kooperation zwischen dem NIZEWT (UdSSR) und Robotron (DDR)
bei der Entwicklung der ESER- Betriebssysteme -
ein historischer und technologischer Rückblick 386

Siegfried Junge
DEKK - das DDR-Konzept zum Zusammenführen von
Büromaschine und Computer und dessen Umsetzung 399

Betriebssysteme und Rechentechnik II

Gert Keller, Gunter Kleinmichel
Entwicklung der Software für Heim-, Klein- und Bildungscomputer als Basis
für die Ausbildung in Schulen und Berufsschulen der DDR 411

Hans-Jürgen Brosch
Über ein „Prinzip der strikten simultanen Rekursion“
beim Entwurf von Rechnerarchitekturen und -arithmetiken 423

Klaus-Dieter Weise
Filmische Retrospektive - Rechentechnik der 1970-er Jahre 435

N. J. Lehmann – sein Wirken für die Informatik in der DDR

Karl Hantzschmann

Abstract: In dem Buch „Die Vergangenheit der Zukunft: Deutsche Computerpioniere“ wird auch N. J. Lehmann zusammen mit einem von Zuse gemalten Portrait vorgestellt. Diese Ehrung basiert dominierend auf seinem Wirken als Konstrukteur und Erbauer der D-Rechenautomaten an der TU Dresden in den 50-er und 60-er Jahren. Das wissenschaftliche Werk von N. J. Lehmann muss aber wesentlich weiter gefasst werden. Das Spektrum seiner Arbeiten reicht von seiner Pionierrolle in der Rechenautomatenkonstruktion über grundlegende Arbeiten im Bereich der Numerischen Mathematik, Arbeiten zu einer sprachlich geführten Programmierungstechnologie bis hin zur Computer-Analytik, als deren Begründer er international angesehen wird.

Das Anliegen dieses Vortrages besteht nicht darin, dieses vielseitige Wirken überblicksmäßig zu würdigen. Insbesondere sollen hier seine in der internationalen Fachwelt hoch angesehenen mathematischen Beiträge auf dem Gebiet der Differential- und Integralgleichungen außen vor bleiben, sondern es geht um die von ihm und seinen Mitarbeitern am ehemaligen Institut für Maschinelle Rechentechnik und dem späteren Wissenschaftsbereich Mathematische Kybernetik und Rechentechnik der TU Dresden erbrachten Leistungen, die einen wesentlichen Beitrag zur Entwicklung der Informatik in der ehemaligen DDR darstellen.

1 „Maschinelle Rechentechnik“ an der TU Dresden als Vorläufer der Informatik

Unter dem Namen „Maschinelle Rechentechnik“ begann in den 50-er Jahren an der TU Dresden im Rahmen der Mathematik eine Entwicklung, die unzweifelhaft einen wichtigen Ausgangspunkt für die sich in den folgenden Jahrzehnten vollziehende Entwicklung einer Wissenschaftsdisziplin Informatik darstellt.

Die am Institut für Maschinelle Rechentechnik entwickelten und gebauten Rechenautomaten D1, D2, D4a bildeten damals eine rechentechnische Basis, auf der an der TU erste Arbeiten zur effizienten Nutzung dieser neuen Technik in den unterschiedlichsten Anwendungsbereichen begonnen werden konnten. Bereits die Geräte D1 und D2 demonstrieren in anschaulicher Weise, wie mit einer gut durchdachten logischen Struktur und einem leistungsstarken Befehlscode bei minimalem technischen Aufwand ein Optimum an Leistungsfähigkeit erreichbar ist. Mit der Konstruktion des Klein-Rechenautomaten D4a verfolgte N. J. Lehmann bereits damals das Ziel, dezentral einsetzbare Kleinrechentechnik bereitzustellen, die sich bequem zum Gebrauch auf dem

Schreibtisch eignet, auf Grund ihres einfachen Aufbaus billig produzierbar ist, aber durch eine ausgefeilte logische Struktur relativ leistungsfähig und nutzerfreundlich handhabbar ist.

Über die Besonderheiten der Dresdner D-Automaten wurde bereits in einem Vortrag auf dem 1. Symposium zur Geschichte der Informatik in der DDR ausführlich berichtet. Die auf dieser Grundlage entwickelten Konzepte zu einer effizienten Maschinenprogrammierung waren wesentliche Beiträge für die spätere Entwicklung der „Programmierung im Kleinen".

Im Rahmen des Studienganges Mathematik wurde bereits Mitte der 50-er Jahre eine Studienrichtung „Maschinelle Rechentechnik" implementiert, die als Vorläufer einer späteren Informatik-Ausbildung angesehen werden kann und damals bildungspolitisches Neuland darstellte.

2 Sprachlich geführte Programmierungstechnologie

Zweckmäßige Hilfsmittel für die Programmierung von Computern lagen N. J. Lehmann permanent besonders am Herzen. In den 70-er Jahren rückten Untersuchungen zur Rolle von Programmiersprachen als Basis einer Softwaretechnologie in den Vordergrund. Sein Konzept zur sachgerechten Beschreibung und automatischen Implementierung von Fachsprachen wurde ein wichtiger Schritt bei der Entwicklung von Hilfsmitteln zur Realisierung von Nutzersichten. Damit konnte dem sich auftuenden Widerspruch zwischen der Herangehensweise des Informatikers und der Betrachtungsweise eines Anwenders begegnet werden.

Die zahlreichen praktischen Anwendungen des unter seiner Leitung entwickelten Fachsprachensystems DEPOT haben den von ihm propagierten Weg glänzend bestätigt. Der Anwender erhält damit die Möglichkeit, in seinem Fachgebiet zu arbeiten und dazu eine seinem Fachgebiet angepasste Fachsprache benutzen zu können. Vom System bereitgestellte metasprachliche Komponenten erlauben die Definition der Fachsprache und die automatische Generierung des benötigten Compilers zur Übersetzung von Fachsprachtexten in eine üblicherweise als Zielsprache genutzte allgemeine Programmiersprache.

Die Dominanz der sprachlichen Führung im Gesamtprozess der Informationsverarbeitung bedingt aber, dass neben der Aufbereitung des jeweiligen Fachgebietes durch Nutzung und gegebenenfalls Entwicklung einer problemorientierten Fachsprache auch die mit einer zugehörigen Systembeschreibungssprache bereitgestellten datenverarbeitungsspezifischen Möglichkeiten der Informationsverarbeitungssysteme ausgenutzt werden. Die Zusammenfassung der Fachsprache eines Problemkreises mit den angemessenen Bearbeitungshilfen des Informationsverarbeitungssystems hat möglichst formalisiert im Rahmen einer so genannten „problemorientierten Programmierungssprache" zu erfolgen. Eine solche problemorientierte Programmierungssprache fasst also sowohl fachgebiets- wie informationsverarbeitungsgemäße Erfahrungen und Erkenntnisse zusammen und ist

Basis und hocheffektives Werkzeug für die sachgerechte Erfassung, Aufbereitung und Lösung der Aufgaben der zugehörigen Problemklasse.

In enger Beziehung zu diesem Problemkreis stehen die intensiven Bemühungen, die anwenderfreundliche Nutzung von Computern durch eine Softwareintegration im Rahmen von Programmpaketen und Programmsystemen zu fördern und auf dieser Basis die weitestgehend automatische Bearbeitung ganzer Aufgabenklassen in einer technologisch einheitlichen Form zu ermöglichen. Das Konzept der „problemorientierten Programmierungssprache für Programmpakete“ (PPS/PP) geht davon aus, dass jedem in einem integrierten Programmsystem enthaltenen relativ abgeschlossenen Programmteil/Programmpaket eine solche Sprache zuzuordnen ist, die allein Zugang zu den Leistungen dieses Programmpaketes ermöglicht. Diese Spezialsprachen machen das im Programmpaket implementierte Fachwissen des jeweiligen Anwendungsgebietes in einfacher und problemorientierter Form zugängig und erschließen zusätzlich in derselben Weise die Leistungen des EDV-Systems und EDV-Fachwissen. Beide Komponenten bilden also in einer PPS/PP wieder eine untrennbare Einheit.

3 Computer-Analytik als Werkzeug für den analytisch arbeitenden Wissenschaftler

Als Vertreter der Angewandten Mathematik erkannte N. J. Lehmann schnell das Potential leistungsfähiger Computer für den vorwiegend formelmäßig arbeitenden Wissenschaftler. Er gilt international als Begründer der so genannten Computer-Analytik, eines an der Nahtstelle zwischen Mathematik und Informatik angesiedelten Wissenschaftszweiges. Die Ziele dieser neuen Nutzungsform von Computerleistungen lassen sich wie folgt abstecken:

- Bestimmung formelmäßiger Näherungslösungen, die die Problemeigenschaften möglichst gut widerspiegeln und eventuelle Parametereinflüsse zum Ausdruck bringen

- Formelausdrücke möglichst einfach und übersichtlich gestalten bei Gewährleistung realer Genauigkeitsansprüche

- Beurteilung der Genauigkeit durch Fehlerabschätzungen, die ohne manuelle Aufbreitung vollständig vom Computer geliefert werden.

Wichtig ist, dass sich diese analytisch orientierte Arbeitsweise in Form symbolischen Rechnens sinnvoll mit der bewährten numerischen Arbeitsweise des Computers verbinden lässt.

Leider war es N. J. Lehmann nicht vergönnt, dass von ihm begonnene Softwareprojekt „Experimentalsystem Computer-Analytik“ zu vollenden.

4 Wissenschaftsorganisatorisches Engagement

Der Blick allein auf die wissenschaftlichen Aktivitäten als wichtige Beiträge zur Entwicklung der Informatik wird dem Wirken von N. J. Lehmann nur unvollkommen gerecht. Zum Abschluss des Vortrages soll noch ein kurzer Blick auf sein vielseitiges Wirken auf der wissenschaftsorganisatorischen und wissenschaftspolitischen Ebene geworfen werden:

- Gründung des Instituts für Maschinelle Rechentechnik an der TU Dresden
- Einrichtung eines Weiterbildungszentrums „Mathematische Kybernetik und Rechentechnik / Informationsverarbeitung“ (MKR/IV)
- Gründung und Leitung eines Akademieinstituts für Maschinelle Rechentechnik
- Verantwortlicher Leiter der Hauptforschungsrichtung „Mathematische Grundlagen der Informationsverarbeitung“ im Rahmen des Gesamtforschungsprogramms der DDR
- Initiierung von Lehrprogrammen für die EDV-Grundausbildung
- Vertretung der DDR in der Internationalen Förderation für Informationsverarbeitung (IFIP, von 1980 bis 1983 deren Vizepräsident)
- Engagement für die wissenschaftliche Integration der Akademien der östlichen Nachbarländer (KNWWT)
- Mitgliedschaft in der Sächsischen Akademie der Wissenschaften und der Akademie der Wissenschaften der DDR
- Mitglied des Forschungsrates.

In zahlreichen nationalen und als Repräsentant der DDR in internationalen Gremien hat er an vielen Entscheidungen zur Herausbildung und Entwicklung der Informatik in Forschung und Lehre mitgewirkt und mit Engagement die junge Wissenschaft Informatik in der ehemaligen DDR ganz wesentlich befördert.

Forschungsarbeiten zur Informatik in der Akademie der Wissenschaften der DDR

Gerhard Merkel

Am Sonnenhang 88
01744 Dippoldiswalde
merkel.gerhard@t-online.de

Abstract: Die Geschichte und das Wirken der Akademie der Wissenschaften der DDR (AdW) sowie deren Umfeld werden grob skizziert und auch die Einordnung der Informatik-Forschung innerhalb der AdW. Die Informatik-Forschungsarbeiten des Zentralinstituts für Kybernetik und Informationsprozesse sowie des Instituts für Informatik und Rechentechnik werden skizziert und über die Auflösung der Institute und der Gelehrtengesellschaft der AdW wird berichtet.

1 Zur Akademie der Wissenschaften der DDR (AdW)

Der Begriff Akademie geht auf eine von Platon 374 v. u. Z. gegründete Schule zurück. Er wird doppelt genutzt: Als Bezeichnung für Schulen ("Bergakademie" Freiberg) und für Wissenschaftler-Vereinigungen. Nachfolgend sind nur letztere gemeint. Ab dem 15. Jh. entstanden Akademien in nahezu allen Ländern nach folgendem Muster:

- Unter Obhut des Landesherrn organisieren sich geachtete Wissenschaftler vereinsanalog; sie wählen weitere Mitglieder auf Lebenszeit und treffen sich zu Vorträgen, zum Erfahrungsaustausch, zu Beratungen über die Perspektive von Wissenschaftsgebieten; sie organisieren sich in "Klassen".
- Es werden kurz- und langzeitige Forschungsvorhaben durchgeführt, Abhandlungen, Sitzungsberichte und Jahrbücher veröffentlicht.
- Ein Statut regelt Rechte, Pflichten, die Arbeitsweise, den Zuwahlmodus.

Akademien änderten ihren Namen, überstanden aber bisher ohne Zwangsauflösungen alle Formen gesellschaftlichen Wandels.

1945 bestanden in der Sowjetischen Besatzungszone drei Akademien:

- Die Preußische Akademie der Wissenschaften zu Berlin, nachfolgend ab 1946 als "Deutsche Akademie der Wissenschaften zu Berlin" (DAW) und nach der Akademiereform ab 1972 als "AdW der DDR" (nachfolgend kurz nur "AdW" genannt) firmierend.
- Die Sächsische Akademie der Wissenschaften mit Sitz in Leipzig (SAW)
- Die Akademie der Naturforscher LEOPOLDINA, Sitz Halle.

Die beiden letztgenannten Akademien arbeiten als Gelehrtengesellschaften bis in die Gegenwart. Neu gebildet wurden in der DDR die *Bauakademie,* die *Akademie der Landwirtschaftswissenschaften* und die *Akademie der Pädagogischen Wissenschaften* (AdPW) als disziplinorientierte Organisationen mit Gelehrtengesellschaft und For-

schungseinrichtungen. Sie befassten sich mit Informatik im Zusammenhang mit ihrer disziplinorientierten Bestimmung.
1946 wurde auf Befehl der Sowjetischen Militäradministration die *Deutsche Akademie der Wissenschaften zu Berlin* (DAW) als "aus der Preußischen AdW hervorgegangen" wieder eröffnet. Sie entwickelte sich als Einheit von *Gelehrtensocietät* und *Forschungsakademie* (geordnete Menge von Instituten und Dienstleistungseinrichtungen). Ab 7.10.1972 trug sie im Ergebnis der Akademiereform den Namen Akademie der Wissenschaften der DDR (AdW), siehe hierzu auch www.bbaw.de.
Zur Gelehrtensocietät der AdW: Die Akademiemitglieder organisierten sich in zuletzt 11 Klassen. Belange der Informatik wurden ab 1981 von einer Gruppe, dann einer *Klasse Informatik, Kybernetik und Automatisierung* behandelt. Ihr gehörten 1982 zwei, 1989 sechs Mitglieder aus der Informatik und neun Mitglieder aus fachübergreifenden Disziplinen an. Der gewünschte Einfluss der Klasse auf die Wissenschaftsentwicklung kam nicht wirklich zustande. Als besondere Aktivitäten sind die Veranlassung der Konferenzen "Zur Bedeutung der Information für Individuum und Gesellschaft" am 29./30.6.1983 und "Zuverlässigkeit und Verfügbarkeit großer technischer Anlagen" am 13.10.1988 sowie eine gemeinsame Sitzung der Klasse mit der Gruppe Informatik der AdPW im Jahre 1987 zum Informatikunterricht an allgemein bildenden Schulen erwähnenswert.
Die Forschungsakademie der AdW wurde 1949 mit einer "Verordnung über die Erhaltung und Entwicklung von Wissenschaft und Kultur" ins Leben gerufen. 1968 bestanden 90 Forschungseinrichtungen, die 1969 teils zusammengefasst und nachfolgend nach dem Leitungssystem Präsident – Forschungsbereichsleiter – Institutsdirektor neu geordnet wurden.
Außeruniversitäre Informatik-Forschung wurde in der DDR insbesondere in folgenden Einrichtungen betrieben:

- Institut für Maschinelle Rechentechnik, 1964 gegründet als Einrichtung der DAW, 1967 überführt in die VVB Datenverarbeitungs- und Büromaschinen und 1969 als Forschungsbereich in das Kombinat Robotron eingegliedert. → [Br06], [JM06] und viele andere Quellen.
- Institut für Datenverarbeitung, 1969 in das Zentrum für Forschung und Technik des VEB Kombinat Robotron eingeordnet → [Me06a], [JM06].
- Zentrum für Rechentechnik (ZFR) / Institut für Informatik und Rechentechnik (iir) der AdW Berlin
- Zentralinstitut für Kybernetik und Informationsprozesse (ZKI) der AdW
- Karl-Weierstraß-Institut für Mathematik der AdW →www.wias-berlin.de.

Forschungsprogramme, Forschungsplanung: Wissenschaft in der DDR sollte „organisch in den gesellschaftlichen Reproduktionsprozess" eingegliedert werden. Forschung gliederte man in *erkundende Grundlagenforschung* als Quelle für prinzipiell neue Erkenntnisse oder Lösungen, in *zielverfolgende Grundlagenforschung* als Weg zu für nützlich gehaltenen Erkenntnissen, Produkten oder Verfahren, und in *angewandte Forschung* mit auf Überführung zielenden Aufgabenstellungen. Die Informatik-Forschung der AdW wird im Jahre 1965 zu etwa 70% Grundlagenforschung gewesen sein, zu vielleicht 30 % im Jahre 1989. Der Anteil von Überführungsleistungen wuchs nicht allein politischen Einflusses wegen, sondern auch infolge des Strebens der Forscher nach sichtbarem, nicht nur deklariertem Erfolg. 1989 hatte das iir 126 VbE (Mannjahre) für Themen mit dem Vertragswert 7,2 MIO M mit Industriekombinaten und der Staatsbank (Computer-

Vernetzung) gebunden.
Die Grundlagenforschung verantworteten der Präsident der AdW und der Minister für das Hoch- und Fachschulwesen (MHF) gegenüber der Regierung gemeinsam. Die Informatik war dem Forschungsprogramm "Mathematik, Mechanik, Kybernetik und Informatik" zugeordnet, 1987 gegliedert in die Hauptforschungsrichtungen (HFR) Mathematische Grundlagen, Grundlagen der Informatik, Verteilte Rechnersysteme, Kybernetik, Künstliche Intelligenz und untersetzt in „Forschungsrichtungen" (FR). Die Basis bildeten die Themen. Für jede Niveauebene darüber gab es (die zuständigen Leiter beratende) "Räte" mit einem "Beauftragten" als Leiter und einem "Stellvertreter" – streng paritätisch (MHF/AdW) besetzt. Präsident und Minister sollten bei der Wahrnehmung ihrer gemeinsamen Verantwortung für die Ausarbeitung, Koordinierung und Kontrolle der Forschungsarbeit unterstützt werden,

Forschungsrichtungen (FR) 1985-1987	*Forschungsrichtungen ab 1988*
Rechnerstrukturen	Rechnerarchitekturen
Daten- und Programmbanken	Datenbanken
Programmiersprachen und Sprachprozessoren	Programmiersprachen
Programmkonstruktion, Technologie der Programmierung	Softwaretechnologie
Betriebssysteme	

Tabelle 1: Die Gliederung der Hauptforschungsrichtung Grundlagen der Informatik

Praktisch jedoch war die Tätigkeit der Räte auf gegenseitige Information, Abstimmungen zur Arbeitsteilung, Abfassung von Berichten über Forschungsergebnisse und die Ausarbeitung von Prognosen beschränkt.
Aus dem Forschungsbericht der FR Grundlagen der Informatik für 1987: Es ruhte die Arbeit in den FR Rechnerarchitektur und Betriebssysteme, die Umbildung wurde beschlossen. 25 Themen werden als abgeschlossen aufgeführt, davon zwei als besonders würdigungswert ("Pascal-Compiler nach C" der Humboldt-Universität Berlin mit Stufe A 4; "Multiprozessorsystem" im iir mit G 4). 11 Themen gehörten zur Angewandten Forschung, 10 zur Grundlagenforschung.
Übrigens: An kritischen Stimmen der „Basis" zur generellen Situation der Informatik-Forschung hat es nachweisbar nicht gefehlt.

2 Zum Forschungsgebiet Informatik in der AdW

Der Vizepräsident für Forschung der AdW stellte Anfang 1990 (selbstkritisch?) fest: "Die Informatik-Forschung und die Informatiker-Ausbildung wurden in der DDR weit weniger staatlich gefördert als die auf Produkte und Technologien gerichteten Disziplinen... . Im Ergebnis davon sind an der AdW insgesamt nur 4 % der Beschäftigten in den beiden Informatik-Instituten ZKI und iir tätig" ([Ad90], S. 14).
Das Zentralinstitut für Kybernetik und Informationsprozesse (ZKI), 1969 gegründet, war eine fachlich etwas inhomogene Einrichtung. Der Institutsteil Dresden (138 Mitarbeiter, davon 46 promoviert, 15 habilitiert) war 1969 aus dem Institut für Regelungs- und Steuerungstechnik der DAW hervorgegangen und blieb eine fachlich und organisatorisch selbständige Einheit. Bis 1980 lag der Schwerpunkt der Arbeiten auf

dem Gebiet Steuerungs- und Regelungstechnik. Unter dem Einfluss der politischen Offensive Mikroelektronik in der DDR wandelte sich das Profil; die Bearbeitung von theoretischen Fragestellungen (Identifikation und Modellbildung in großen Systemen, adaptive Regelung, strukturelle Regelungstheorie) wurde durch praxisrelevante Aufgabenstellungen ergänzt (Steuerung überregionaler Gashochdrucknetze und des Elektroenergieverbundsystems). Die Arbeitsrichtung Schaltsysteme wurde in Richtung "Obere Ebene für ein durchgängiges VLSI-Entwurfssystem" verändert und aus dem Bereich Steuerungstechnik heraus wurden ab 1982 zunehmend Arbeiten zum Komplex "Schaltkreisentwurf / Mikroprozessoren" (mit bis zu 40 Forschern) übernommen. Der Wissenschaftsrat beurteilte diese Arbeiten als sehr qualifiziert [Wi91b]; drei FhG-Institute schlossen sich dieser Auffassung an und bildeten Außenstellen in Dresden.

Im Institutsteil Berlin des ZKI waren auch drei nicht der „Kerninformatik" zurechenbare Bereiche wirksam: Magnetische Signalspeicher (50 Mitarbeiter), Kognitionspsychologie (etwa 20 Mitarbeiter) und Fertigungsautomatisierung (1988 als Institut für Automatisierung ausgegründet). 1990 waren im Institutsteil Berlin insgesamt 403 Personen beschäftigt, davon 257 in wissenschaftlichen Bereichen (92 promoviert, 10 habilitiert). Es gab ein Kleinfertigungszentrum mit Elektronikentwicklung (45 Mitarbeiter).

Forschungsrichtungen:
- Grundlagen / Verfahren der automatischen visuellen Inspektion, der Farb-, Textur- und Linienbildver-arbeitung, Systemlösungen der echtzeitnahen Bildverarbeitung und -analyse (bis zu 60 Mitarbeiter).
- Entwicklung funktionaler KI-Sprachen mit Entwicklung eines LISP-Interpreters/Compilers, Erweiterung von PROLOG, Arbeiten zur Synthese von Programmen für die Bildauswertung; Expertensysteme.
- Projektorientierte Arbeiten auf dem Gebiet Rechnerarchitektur (Entwurf von Spezialprozessoren). Entwicklung von Bildverarbeitungssystemen für das Kombinat Robotron; Ergebnis 1982: BVS A 6471 bis 6473.
- Forschungsarbeiten zur Analyse, Modellbildung und zum Entwurf flexibler und Computer-integrierter Fertigungssysteme (Modellierung von CIM, dispositive Steuerung, automatisierte Generierung von Schemata, CAD-Arbeiten zu Bruchdiagnose und Bruchmechanik).

1989 wurden verstärkt Themen der Modellierung, qualitativen Systemanalyse und mehrkriteriellen Entscheidungsfindung einschließlich der Entwicklung systemanalytischer Softwarewerkzeuge aufgenommen und anwendungsorientierte Projekte bearbeitet, z.B. zum Umweltschutz [Wi91b].

Das ZKI beteiligte sich aktiv am RGW-Programm zur "Entwicklung neuer Generationen von Rechnersystemen" mit dem Thema "Grundlegende Hilfsmittel für die Wissensverarbeitung" und der Leitung des Komplexprogrammes Nr. 2 "Bildverarbeitung und Computergrafik" (Prof. Wilhelmi) [KR85].

Der Wissenschaftsrat meinte bezogen auf einzelne Aufgaben, dass "...auch eine große Zahl an qualitativ hochwertigen und aktuellen Forschungsprojekten durchgeführt wird." ([Wi91b], S. 71). Gesondert genannt werden Bildverarbeitung und Rechnerarchitektur.

Zentrum für Rechentechnik (ZfR) / Institut für Informatik und Rechentechnik (iir)

Im Institut für Hochenergiephysik Zeuthen der DAW (IfH) war in den 60-er Jahren eine "Abteilung Elektronisches Rechnen" entstanden, ein ZRA 1 wurde installiert und genutzt. Erste wissenschaftliche Arbeiten bestanden in der Bereitstellung und Entwicklung von Programmen/Programmbausteinen für Nutzer sowie in der Entwicklung numerischer Verfahren. 1970 wurde die in Moskau gefertigte BESM 6 installiert; wegen unzureichender E/A-Ausstattung wurde sie mit einer SIEMENS 4004/26 gekoppelt. Systemverständnis in Hardware und Software waren gefordert, um das Betriebssystem überarbeiten und das System betreiben zu können. Der Forderung nach Schaffung eines zentralen Dienstleistungsrechenzentrums für die DAW folgend wurde in Berlin-Adlershof am 1.1.1972 das "*Zentrum für Rechentechnik*" (ZfR) durch Zusammenschluss des Bereichs Informationsverarbeitung des ZKI (vorher im ZOS tätig) und der Abteilung Elektronisches Rechnen des IfH (71 Personen) gegründet. Die Investitionsmaßnahme "Aufbau eines Rechenzentrums mit Installation einer BESM 6, gekoppelt mit einer ES 1020" wurde zum ZfR überführt, das Zeuthener Rechenzentrum dem ZfR zugeordnet (ab 1.1.1981 wieder dem IfH zugeführt).

Personalentwicklung, Investitionen, Computerausstattung
Im Dienstleistungs-Rechenzentrum des ZfR/iir waren durchgängig 80 bis 100 Personen tätig (Juni 1990: 85); der Anteil der Forscher stieg kontinuierlich.

Jahr	Beschäftigte	Forscher	Promotion A	Habilitiert
1972	151	27	7	0
1980	214	85	29	3
1984	303	115	40	5
Juni 1990	355	179	65	12

Tabelle 2: Personalentwicklung im iir / Geplant für 1991: 450 Beschäftigte

Das durchschnittliche Monatsgehalt der Beschäftigten betrug 1984 1154 Mark, 1989 1381 Mark. **Investitionsmaßnahmen** wurden für das iir bis 1986 fast ausnahmslos zwecks Installation neuer Rechentechnik durchgeführt. Am 17.05.1989 endlich wurde ein 9-geschosssiges Gebäude fertig gestellt und anteilig durch iir und ZKI bezogen (iir: 125 Arbeitsplätze). Ein weiterer Bauabschnitt mit 50,5 Mio. M war beschlossen (Laborgebäude mit 257 Arbeitsplätzen für das iir + Technikum mit 348 Arbeitsplätzen für ZKI und iir).
Der Bestand an arbeitsplatzgebundener Rechentechnik entwickelte sich auf 0,8 Computerarbeitsplätze/Forscher im Juni 1990 (12 Workstations, 73 AT- bzw. XT-PC und weitere 73 einfache PC).

Forschungsleistungen in der zeitlich ersten Phase konzentrierten sich auf FuE-Arbeiten zur Kopplung der EC 1020 und der BESM 6 sowie die Entwicklung eines neuen Betriebssystems für die BESM 6 ("BAMOS"). Anschließend war die Schaffung eines Rechnernetzes für Forschung und Lehre ("DELTA") auf der Grundlage des Paketvermittlungsnetzes KOMET, verteilt stehender BESM 6-Anlagen und Stapelterminals Schwerpunkt. Es wurden u. a. Kommunikationsdienste für die BESM 6 implementiert, ein Systemprozess für die Bedienung peripherer Geräte bereitgestellt, eine Terminal-

sprache für Operatorkommandos und Nutzeranweisungen programmiert, ein System der Kommunikationsdienste und Übertragungsprotokolle konzipiert, eine erste Ausbaustufe eines Kommunikationssubnetzes KOMET in Form eines Zwei-Knoten-Systems realisiert und getestet (1979). Ein Paketvermittlungsdienst für die entfernte Rechner-Rechnerkommunikation war erarbeitet worden (1979) - Übertragungsleistung 48 KBit/s - und ein Mailbox-Dienst für das Rechnernetz DELTA (1981).
1981 wurde der Nachweis des Zusammenwirkens lokaler und entfernter Komponenten des Netzsteuer- und Arbeitsrechnersystems auf der Grundlage produktiver Arbeit erbracht. Anfangs waren die Rechenzentren in Berlin-Adlershof, Zeuthen und Dresden (TU) Netzbestandteil. Erste Nutzprojekte bezogen sich auf Datentransportaufgaben von Dresden nach Berlin für die Akademie der Landwirtschaftswissenschaften und die Leistungsverlagerung (entfernte Auftragsbearbeitung) von Berlin-Adlershof nach Zeuthen.

Beispiele für Programmentwicklungen im Zeitraum bis etwa 1980 sind eine Bibliothek numerischer Grundmoduln (überführt u. a. zum Rechenzentrum der Sibirischen Abteilung der Akademie der Wissenschaften der UdSSR und andere BESM 6-Nutzer in der UdSSR), das Generierungsprogramm für allgemeine Listenprogramme zu Programmierungs- und Testaufgaben für ökonomische Projekte (GENERAL), ein Programmpaket Isolinien für verschiedene Einsatzgebiete der Digitalgraphik, Arbeiten auf dem Gebiet der automatischen Generierung von Compilern (Compiler-Compiler-System COCOS) in Kooperation mit dem VEB Robotron Zentrum für Forschung und Technik, Programmentwicklungen zur graphischen Ausgabe von Molekülbildern, zum Archivieren und statistischen Auswerten von strukturanalytischen Daten, (genutzt u. a. zur Röntgenkristallstrukturanalyse), die Entwicklung leistungsstarker Verfahren zur Lösung von Anfangswertaufgaben steifer und nicht steifer expliziter gewöhnlicher Differentialgleichungssysteme

Diese Arbeiten begleitend entwickelten sich Bedürfnisse zur *Schaffung von Programmbanken, zur Analyse und Simulation des Verhaltens von Rechnern, Rechnernetzen und verteilten Systemen sowie nach verbesserten und verallgemeinerten methodischen Grundlagen für die Rechner-Rechner- und die Mensch-Rechner-Kommunikation.* Hieraus resultierten verallgemeinerte wissenschaftliche Arbeitsrichtungen des ZfR, die zu einer systematischen inhaltlichen und organisatorischen Loslösung der wissenschaftlichen Arbeitsgruppen und Bereiche von den engeren Aufgabenstellungen des Rechenzentrums führten.
So traten im Zeitraum 1979...1981 in der Arbeitsrichtung Rechnerkommunikation an die Stelle des privaten Netzes die **Technologien öffentlicher Netze** und ihrer Nutzung; bezogen auf die Rechnerkopplungen erweiterte sich das Feld auf allgemeine **Rechnerarchitektur mit Fehlertoleranz / Verlässlichkeit und Echtzeitanforderungen** als Rahmenbedingungen. Die Arbeiten zu Programmbanken wurden weitergeführt, die zur Computergeometrie mündeten in der Bearbeitung von CAD-Problemen: **Produktdatenorganisation und Produktdatenaustausch**. Die Abteilung numerische Verfahren wandte sich CAD-Problemen externer Partner zu. In der letzten Phase der Institutsentwicklung kamen allgemeine Arbeiten zur **Systemanalyse** durch Übernahme einer Arbeitsgruppe aus einer anderen Einrichtung der AdW und so genannte "**Initiativforschungsaufgaben**" aus dem Grundlagenforschungsbereich (GE-Themen) hinzu.
Die Umbenennung des "ZfR" in "Institut für Informatik und Rechentechnik" am

25.07.1984 war die logische Folge dieser Entwicklung; sie erfolgte nicht als Anlass zu grundlegendem Wandel, sondern in Vollzug eines schon laufenden Vorgangs. Rechnerdienstleistungen für Dritte verblieben als Nebenaufgabe.

Forschungsarbeiten im ZfR (ab 1984 iir) 1980-1986

Zu den Rahmenbedingungen: Ab 1980 hatte sich in der DDR die Hinwendung zu "Schlüsseltechnologien" vollzogen. Nach einem ersten Ansatz 1981 wurde 1983 auf Empfehlung der AdW von der Regierung der DDR auch die Schaffung eines Datenpaketnetzes mit der Vorstufe Versuchsdatennetz beschlossen, der VEB Kombinat Robotron war als Finalproduzent benannt und begleitete (erst) ab 1986 Forschungsaufgaben zum "automatisierten Datennetz" als Partner für Software und konstruktiv-technologische Aufgaben. Die Deutsche Post, die Deutsche Reichbahn und die Staatsbank der DDR erwiesen sich als konkret interessierte Nutzer am iir-Programm Kommunikationstechnologien; die Regierung setzte 1988 unter Leitung von Staatssekretär Calow eine "Staatliche Leitergruppe Datennetz" für das Gesamtprogramm Datennetz von der Forschung bis zur Produktions- und Investitionsplanung ein. Am 22.10.1988 billigte der Ministerrat Verhandlungen zum Abschluss eines Vertrages mit SIEMENS über die Lieferung von Technik für ein Paketdatennetz mit 4000 Anschlüssen (pro forma?). SIEMENS schloss einen Vertrag, scheiterte jedoch wie erwartet am Embargo.

Ergebnisse in den Arbeitsrichtungen Rechnerkommunikation und Rechnerarchitektur 1980-1986: Konzipierung des stufenweisen Aufbaues öffentlicher Netze in der DDR mit einer ersten Etappe "Experimentelles Datennetz" und Schaffung aller dazu erforderlichen Grundlagen sowie Hard- und Software-Komponenten (als funktions- und einsatzfähige Muster). Für Netzarchitekturlösungen Privater Netze wurden Beiträge geleistet (Rechnernetzentwurf, Simulation, Modellierung, Leistungsbewertung und Verhaltensanalyse von Rechnernetzen mit praktischer Nutzanwendung im Netz der Banken der DDR), zur Softwarestrukturierung und -entwicklung auf Basis der OSI-Konzepte (einschl. eines allgemeinen Transportstationsmodells 1983 und Entwicklung der Transportschicht), zum Nutzerzugriff auf Netze über Terminals (realisiert mit Computern des VEB Kombinat Robotron unterschiedlicher Varianten z.B. zur Realisierung eines Bürocomputerverbundes für eine integrierte Produktionskette Eisenerz/Verkehr sowie für die Containerumschlagkette Riga-Rostock), Entwicklung eines Mehrprozessor-Paketvermittlungsrechners "BMP8" (8 Bit Verarbeitungsbreite), Arbeiten zur Architektur von Multiprozessor/Multirechnersystemen, Entwicklung eines Programmsystems zur Steuerung des Betriebes schneller Leitungen, umfangreiche Testarbeiten zu Rechnernetzen (auch auf internationalen Strecken) mittels des Netzes DELTA/KOMET und beim Aufbau des Experimentellen Datennetzes .

Zu Beiträgen für die Entwicklung der Softwaretechnologie sind die Arbeiten an einem incrementalen Dialogcompiler (INDIA), Kernstück einer Programmierumgebung für Multiprozessor-Computer, zu rechnen. Arbeiten an einem Programmbanksystem wurden bis zur Einführung in drei Rechenzentren der AdW und des Ministeriums für Hoch- und Fachschulen geführt.

Arbeitsrichtung rechnergestützte Arbeitsplätze: Ergebnisse waren die Implementation des Strukturkerns ALGRA mit Datenverwaltungssystem für einen interaktiven graphischen Arbeitsplatz; weiter wurde ein experimenteller Geometriespeicher als Beitrag für die Entwicklung eines 3D-fähigen CAD-Arbeitsplatzes realisiert.

Arbeitsrichtung numerische Verfahrenstechnik: Algorithmen zur numerischen Lösung von linearen und nichtlinearen Fredholmschen Integralgleichungen zweiter Art sowie von Systemen solcher Gleichungen und deren Implementierung.

1986 ... Februar 1990 bestanden wegen der politischen Förderung von CAD/CAM/CIM-Vorhaben, der (zu späten) Einordnung der Software als volkswirtschaftlicher Schwerpunkt und langsam beginnender Aufgeschlossenheit gegenüber Computervernetzungen günstigere äußere Bedingungen für die Arbeiten des iir als vorher. Arbeitsschwerpunkte in dieser Etappe:

Arbeiten zur Forschungsrichtung Rechnerarchitektur (50 Mitarbeiter):
Die Arbeiten zu verlässlichen Rechnersystemen, fehlertoleranten, verteilten Betriebssystemen, zu Analyse und Test verlässlicher Systeme, Methoden der Fehlervermeidung beim Software-Entwurf und zu integrierten Programmiersystemen für Parallelrechner waren zu Beginn ausschließlich auf Schaffung des Paketvermittlungsknotens für ein X.25-Datennetz gerichtet (im Oktober 1989 wurde das Ergebnis vor Staatssekretär Nendel verteidigt), im Laufe der Jahre jedoch auf Anwendungen für Steuerungs-Computer in sensiblen Einsatzfällen erweitert. Der Wissenschaftsrat (nachfolgend kurz „WR") bewertete die Arbeiten zu Programmiersystemen (Compilergeneratoren, Parallelisierung und Werkzeuge für Parallelrechner) als qualifizierte Grundlagen- und angewandte Forschung, er empfahl die weitere Förderung. Zu den Arbeiten auf dem Gebiet Modellierung und Optimierung (Warteschlangentheorie und -Anwendungen, Petri-Netze) wurde vom WR ausgeführt, dass sie als Grundlagenforschung an Hochschulen fortgeführt werden sollten. Die Arbeiten direkt am Paketvermittlungsknoten (Hardware und Betriebssystem) wurden erwartungsgerecht als industrie- und produktionsnahe Entwicklungsarbeiten klassifiziert und nicht als weiterhin förderungswürdig eingestuft.
Die Arbeiten auf dem Gebiet Rechnerkommunikation (50 Mitarbeiter) hatten angewandte Forschung zur Nutzung moderner Kommunikationstechnologien zum Inhalt. Ein Arbeitsschwerpunkt war die Konformitätsbewertung von Rechnernetzkomponenten (Spezifizierung der Testarchitektur, Testbeschreibungssprache, formale Spezifikation/OSI-Schichten); der WR sprach sich für Fortführung in größerer Forschungsumgebung aus; weiter gab es theoretische Untersuchungen zur Unterstützung kooperativer Gruppenarbeit mittels verteilter Systeme (mit prototypischem Nachweis von Gruppenunterstützungssystemen); der WR empfahl die Angliederung dieser Gruppen an eine Einrichtung der Grundlagenforschung. 1989 konnten Verträge über Auftragsforschung mit dem Konzern SEL abgeschlossen werden (ISDN-Endsysteme), die Mitarbeiter gingen 1990 direkt zu SEL. Aufgebaut wurde auch die Kommunikationsinfrastruktur des iir (Ethernet-LAN für 32-Bit-Computer, LAN mit 60 Stationen incl. Betrieb eines Mitteilungssystems); ein Forschungsnetz für die AdW und ein Wissenschaftsnetz für die DDR wurden projektiert, das iir wurde an das WIN der BRD mit bemerkenswerter Unterstützung des DFN-Vereins (Herr Ullmann) angeschlossen. Die betreffenden Mitarbeiter sind ab 1990 dort tätig geworden.
Arbeitsgebiet Produktdaten (40 Mitarbeiter): Zu den Grundlagenarbeiten gehörten eine Analyse der Basisfunktionen der Produktdatenorganisation sowie Untersuchungen zur algebraischen Spezifikation von Produktdaten. Arbeiten in der angewandten Forschung waren der Produktdatenaustausch (Entwicklung und Implementation von IGES-Prozessoren) sowie zu Komponenten von STEP-Prozessoren. Weitere Arbeiten bezogen

sich auf geometrisch-technische Datenverwaltungssysteme. Ergebnisse wurden in CIM-Vorhaben des Werkzeug- und des Druckmaschinenbaues überführt.
Die Arbeiten zu Programmbanken mündeten im verteilten Programmbanksystem SORBAS. Die Werkzeuge für Erzeugung, Manipulation und Integration von Metaschemata sowie externen Schemata für Anwendungsgruppen wurden implementiert und mit einer Programmbank mit 15 Sachgebieten und über 2000 Programmen betrieben; eine Methode für die Implementierung einer wissensbasierten Endnutzerstelle wurde gestaltet, am Beispiel eines Sachgebietes wurden die Komponenten für die Wissensrepräsentation und -nutzung in einem frameorientierten System abgeleitet und implementiert.
Themen der numerischen Verfahrenstechnik: Berechnung des Elektronenstrahlverlaufs in komplexen elektronischen Systemen (Farbbildröhren), Analyse, Synthese und Simulation thermohydraulischer Prozesse in Rohrleitungssystemen (zwecks Leckidentifizierung mit Ableitung von Havariekonsequenzen speziell in Kernkraftwerksanlagen). Der WR gab ein positives Votum.
Arbeiten unter der Bezeichnung "Systemanalyse" wurden im iir ab 1989 zu massiv parallelen, adaptiven und allgemein naturanalogen Prinzipien der Informationsverarbeitung durchgeführt mit Ergebnissen zu Replikatornetzen auf der Grundlage von Modellen der Evolutionstheorie.
Initiativforschungsaufgaben dienten als Basis für künftig geplante breitere Grundlagenforschung. Die Arbeiten bezogen sich auf "Gestaltungsgrundsätze der Architektur komplexer Informationssysteme", die Verbesserung/Optimierung von Prozessorarchitekturen und "untere Schranken für die algorithmische Komplexität ausgewählter kombinatorischer Probleme". Zwei dieser 4 Mitarbeiter wurden nach 1990 zu Professoren neuen Rechts berufen.

Das iir beteiligte sich an der **Zusammenarbeit der Akademien sozialistischer Länder zur Informatik** ab 1962 [Me04] und ab 1985 am RGW-Forschungsprogramm Neue Generationen von Rechnersystemen; es übernahm hier die Verantwortung für die Komplexprojekte Nr. 4 "Rechnernetze" (Leiter: V. Heymer) und Nr. 6 "Fehlertolerante Systeme" (Leiter D. Hammer) [KR85].

3 Absturz in die Zukunft

So kennzeichnete der am 17.05.1990 erstmals in der Akademiegeschichte von einem aus von allen Mitarbeitern der AdW gewählten Konsilium gewählte neue Akademiepräsident H. Klinkmann die letzte Phase der Existenz der AdW [Kl91]. Für die Akademie als Gesamtheit ergab sich ihre Perspektive in einem vereinigten Deutschland bereits aus der Entscheidung der Bundesregierung Anfang 1990, die Wiedervereinigung über den *Anschluss* neuer Bundesländer zu regeln und damit in allen Belangen Struktur und Funktionsweise der BRD als Basis zu nehmen. Akademien wie die AdW mit eigenen Forschungsinstituten waren somit nicht systemkompatibel. Die Sächsische Akademie und die LEOPOLDINA wurden nach 1991 neu belebt →www.saw-leipzig.de, www.leopoldina-halle.de.
Zur **Perspektive der Gelehrtengesellschaft** der AdW gab es wochenlang erhebliche Differenzen zwischen den Ministerien in Ost und West, konkret um die Aufnahme der Wörter "ob und" in den Einigungsvertrag: "Die Entscheidung, (ob und) wie die Gelehr-

tensocietät der AdW der DDR fortgeführt werden soll, wird landesrechtlich getroffen." Beim Vertragstext setzte sich Prof. Terpe mit seiner Forderung durch, dass das Weiterbestehen der Gelehrtengesellschaft nicht durch das Wort "ob" in Frage gestellt werden sollte. Dennoch verfügte das Bundesland Berlin in eigener Verantwortung auch das Ende der Gelehrtengesellschaft. Eine Berlin-Brandenburgische Akademie der Wissenschaften konstituierte sich am 28.03.1993 neu →www.bbaw.de.
Der Forschungslandschaft der DDR mit der **AdW-Forschungsakademie** als Konzentrat außeruniversitärer Forschung stand eine grundlegend andere Struktur der Forschungsinstitutionen in der BRD gegenüber: Die Max-Planck-Gesellschaft MPG (mit einem Institut für Informatik in Saarbrücken); 13 Großforschungseinrichtungen mit über 20 000 Mitarbeitern (darunter die Gesellschaft für Mathematik und Datenverarbeitung mit 700 Mitarbeitern), die Fraunhofer-Gesellschaft FhG mit 33 Forschungsinstituten und 3560 Mitarbeitern (darunter 389 im damals so bezeichneten Sektor IT); und letztlich die Einrichtungen der Blauen Liste (1990 35 Einrichtungen; darunter wie heute noch keine Informatik-Einrichtung) [Bu90]. Versuche, die Forschungsakademie als Ganzes mit neuen Zielsetzungen und personellen Änderungen oder mindestens den Teil Informatik-Forschung als Großforschungseinrichtung analog zur GMD (oder als deren Teil) zu erhalten schlugen fehl. Die vom zuständigen DDR-Minister vertretene Forderung, im Einigungsvertrag den Fortbestand der Forschungsinstitute der AdW als Einrichtungen der Länder festzulegen [MF90], fand keine Zustimmung. Neben politischen Motiven gab es auch ein Mengenproblem, weniger ein Qualitätsproblem: "Die DDR verfügt entgegen dem allzu vereinfachten Querschnittsbild in der Öffentlichkeit (10 Jahre zurück etc.) über eine qualitativ durchaus hochwertige, selbstverständlich unterschiedliche Forschungs- und Entwicklungsqualität" [Bm90]. Im IT-Bereich (ohne Nachrichtentechnik und Mikroelektronik) waren 1989 in der BRD insgesamt 750 Personen forschend tätig (389 FhG, 350 GMD, 8 MPG) [Bu90], allein im Sektor Informatik der AdW 538 Personen. Folglich gab es bereits im März 1990 im BMFT Überlegungen zur Fragestellung "Was soll mit der (zu) großen FuE-Kapazität der DDR im Bereich der Informationstechnik geschehen?" [Bm90]. "Filetstücke aus der DDR- Forschungslandschaft" heraussuchen, oder eine leistungsfähige Forschungsinfrastruktur" erhalten? Schließlich wurde das Problem bei sich "deckenden Interessenslagen der Forschungseinrichtungen und der Bundesregierung" gelöst: Evaluierung der Akademieinstitute durch den Wissenschaftsrat, auf dessen Ergebnisse stützend sollten MPG, GMD und FhG in Abstimmung mit den Regierungen der Neuen Länder über die Bildung neuer Einrichtungen ("bottom up, peer review, und das sofort") befinden. So geschah es.

Die Zeit ab Ende 1988 bis zur Auflösung der AdW Ende 1991 war durch starke, auf Demokratisierung gerichtete politische Veränderungen in den Instituten und in der AdW-Leitung gekennzeichnet, allerdings auch gehemmt durch solche Kräfte im Plenum, die die vorgeschlagenen prinzipiellen Änderungen wie die Trennung von Gelehrten- und Forschungsakademie blockierten.

Im iir bestand schon im Frühjahr 1990 Klarheit über die zu erwartende Auflösung des Instituts, bedingt durch Kontakte in die BRD. Es wurden bei noch ausstehender Evaluierung zielgerichtet neue Arbeitsplätze für alle Mitarbeiter gesucht. Im August 1990 beschloss der Vorstand von SIEMENS den Aufbau eines neuen Entwicklungszentrums (Leitthema "Intelligente Netze") mit 55 Mitarbeitern, stellte im iir offiziell sein Konzept

vor und warb vor Ort Mitarbeiter; am 1.10.1990 nahm es unter Leitung von Prof. D. Carl, Bereichsleiter im iir, seine Arbeit auf und entwickelte sich stabil weiter. Ein „Filetstück“ war vergeben. 27 Mitarbeiter fanden bei SEL eine Arbeitsstelle. Der Vorstand der GMD beriet im Dezember 1989 über mögliche Zusammenarbeit mit Einrichtungen der DDR und stellte fest: “Die GMD pflegt seit längerem zunehmend gute Beziehungen zur AdW, insbesondere mit dem iir”; Personen wurden genannt [Ge89]. Am 26.01.1990 wurde ein MEMORANDUM zur Zusammenarbeit an Software für rekonfigurierbare Hochleistungsparallelrechner unterzeichnet. Die GMD rechnete damals noch mit 45 zusätzlichen Planstellen ab 1991 in Berlin für Erweiterungen bei FIRST und FOKUS – nicht wissend, dass die GMD insgesamt bald nicht mehr existieren würde. Ähnlich rege Beziehungen gab es vom iir zur FhG. Folge insgesamt: Die Beschäftigtenzahl im iir ging schon vor dem offiziellen Ende, unabhängig von der Evaluierung, sehr stark zurück: Von 355 Mitarbeitern am 01.07.1990 auf 143 am 01.07.1991 und 99 am 31.12.1991 (44 davon hatten ab 1.1.1992 keinen Arbeitsvertrag, 22 davon aus der Verwaltung).

Im Ergebnis der Evaluierung empfahl der Wissenschaftsrat im März 1991, von den noch vorhandenen Mitarbeitern des iir 130 in einem in Berlin-Adlershof neu zu bildenden Fraunhofer-Institut für informationstechnische Systeme, Werkzeuge und Methoden (ISST) einzustellen ([Wi91a], S. 44). Zur “Bedeutung des iir” ([Wi91b], S. 30) gab es u. a. folgende Feststellung: “Die Beschreibung und Bewertung der Aufgaben des iir ergibt, dass im iir sowohl in der Grundlagenforschung als auch in der angewandten Forschung Arbeitsgruppen qualifizierte Arbeiten durchführen, die auch unter den veränderten Arbeitsbedingungen aktuell sind und fortgeführt werden sollten. Darüber hinaus hat die Arbeitsgruppe des Wissenschaftsrates bei ihrem Besuch des iir und dabei in den Gesprächen mit den wissenschaftlichen Mitarbeitern überwiegend gute Fachkenntnisse angetroffen. Das gilt vor allem auch für die jüngeren Mitarbeiter. Die in den letzten Monaten erfolgte Fluktuation von wissenschaftlichem Personal in die Wirtschaft bestätigt diesen Eindruck”.
Die FhG entschied sich jedoch letztlich für den Aufbau eines eher bescheidenen ISST im Zentrum Berlins (33 Mitarbeiter) und parallel dazu in Dortmund (17 Mitarbeiter). 2008 bestehen im ISST in Berlin die Abteilungen Location-based Services, Sichere Business-IT-Infrastrukturen (Geschäfts- und Verwaltungsabläufe) sowie Verlässliche technische Systeme (Unterstützung von Automobilherstellern und Zulieferanten).
In den Neuen Ländern engagierte sich die FhG wohl stark auch zur Informationstechnik, nicht aber zur Informatik. Den Länderregierungen unterbreitete Vorschläge, in den neuen Ländern dann wenigstens an Universitäten Forschungsgruppen Informatik zu bilden und daraus außeruniversitäre Forschungs-Einrichtungen Informatik zu entwickeln, blieben leider ohne Resonanz – nur bisher?

Literaturverzeichnis

[Ad90] AdW der DDR, Forschungsgemeinschaft: Kurzcharakteristik der Institute und Einrichtungen..., Teil I (Gesamtdarstellung). 23.06.1990; Druckschrift der AdW.

[Be70] Beschluss des Staatsrates der DDR zur weiteren Durchführung der Akademiereform; GBl der DDR 1970, Teil I, Nr. 7, Seite 19

[Bm90] BMFT Abteilung 414-9518-31; Vorlage 1/90 (2/90) vom 29.03.1990 (12.04.1990): Was soll mit der (zu) großen FuE-Kapazität der DDR im Bereich

der Informationstechnik geschehen? (Neuorientierung des Förderprogramms "Auftragsforschung und –entwicklung für die Zusammenarbeit mit der DDR")
[Br06] Brosch, H.-J.: Informatik-Grundlagenforschung im Kombinat Robotron – ein Erfahrungsbericht. In (Naumann, F., u. Schade, G.): Informatik in der DDR - eine Bilanz. Gesellschaft für Informatik, Bonn 2006; S. 501-512.
[Bu90] Der Bundesminister für Forschung und Technologie: Faktenbericht 1990 zum Bundesbericht Forschung; Bonn 1990.
[Ge89] GMD: Vorstands-Vorlage vom 22.12.1989 "Mögliche Zusammenarbeit der GMD mit Einrichtungen der DDR auf dem Gebiet der Informatik und Informationstechnik"; 7 Seiten.
[JM06] Junge, S.; Merkel, G.: Die zentrale FuE-Einrichtung des VEB Kombinat Robotron. Ein Beitrag zur Industriegeschichte der Stadt Dresden. 209 Seiten. http://robotron.foerderverein-tsd.de/
[Kl69] Klare, H.: Vortrag zum Leibniz-Tag am 3.07.1969. In: Jahrbuch der Deutschen Akademie der Wissenschaften zu Berlin 1969. Akademie-Verlag GmbH Berlin, Seite 31 bis 39.
[Kl91] Klinkmann, H.: Absturz in die Zukunft – Die Akademie der Wissenschaften der DDR in der Wendezeit 1989/90. Sonderdruck der Gelehrtensocietät der ehemaligen AdW der DDR.
[KR85] Koordinierungsrat für Rechentechnik und Informatik der Akademien der Wissenschaft der sozialistischen Länder: Konzeption zur Entwicklung neuer Generationen von Rechnersystemen. Bearbeiter D. Pötschke, ZKI der AdW der DDR, III-12-12 Ag 521/546/87, 334 Seiten
[Me04] Merkel, G.: Zur Tätigkeit der „Kommission für Mehrseitige Zusammenarbeit der Akademien der Wissenschaften sozialistischer Länder zu Problemen Wissenschaftlicher Fragen der Rechentechnik 1962 bis 1990. Bericht Dezember 2004, 153 Seiten; Archiv des Deutschen Museums München.
[Me06a] Merkel, G.: VEB Kombinat Robotron; Ein Beitrag zur Industriegeschichte der Stadt Dresden. 44 Seiten. Stadtarchiv Dresden und http://robotron.foerderverein-tsd.de/111.html.
[Me06b] Merkel, G.: Institut für Datenverarbeitung (idv); Ein Beitrag zur Industriegeschichte der Stadt Dresden. 28 Seiten. Stadtarchiv Dresden und http://robotron.foerderverein-tsd.de/.
[MF90] Textvorschlag des MFT ... zum Entwurf des Einigungsvertrages ... präzisiert entsprechend einer Beratung zwischen den Ministern Prof. Terpe und Dr. Riesenhuber am 13.08.1990 im MFT
[VO72] VO vom 23. 08. 1972 und vom 12.12.1985 über die Leitung, Planung und Finanzierung der Forschung an der AdW und an den Universitäten und Hochschulen
[Vo97] Vorlage zur Beratung im Kollegium der AdW am 21.01.1987: Ergebnisse, Probleme und weitere Entwicklung des iir der AdW der DDR; VD iir-D/9/86/30.
[Wi91a] Wissenschaftsrat (Drucksache 320/91 vom 5.7.1991): Stellungnahmen zu den außeruniversitären Forschungseinrichtungen in den neuen Ländern und in Berlin.
[Wi91b] Wissenschaftsrat (Drucksache 92/91 vom 13.03.1991: Stellungnahmen zu den außeruniversitären Forschungseinrichtungen der ehemaligen Akademie der Wissenschaften der DDR in den Fachgebieten Mathematik, Informatik, Automatisierung und Mechanik; 145 Seiten.

Forschung zu Abstrakten Datentypen im ZFT des Kombinates ROBOTRON

Horst Reichel

reichel@tcs.inf.tu-dresden.de

Abstract: Im folgenden werden die Anfänge, die wesentlichsten Ergebnisse und die damaligen Randbedingungen der Forschungen auf dem Gebiet der Abstrakten Datentypen der Gruppe Hupbach, Kaphengst Reichel im VEB Kombinat Robotron, ZFT, Fachgebiet Grundlagenforschung, dargestellt. Abschließend wird der Einfluss dieser Arbeiten auf den aktuellen Stand der Algebraischen Spezifikationssprachen diskutiert.

1 Einleitung

In der zweiten Hälfte der 60er Jahre des vergangenen Jahrhunderts wurden nichtnumerische Datentypen meistens mittels der Programmiersprache LISP dargestellt. Diese Sprache war in der akademischen Welt sehr verbreitet, da man mit ihr symbolische Ausdrücke gut manipulieren konnte. Dies bedeutete jedoch, dass man einen nichtnumerischen Datentyp, nämlich die geschachtelten Listen bereitstellt, mit dem dann alle anderen nichtnumerischen Datentypen implementiert werden müssen. Die Semantik von LISP, als einer der ersten funktionalen Programmiersprachen, basiert auf dem Lambda–Kalkül von Church, das in den 30er Jahren im Rahmen der Berechenbarkeitstheorie entwickelt wurde. Da der Lambda–Kalkül eigentlich nur Funktionen kennt, war damit keine gute Grundlage für die Semantik von nichtnumerischen Datentypen gegeben. Deshalb bestand damals ein allgemeines Bedürfnis, eine *implementationsunabhängige, formale Semantik für Datentypen* von Programmiersprachen zu entwickeln. Dieser Aufgabe stellte sich eine kleine Arbeitsgruppe im damaligen Institut für Maschinelle Rechentechnik, das von Professor N.J. Lehmann gegründet wurde und damals von Professor Adler gleitet wurde. Später wurde dieses Institut als ein Fachgebiet in das Zentrum für Forschung und Technik des Kombinates Robotron integriert.

Die Arbeitsgruppe bestand aus U. Hupbach, K.-H. Kaphengst und dem Autor. Kaphengst brachte fundierte Kenntnisse aus der Automatentheorie und der Theorie berechenbarer Funktionen mit und der Autor hatte sich mit Kategorientheorie und Universaler Algebra, insbesondere mit freien algebraischen Strukturen beschäftigt. Hupbach arbeitete an einer Dissertation, mit der er nachweisen konnte, dass berechenbare Funktionen über Bäumen auf berechenbare Funktionen über natürlichen Zahlen oder über endlichen Zeichenketten zurückgeführt werden können.

Diese Kombination der fachlichen Voraussetzungen erwies sich als ein glücklicher Um-

stand. Aus algebraischer Sicht war nämlich die axiomatische Definition der natürlichen Zahlen nach Dedekind die Definition einer freien, algebraischen Struktur. Die endlichen Zeichenketten über einem Alphabet und einige Klassen von Bäumen konnten ebenfalls als freie, algebraische Strukturen identifiziert werden. Damit lag in der Arbeitsgruppe die Vermutung nahe, *freie, algebraische Strukturen für die axiomatische Definitionen von Datentypen* zu nutzen.

2 Initiale Semantik Abstrakter Datentypen

In einer Klasse $\mathcal{K}$ typgleicher algebraischer Strukturen heißt eine algebraische Struktur $\mathbf{I}$ *initial*, wenn

1. $\mathbf{I} \in \mathcal{K}$ und
2. für jede Algebra $\mathbf{A} \in \mathcal{K}$ genau ein Homomorphismus $h : \mathbf{I} \rightarrow \mathbf{A}$ existiert.

Damit ergibt sich, dass im Falle der Existenz einer initialen Algebra $\mathbf{I}$ in $\mathcal{K}$, diese bis auf die Wahl der Bezeichnung eindeutig bestimmt ist. Die Eindeutigkeit bis auf die Wahl der Bezeichnung führte zu dem Begriff *Abstrakter* Datentyp. Aus der Universalen Algebra war bekannt, dass für eine Klasse von Algebren, die durch eine Menge von bedingten Gleichungen definiert ist, die Existenz einer initialen Algebra in dieser Klasse gesichert ist. Ein Abstrakter Datentyp ist somit durch einen algebraischen Typ und eine Menge von bedingten Gleichungen bestimmt. Die innere Struktur eines derartig definierten Datentyps ergibt sich aus der Faktorisierung der Menge der Grundterme über dem algebraischen Typ nach der von den bedingten Gleichungen erzeugten Kongruenzrelation.

Etwas anschaulicher kann man sagen, dass ein Datentyp durch eine endliche Menge von Konstanten und ein– oder mehr–stelligen Konstruktoren gegeben ist, wobei die bedingten Gleichungen beschreiben, ob verschiedene Kombinationen von Konstruktoren gleiche Werte repräsentieren. Die *natürlichen Zahlen* sind auf diese Weise durch eine Konstante 0, einen einstelligen Konstruktor $\_ + 1$ und eine leere Menge von Gleichungen gegeben.

Ein gleichartiger Ansatz wurde auch am IBM Watson Research Center verfolgt [GTW$^+$75]. Allerdings gab es damals keine Möglichkeit zu einem fachlichen Dialog mit der Arbeitsgruppe bei IBM. Das Anfordern von Sonderdrucken war nicht erlaubt. Deshalb versuchten wir, uns einem möglichst guten Überblick über die aktuelle Entwicklung in der Welt dadurch zu verschaffen, dass wir Rezensionen für *Mathematical Reviews* anfertigten. Etwa eine bis zwei von zehn rezensierten Arbeiten war dabei für den eigenen Problembereich nützlich.

Unser Ansatz zielte jedoch von Anfang an darauf ab, partielle Algebren zu verwenden, da aus der Rekusionstheorie bekannt war, dass partielle, berechenbare Funktionen benötigt werde. Derartige Funktionen entstehen durch nicht terminierende Turingmaschinen oder Schleifenkonstruktionen. Das führte aber zu einem ernsthaften Problem. Die Theorie partieller Algebren war in der Mathematik damals nicht so weit entwickelt, dass sie für unseren Ansatz verwendet werden konnte. Da in freien Algebren die Definitionsgebiete der parti-

ellen Operationen minimal sind, werden sie leer, und die entsprechenden Konstruktoren können keine neuen Elemente im Datentyp erzeugen. Ein typisches Beispiel ist der parametrische *Datentyp aller endlichen Wege über einem gerichteten Graphen*. Ein Konstruktor für diesen Datentyp müsste eine Kante an den Anfang eines Weges hängen können, genau dann, wenn der Endpunkt der Kante mit dem Anfangspunkt des Weges übereinstimmt. Deshalb hatten wir zunächst einen Zugang zu partiell, mehrsortigen Algebren entwickelt, der für unseren Ansatz verwendbar war [KR72].

Die Grundidee dieses Ansatzes besteht darin, die Operationen beginnend mit den totalen Operationen hierarchisch zu ordnen und einer Operation der Ebene n als Definitionsbedingung ein Gleichungssystem zuzuordnen, das nur Operationen niedrigerer Ebenen verwendet darf. Das Gleichungssystem wird als notwendige und hinreichende Definitionsbedingung für die Operation der Ebene n verwendet. Damit kann im obigen Beispiel die notwendige und hinreichende Definitionsbedingung für das Anfügen einer Kante k an den Anfang eines endlichen Weges w über einem gerichteten Graphen durch die totalen Operationen α, β dargestellt werden, die jeweils einer Kante deren Endknoten $\beta(k)$ und einem endlichen Weg dessen Anfangsknoten $\alpha(w)$ zuordnen. Die Definitionsbedingung ist dann durch die Gleichung $\beta(k) = \alpha(w)$ gegeben. Von den Homomorphismen zwischen den so entstehenden *gleichungs–partiellen Algebren* wird nur gefordert, dass sie mit Gleichungen verträglich sind, d.h., sie erhalten die Definiertheit, müssen diese aber nicht reflektieren.

Eine erste Fassung unseres Ansatzes zur axiomatischen Spezifikation Abstrakter Datentypen wurde in einem Forschungsbericht des ZFT dargestellt [KR71]. Aus dem Ansatz wurde dann eine Entwurfssprache für Programmsysteme entwickelt [KR76] und die formalen Grundlagen wurden überarbeitet [HKR80]. Heute würde man diese Sprache eine algebraische Spezifikationssprache nennen.

Allerdings konnten diese Berichte keinen Einfluss auf die wissenschaftliche Öffentlichkeit haben. Dazu hätte der Ansatz in einer internationalen wissenschaftlichen Zeitschrift veröffentlicht werden müssen. Deshalb wurden von der Arbeitsgruppe Beiträge zu den internationalen Konferenzen FCT und MFCS erstellt. Diese Konferenzen fanden regelmäßig in der DDR, in Polen und in Tschechien statt, wurden aber auch von Wissenschaftlern Westeuropas besucht. Unser erster Versuch [KR77] war nur teilweise erfolgreich, da unser Beitrag zwar angenommen, wir aber nicht Mitglieder der DDR–Delegation zur FCT–Konferenz waren. Erst als 1980 nur die beiden Beiträge von Hupbach und dem Autor ([Hup80], [Rei80]) zur MFCS–Konferenz aus der DDR angenommen wurden, gelang es uns, mit Unterstützung der Leitung des ZFT, an dieser Konferenz teilzunehmen und unsere Beiträge zu präsentieren. Dies war die erst Möglichkeit, die bisherigen Arbeiten der wissenschaftlichen Öffentlichkeit bekannt zu machen. Eine weitere Möglichkeit, mit Wissenschaftlern der westlichen Welt in Kontakt zu kommen, waren die thematisch orientierten Semester am Banach Zentrum in Warschau, das von den Akademien der sozialistischen Länder organisiert und betrieben wurde. Der Autor hatte die Möglichkeit, 1978 an einem Semester zu den algebraischen Grundlagen der Theoretischen Informatik teilzunehmen. Dabei konnte er mit Rod Burstall von der Universität Edinburgh, einem international führenden Wissenschaftler auf dem Gebiet der algebraischen Spezifikationssprachen, ausführlich über die Arbeiten im ZFT zu sprechen. Dieser Kontakt war besonders wertvoll, da er später, als der Autor an der TU Magdeburg war, weiter gepflegt und vertieft werden

konnte.

Nach einigen Experimenten wurde jedoch deutlich, dass die initiale Semantik für Datentypen zu restriktiv ist. Im Falle von Kellerspeichern wird beispielsweise eine traditionelle Implementation mittels Arrays und Zeigern als nicht korrekt ausgeschlossen. Diese Beobachtung führte zu einer Verallgemeinerung des Ansatzes der initialen Semantik, indem die *beobachtbare Äquivalenz* von Elementen eines Datentyps eingeführt wurde und die Gültigkeit von Gleichungen nicht durch die Gleichheit sondern durch die beobachtbare Äquivalenz definiert wurde [Rei81] [Rei85a]. Diese Verallgemeinerung der Gleichungslogik führte zur *Verhaltensspezifikation von Datentypen*.

Bis in die zweite Hälfte der 70er Jahre hatte unsere Arbeitsgruppe die volle Freiheit in der Wahl der Arbeitsgebiete. Da sich diese Situation jedoch gegen Ende der 70er Jahre entscheidend änderte, bemühte sich der Autor, seine Forschungen an einer Universität weiterführen zu können. Er wurde im September 1980 an die Technische Universität Magdeburg berufen und konnte nun die Forschungen in internationale Aktivitäten einbringen. Er konnte regelmäßig an den WADT–Workshops teilnehmen und er wurde in die Arbeit der EU–Workingroup COMPASS einbezogen. In diesem Zusammenhang entstanden unter anderem die Publikationen [Rei85a], [Rei85b], [Rei86]. Eine zusammenfassende Darstellung des Ansatzes wurde mit dem Buch *Initial Computability, Algebraic Specifications and Partial Algebras* [Rei87] gegeben. Er wurde Gründungsmitglied der IFIP TC1 Working Group, WG 1.3 *Foundations of System Specification*.

Eine der Aufgaben der IFIP WG 1.3 war es, aus den verschiedenen Ansätzen zu algebraischen Spezifikationssprachen eine gemeinsame Spezifikationssprache zu entwickeln. Dazu wurde die CoFI–Initiative [1] ins Leben gerufen. Im Ergebnis dieser Initiative entstand die Spezifikationssprache CASL (Common Algebraic Specification Language) und der IFIP State–of–the–Art Report *Algebraic Foundations of Systems Specification* [AKKB99] in dem der Autor an zwei Kapiteln beteiligt war [CMR99], [Rei99].

Im Rahmen eines regelmäßigen Forschungsseminars, das von Dr. Wolfgang Wechler [2] an der Sektion Mathematik der TU Dresden organisiert wurde, konnte der Autor einen engen fachlichen Kontakt zu Hupbach und Kaphengst aufrechterhalten. Allerdings hatten diese nun leider nicht mehr die fachlichen Freiräume, um sich intensiv mit der Weiterentwicklung des gemeinsamen Ansatzes zu beschäftigen.

3 Spezifikation zustandsorientierter Systeme

Mit der Erweiterung des Blickwinkels von der Spezifikation von Datentypen zur Spezifikation von Systemen rückten *Koalgebren* in das Zentrum des Interesses. Der Autor erkannte Koalgebren als eine geeignete semantische Basis für die Spezifikation zustandsorientierter Systeme und als semantische Basis für die objektorientierte Programmierung [Rei95]. Zugleich sah er im koalgebraischen Zugang eine konsequente Weiterentwicklung der Verhaltensspezifikation von Datentypen zur Spezifikation von reaktiven Prozesstypen.

[1] `http://www.informatik.uni-bremen.de/cofi/wiki/index.php/CoFI`

[2] Autor der sehr beachteten EATCS–Monographie [Wec92]

Koalgebren sind im Sinne der Kategorientheorie eine duale Konstruktion zu Algebren. Wenn für algebraischen Strukturen das Definitionsgebiet einer Operation ein strukturierter Datentyp, im klassischen Fall ein Kartesisches Produkt der Trägermengen der Algebra, und der Wertebereich eine der Trägermengen der Algebra ist, so ist im Falle von Koalgebren das Definitionsgebiet eine der Trägermenge der Koalgebra und der Wertebereich ein strukturierter Datentyp.

Ein typisches Beispiel für eine Koalgebra ist

$$f : S \to S^{(Out \times \mathcal{P}_\omega(S))^{In}}$$

wobei S die Zustandsmenge eines nichtdeterministischen Automaten mit Eingabe In und Ausgabe Out, und die koalgebraische Operation f ordnet jeden Zustand $s \in S$ und jedem Eingabewert $x \in In$ ein Paar $(f(s))(x) = (y, \mathcal{S}) \in Out \times \mathcal{P}_\omega(S)$ zu, so dass y ein Ausgabewert und $\mathcal{S}$ die endliche Menge von potentiellen Nachfolgezuständen ist ($\mathcal{P}_\omega(S)$ bezeichnet dabei die Menge aller endlichen Teilmengen von S). Homomorphismen zwischen Koalgebren entsprechen Simulationen. Von großer Bedeutung ist die Tatsache, dass in der Klasse aller nichtdeterministischen Automaten, des obigen Typs, mit festen Mengen von Ein– und Ausgabewerten In und Out, eine terminale Koalgebra $\mathbf{T}$ existiert, zu der es von jeder Koalgebra dieser Klasse genau einen Homomorphismus gibt. Dies bedeutet, dass $\mathbf{T}$ das Verhalten jedes anderen Automaten aus dieser Klasse simulieren kann.

Die Existenz *terminaler Koalgebren* ist nicht nur auf die obige Klasse von Koalgebren beschränkt. Somit können terminale Koalgebren als *abstrakte Prozesstypen* verstanden werden. Entsprechend dieser Intuition wurde vom Autor angeregt, eine Erweiterung der Spezifikationssprache CASL zu COCASL zu entwickeln, mit der sowohl Abstrakte Datentypen als auch Abstrakte Prozesstypen spezifiziert werden können [MSRR06].

So wie die initiale Semantik für Abstrakte Datentypen jeweils ein Induktionsschema als Definitions– und Beweisprinzip für den jeweiligen Datentyp bereitstellt, erhält man durch die terminale Semantik für Abstrakte Prozesstypen mit der *Ko–Induktion* ein Definitions– und Beweisprinzip für den jeweiligen Prozesstyp. Das koinduktive Beweisprinzip verallgemeinert die *Bisimulation*, ein allgemein bekanntes Beweisprinzip für Prozesskalküle wie CCS von R. Milner. Die Kombination von Induktion und Koinduktion liefert eine große Ausdrucksstärke, sie erlaubt es, Sicherheits– und Lebendigkeitseigenschaften von reaktiven Systemen darzustellen.

Die Untersuchung von Koalgebren und deren Anwendungen in der Informatik ist inzwischen zu einem eigenständigen Forschungsgebiet geworden. Seit 1998 finden jährlich CMCS–Workshops als Sattelitenworkshops zu ETAPS statt (CMCS steht für Coalgebraic Methods in Computer Science). Der Autor war einer der Initiatoren dieser Workshopserie und ist seit Beginn der CMCS–Workshopserie Mitglied des Steering Committe.

4 Schlussbemerkungen

Es war ein glücklicher Umstand, dass sich die fachlichen Voraussetzungen und Interessen in der Arbeitsgruppe gut ergänzten. Auf dieser Basis konnte ein Ansatz zur Spezifikati-

on von Datentypen entwickelt werden, der auch in die internationale Entwicklung Bleibendes einbrachte, obwohl längere Zeit keine Möglichkeit zur Kommunikation mit der internationalen, wissenschaftlichen Öffentlichkeit bestand und die Forschung zur Semantik von Programmiersprachen in der DDR kein Forschungsschwerpunkt war. So wurden in die gemeinsame algebraische Spezifikationssprache CASL partielle Konstruktoren und partielle Funktionen über Datentypen aufgenommen. Die ersten Grundlagen dafür wurden in der Arbeitsgruppe am ZFT des Kombinates Robotron geschaffen. Der in der zweiten Hälfte der 70er Jahre entworfene Ansatz zur Verhaltensspezifikation konnte in den 80er Jahren an der TU Magdeburg weiter ausgebaut werden. Er führte in den 90er Jahren zur koalgebraischen Spezifikation von Prozesstypen. Diese Ideen wurden letztendlich mit der Spezifikationssprache COCASL für die allgemeine Nutzung verfügbar gemacht.

Literatur

[AKKB99] E. Astesiano, H.-J. Kreowski und B. Krieg-Brückner, Hrsg. *Algebraic Foundations of Systems Specification*. IFIP State–of–the–Art Reports. Springer, 1999.

[CMR99] M. Cerioli, T. Mossakowski und H. Reichel. From total equational to parial first order. In H.-J. Kreowski, E. Astesiano und B. Krieg-Brückner, Hrsg., *Algebraic Foundations of System Specifications*. Springer, 1999.

[GTW+75] Goguen J. A., Thatcher J. W., Wagner, E.G. und J.B. Wright. Initial algebra semantics. Bericht RC 5364, IBM, T.J. Watson Research Center, 1975.

[HKR80] Ulrich L. Hupbach, Heinz Kaphengst und Horst Reichel. Initial algebraic specifications of data types, parameterized data types, and algorithms. WIB 15, VEB Robotron, Zentrum für Forschung und Technik, 1980.

[Hup80] Ulrich L. Hupbach. Abstract Implementation of Abstract Data Types. In Piotr Dembinski, Hrsg., *Proc. Mathematical Foundations of Computer Science*, Seiten 291–304. Springer LNCS 88, 1980.

[KR71] Heinz Kaphengst und Horst Reichel. Algebraische Algorithmentheorie. Bericht, VEB Robotron, Zentrum für Forschung und Technik, 1971.

[KR72] Heinz Kaphengst und Horst Reichel. Operative Theorien und Kategorien von operativen Systemen. *Schriftenreihe des Zentralinstituts für Mathematik und Mechanik, Akademie–Verlag*, (Heft 16):41–56, 1972. In: Studien zur Algebra und ihren Anwendungen.

[KR76] Heinz Kaphengst und Horst Reichel. EPS – eine Sprache zum Entwurf komplexer Programmsysteme. WIB 8, VEB Robotron, Zentrum für Forschung und Technik, 1976.

[KR77] Heinz Kaphengst und Horst Reichel. Initial algebraic semantics for non context free languages. In *Proc. FCT'77*, number 56 in Lecture Notes in Computer Science, Berlin, 1977. Springer Verlag.

[MSRR06] Till Mossakowski, Lutz Schröder, Markus Roggenbach und Horst Reichel. Algebraic-co-algebraic specification in CoCASL. *Journal of Logic and Algebraic Programming*, 67(1-2):146–197, 2006.

[Rei80] H. Reichel. Initially restricting algebraic theories. In Piotr Dembinski, Hrsg., *Proc. Mathematical Foundations of Computer Science*, Seiten 504–515. Springer LNCS 88, 1980.

[Rei81] H. Reichel. Behavioural Equivalence – a unifying concept for initial and final specification methods. In M. Arato und L. Varga, Hrsg., *Proc. 3rd Hungarian Comp. Sci. Conf.*, Seiten 27–39, Budapest, 1981. Akademiai Kiado.

[Rei85a] H. Reichel. Behavioural Validity of Conditional Equations in Abstract Data Types. In *Proc. of the Vienne Conference, june 1984*, Contributrions to General Algebra 3, Seiten 301–324, Stuttgart, 1985. Verlag B.G. Teubner.

[Rei85b] H. Reichel. Initial Restrictions of Behaviour. In *Proc. IFIP Working Conference The Role of Abstract Models in Information Processing*. North Holland, 1985.

[Rei86] H. Reichel. Behavioural Program Specification. In D. Pitt et all, Hrsg., *Proc. Category Theory and Computer Science*, number 240 in Lecture Notes in Computer Science. Springer Verlag, 1986.

[Rei87] H. Reichel. *Initial Computability, Algebraic Specifications, and Partial Algebras*. Oxford Univ. Press, 1987.

[Rei95] Horst Reichel. An Approach to Object Semantics based on Terminal Co–Algebras. *Math. Struc. in Comp. Science*, 5:129–152, 1995.

[Rei99] Horst Reichel. Specification Semantics. In H.-J. Kreowski, E. Astesiano und B. Krieg-Brückner, Hrsg., *Algebraic Foundations of System Specifications*. Springer, 1999.

[Wec92] Wolfgang Wechler. *Universal Algebra for Computer Scientists*. EATCS Monographs on Theoretical Computer Science. Springer, April 1992.

Künstliche Intelligenz und automatische Sprachverarbeitung - ausgewählte Forschungen im Kombinat ROBOTRON

Frank Zänker

Am Weißiger Bach 147 a
01328 Dresden
frank.zaenker@t-online.de

Abstract: Der Autor beschreibt aus seiner Erinnerung die Grundlagenforschungen auf den Gebieten *Künstliche Intelligenz* und *Computerlinguistik* bei ROBOTRON. Er nennt die Bedingungen, Umstände und Veränderungen, unter denen diese Arbeiten stattfanden und wie er sie wahrgenommen hat, nennt Ziele, Leistungen und Ergebnisse und resümiert das Ende dieser Arbeiten 1989/90. Insofern ist der Aufsatz ein Beitrag zur Chronik, zur Industriegeschichte des VEB KOMBINAT ROBOTRON. Zum Schluss wagt der Autor eine Prognose, wie die Künstliche Intelligenz durch einen Paradigmenwechsel in ihrem Verständnis und ihren Methoden, durch ein anderes Modell der Intelligenz eine neue Perspektive erhalten könnte.

1 Die Ursprünge in Dresden 1965 bis 1969

Im Jahre 1965 war unter Leitung von Prof.N.J.LEHMANN das *Institut für maschinelle Rechentechnik* (IMR) als eine Einrichtung der *Deutschen Akademie der Wissenschaften zu Berlin* (DAW) in Dresden gegründet worden. Diese Gründung ging formal auf das "Programm von Maßnahmen zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR in den Jahren 1964 bis 1970" vom 3.7.1964 zurück. Bereits 1967 wurde das IMR aber in die *VVB Datenverarbeitung und Büromaschinen Erfurt* eingegliedert, und Prof. LEHMANN zog sich auf seine Forschungs- und Lehraufgaben an der TU Dresden und die Mitarbeit in nationalen Gremien zurück. Die Leitung des IMR wurde an seiner Stelle an Prof. ADLER übertragen.

Das Institut war in zwei Baracken auf der Franklinstraße 17 in Dresden untergebracht, es waren etwa 100 Mitarbeiter beschäftigt, und die Arbeitsthemen waren auf ausgewählte Aspekte künftig erwarteter Computertechnologien orientiert: physikalischen Untersuchungen zu neuen Speichermedien, Entwurf und Konzipierung neuer, effektiver Rechnerstrukturen sowohl für Digital- als auch Analog- und Hybridrechner, optische Zeichen-, Bild- und Mustererkennung, spezielle Architekturen (zellulare und Lernstrukturen), Verfahren, um die räumlichen Eigenschaften eines abgelichteten Objekts in der Reproduktion sichtbar zu machen (Holografie), Grundlagen der Softwaretechnologie wie vergleichende Untersuchungen von bestehenden universellen

Programmiersprachen (ALGOL 68, PL 1, SIMULA-67) einschließlich ihrer algebraischen Beschreibung und der Syntaxanalyse über die Entwicklung neuer Programmiersprachen bis hin zur Analyse und formalen Beschreibung von Lern- und Entscheidungsprozessen, u. a. automatisches Problemlösen. Die softwareorientierten Arbeiten waren im Fachbereich *mathematisch-kybernetische Grundlagen* zusammengefasst. Der andere, der hardwareorientierte Fachbereich hieß *physikalisch-technische Grundlagen.* Die gerätetechnische Basis der Arbeiten war der Digitalrechner D4a, programmiert wurde im Maschinencode.

Zu den Mitarbeitern und Führungskräften des IMR gehörten Wissenschaftler, die die Entwicklung des Digitalrechners vom D1 zum D4a unter Prof. LEHMANN begleitet und wesentlich mitgestaltet hatten, also v. a. Mathematiker, Physiker, Diplomingenieure. Die Arbeitsthemen und Forschungsziele wurden vorwiegend aus den Arbeitsgruppen entwickelt, waren aber inhaltlich und terminlich mit der Institutsleitung abgestimmt. Später kamen auch Biologen, Philosophen, Lehrer usw. als Mitarbeiter zum IMR. Das Studienfach *Informatik* kannte man damals in der DDR ebenso wenig wie den ausgebildeten Informatiker, der gängige russische Begriff *informatika* bezeichnete stattdessen die Lehre von der Information/Dokumentation im Bibliothekswesen und für Dokumentenverwaltungssysteme. Um diese Zeit begann sich das Arbeitsgebiet *automatisches Problemlösen*, in dem anfangs weniger als ein Dutzend Mitarbeiter des Instituts tätig waren, international als *Künstliche Intelligenz* (KI, auch *artificial intelligence*, AI) zu etablieren.

2 Von den Anfängen bei ROBOTRON ab 1969 bis 1980

Am 01.04.1969 wurde das *Großforschungszentrum* (GFZ) als die zentrale Entwicklungs- und Forschungseinrichtung des *VEB Kombinat Robotron* gegründet. Das IMR wurde in das GFZ übernommen und bildete dort als Fachgebiet E1 einen der anfänglich fünf Leistungsbereiche. An dem Wechsel in die volkseigene Industrie nahmen einige Fachkräfte des IMR schon nicht teil, die Fluktuation gerade der fähigsten wissenschaftlichen Führungskräfte und Mitarbeiter setzte sich auch in den folgenden Jahren fort. Die ehem. „wissenschaftlichen Mitarbeiter“ des IMR wurden Ende der 70er Jahre bei ROBOTRON, soweit noch dort tätig, in „Mitarbeiter F/E“ umbenannt.

Auch die beiden Fachbereiche des IMR wurden ins GFZ als fachliche Struktureinheiten übernommen. Im Fachbereich *mathematisch-kybernetische Grundlagen* wurden mit durchschnittlich etwa 15 Mitarbeitern die Forschungsarbeiten auf den Gebieten *Rechnerstrukturen*, *Automatisierung der Programmierung* (heute *Softwaretechnologie*), *Problemlösungsverfahren* (heute *Künstliche Intelligenz*) und *Lernstrukturen* erst einmal unverändert fortgeführt. Unter den Problemlösungsverfahren i. e. S. verstand man algebraische Formelmanipulation, Strategien des automatischen Problemlösens, automatischen Theorembeweis, Frage-Antwort-Systeme und allgemein die Verarbeitung der natürlicher Sprache, insbesondere die natürlichsprachige Kommunikation mit Computersystemen.

Es war die Zeit, als STEINBUCH die *Lernmatrix* (als besonderen Typ eines künstlichen neuronalen Netzes) konzipiert hatte [St63], die Publikationen von BOOTH [Bo67] oder MINSKY [Mi68] beschrieben den *state of the art* der „intelligenten" Maschinenleistungen, und mit Interesse wurden auch die quasi-schöpferischen Experimente von GUNZENHÄUSER oder MAX BENSE zur „künstlichen Kunst", die Geschichte der ENIAC-Symphonie oder das Dialogsystem ELIZA [We66] zur Kenntnis genommen. Es gab damals einen weit verbreiteten, nahezu grenzenlosen Glauben an die technische Entwicklung und die Möglichkeit, zukünftig immer komplexere intellektuelle Tätigkeiten auf Rechenautomaten zu übertragen. „Intelligenz" wurde mittels des *Turing-Tests* [Tu50] definiert. Die weitgespannten Hoffnungen wurden auch durch die großzügigen, vielfältigen Forschungsprojekte in den USA und Westeuropa und ihre ersten Erfolge genährt, und in den 80er Jahren gab das hoch ambitionierte *5th generation project* der japanischen Ministerien für Wirtschaft und Technologie den Erwartungen und Zukunftsvisionen einen weiteren Auftrieb [Gr89].

Die Forschungen zur KI bei ROBOTRON hatten von Anfang an immer eine ambivalente Stellung: Einmal war es allen einsichtig, dass es sich um Grundlagenforschung handelte, deren Nutzen sich vermutlich nicht in kürzester Zeit realisieren würde. Das Verhältnis der anderen Mitarbeiter von ROBOTRON war zu „den Forschern" deshalb eher freundlich-distanziert. Man war vielleicht auch besorgt, eine mögliche Zukunftschance zu verpassen, wenn man nicht zumindest mit einer kleinen Kapazität sich wissenschaftlich auf dem Laufenden hielt. Es gab auch keine Erfahrung, wie man mit einer Industrieforschung, die von zahlenmäßig wenigen Mitarbeitern betrieben wurde, in einem Kombinat mit tausenden Angestellten und klaren kurz- bis mittelfristigen wirtschaftlichen Zielvorgaben am besten umgehen sollte. Später, als die Erwartung wuchs, dass die KI in endlicher Zeit einen Beitrag zur Softwarestrategie und Softwareentwicklung bei ROBOTRON zu leisten habe, wurden die Arbeiten zur KI oft auch als „Vorlauf-" oder „angewandte Forschungen" interpretiert. Aus heutiger Sicht wäre vielleicht der Begriff „Verfolgungsforschung" zutreffender, da zwischen den Forschungsleistungen zumeist an westlichen Universitäten, ihrer Veröffentlichung, ihrem Bekanntwerden bei den KI-Mitarbeitern von ROBOTRON und den Impulsen, die davon wieder für die eigenen Arbeiten ausgingen, in der Regel eine größere Zeitspanne lag. (Es kursierte sogar der böse Ausdruck der „Nachforschung".) Literatur zur komplexen automatischen Verarbeitung der deutschen Sprache gab es in den Anfangsjahren ohnehin nur spärlich.

Die Vielfalt der Arbeiten zur KI bei ROBOTRON, so wie sie anfangs vom IMR übernommen worden waren, wurde bald auf die Verarbeitung der natürlichen Sprache, also speziell des Deutschen, und die Wissensverarbeitung durch Rechenanlagen konzentriert. Einmal war mit dem Personalbestand das gesamte KI-Spektrum ohnehin nicht abzudecken, nicht einmal die Fachliteratur konnte angemessen verfolgt werden. Zum anderen galten und gelten Sprachverstehen und Spracherzeugung als eine genuin menschlich-intellektuelle Leistung (*sprachliche Intelligenz*), für deren Modellierung eine Reihe von Wissensgebieten und Verfahren einzubeziehen sind: Ermittlung der Grundformen flektierte deutscher Wörter (*Lemmatisierung*), die Ermittlung von Struktur und Bedeutung (*Syntax* und *Semantik*) sprachlicher Äußerungen und ihre computergerechte interne Präsentation, Formulieren und Speichern axiomatischer

Grundtatsachen über Sprache und Diskurswelt (*Hintergrundwissen*) sowie geeignete deduktive Verfahren zur Ableitung „neuen Wissens“ (zur *Disambiguierung* von mehrdeutigen Äußerungen, als Antwort auf Fragen oder zur internen Revision des Hintergrundwissens), ggf. unter Einbeziehung geeigneter, auch nichtklassischer Logikkalküle, bis hin zu Verfahren der Generierung von Äußerungen aufgrund eines *Diskursverständnisses*. Dazu sind sprachwissenschaftliche Kenntnisse und Theorien heranzuziehen – von der Morphologie über die Syntax bis zur Semantik aus der klassischen Linguistik wie auch die pragmatische Intention von Äußerungen. Text- und Sprechakttheorie sowie Kommunikations- und Kognitionswissenschaft sind dabei ebenso bedeutsam, wie auch Fragen aus der Erkenntnistheorie und der Zeichentheorie (*Semiotik*) zum wissenschaftlichen Hintergrund gehören. Und für alle diese Prozesse, Beschreibungs- und Darstellungsebenen waren effektive programmtechnische Mittel zu entwickeln: effektive morphologische Zerlegungen, schnelle Wörterbuchzugriffe, Mittel für Aufbau und Pflege von Wörterbüchern, *Parsing*-Verfahren, kompakte Speicherungsformen für Bedeutungen, automatisches Schlussfolgern, Generierungsalgorithmen.

Für das Arbeitsgebiet *Verarbeitung natürlicher Sprache durch Computer* begann sich international der Begriff *Computerlinguistik* (CL) durchzusetzen. In der Computerlinguistik wird untersucht, wie natürliche Sprache mit Hilfe des Computers algorithmisch verarbeitet werden kann. Sie ist ein Teilbereich der Künstlichen Intelligenz und gleichzeitig die Schnittstelle zwischen Sprachwissenschaft und Informatik.

Die 60er und 70er Jahre waren für diese Arbeiten eine günstige, eine fruchtbare Zeit, als einmal CHOMSKY hilfreiche formale Grammatikformalismen vorgelegt hatte [Ch56], [Ch65], aber auch die geisteswissenschaftliche Strömung des Strukturalismus hatte sich weit verbreitet und zu interessanten neuen, theoretischen Einsichten auch über die Sprache geführt hatten [Ca45].

Die weltweiten Forschung- und Entwicklungsarbeiten zur automatischen Verarbeitung natürlicher Sprache umfassten damals schon ein sehr breites Spektrum - von akustischer Spracherkennung und dem *speech understanding* über natürlichsprachige Dialog- und Auskunftssysteme bis zur automatischen Übersetzung. Mit Blick auf die Aufgabenstellungen bei ROBOTRON wurde die Arbeiten speziell auf Auskunftssysteme, Frage-Antwort-Systeme bzw. die natürlichsprachige Abfrage von gespeicherten Datenbankinhalten in Rechnern beschränkt. Das Forschungsthema hieß deshalb auch in der Anfangszeit *POS-Frage-Antwort-System*. Die Arbeiten zielten auf ein System ab, über das man Fragen in natürlicher Sprache über Aufbau und Arbeitsweise von problemorientierten Softwarepaketen (POS) stellen konnte und darauf – ggf. kurze, schlagwortartige - Antworten erhielt, also sozusagen ein natürlichsprachig getriggertes *information retrieval*. Das Programmpaket, mit dem das realisiert werden sollte, war das „Frage-Antwort-System 1975“ (FAS-75).

Die Rechner, die für diese Arbeiten zur Verfügung standen, waren vergleichsweise leistungsfähige Großrechner aus der ROBOTRON-Baureihe R300, bald aber schon der R40. Die Programmierung und die Eingabe von Sprachäußerungen erfolgten mittels Lochkarten, der Output waren gedruckte Listen. Programmiert wurde in der Sprache LISP, die 1958 erstmals spezifiziert wurde und am *Massachusetts Institute of Tech-*

nology in Anlehnung an den Lambda-Kalkül entwickelt worden war. Bei ROBOTRON war 1970 die deutschlandweit erste Implementation eines LISP-Interpreters vorgenommen worden, die später noch um einen Compiler erweitert wurde [St77].

3 Forschungsbedingungen, Kommunikation und Kooperation

Die Fachkollegen und Spezialisten der KI, insbesondere auch aus den modernen Sprachwissenschaften, und die auf Teilgebieten der Computerlinguistik Tätigen innerhalb der DDR kannten sich untereinander alle und unterhielten über Jahre gute fachliche Kontakte. So bestand mit dem *Zentralinstitut für Kybernetik und Informationsprozesse* (ZKI) an der AdW über lange Zeit ein Kooperationsvertrag, der schließlich zur gemeinsamen Implementierung des *FAS-80*, der Nachfolgeversion von *FAS-75*, führte [HB83]. Es gab einen regen Gedankenaustausch mit Vertretern der *Humboldt-Universität* (Sektion Psychologie, Prof. KLIX) und viele Gespräche an der *Karl-Marx-Universität Leipzig, Sektion Angewandte Sprachwissenschaften* (Prof. CONRAD). Auch zum *Zentralinstitut für Sprachwissenschaften* (ZISW der AdW, Prof. KUNZE) gab es ausgezeichnete wissenschaftliche Beziehungen. Besondere Verbindungen bestanden weiterhin zur *Technischen Universität Dresden* (TUD)*, Sektion Informationstechnik, Bereich Akustik* (Prof. TESCHNER). Diese Wissenschaftler arbeiteten bereits damals erfolgreich auf dem Gebiet der akustischen Spracherkennung. Es bestand stillschweigend Konsens darüber, dass in der Zukunft ein Frage-Antwort-System natürlich nur für gesprochene Sprache eine wirkliche Nutzungs- und Marktchance haben würde, und wenn alle anderen Probleme beim FAS gelöst wären, wäre „nur noch" ein *speech interface* zu integrieren. Umgekehrt war es den Kollegen an der TUD auch immer bewusst, dass für das Verstehen gesprochener Sprache letztlich all die syntaktisch-semantisch-pragmatischen Prozesse notwendig wären, zu deren ganzheitlicher Modellierung beispielsweise eben ein FAS diente. Allen solchen externen Kontakten stand man bei ROBOTRON durchaus interessiert und aufgeschlossen gegenüber. Und da Fachgespräche innerhalb des Betriebes, aber außerhalb der KI-Arbeitsgruppe, mangels interessierter oder geeigneter Partner praktisch kaum stattfanden, war dieser Austausch mit externen Partnern natürlich besonders wichtig.

1989 wurde von ROBOTRON und der TUD sogar ein *Hochschul-Industrie-Komplex Künstliche Intelligenz* gegründet, der sich allerdings wegen der bald danach erfolgten Auflösung von ROBOTRON auf die praktische Arbeit überhaupt nicht mehr auswirkte.

Direkte beruflich-dienstliche Kontakte zu Fachkollegen und Forschungseinrichtungen im westlichen Ausland waren natürlich zu allen Zeiten völlig undenkbar. Selbst persönliche Kontakte – brieflich oder per Telefon – waren nicht erlaubt und wurden strengstens geahndet. Dafür war man bei ROBOTRON um Literaturbeschaffung durchaus bemüht: Periodika wie die *Communications of the ACM*, das *Journal of the ACM* oder die Zeitschrift *Artificial Intelligence* wurden – wenn auch verspätet, so doch einigermaßen regelmäßig und zuverlässig – zur Verfügung gestellt, ebenso die Proceedings einschlägiger internationaler Konferenzen. Man konnte über die Bibliothek des GFZ/ZFT Monographien oder Sammelwerke von einschlägigen Verlagen wie Academic Press, MIT Press, Elsevier, Addison-Wesley, Springer-Verlag, Oldenbourg-Verlag, usw.

bestellen und erhielt zumindest die vor geraumer Zeit erschienene Werke oder Beiträge. Nicht beschaffbar waren dagegen neuere Forschungsberichte, wie sie zeitnah und in großer Anzahl an den westlichen, auch westdeutschen Universitäten publiziert wurden. Bereitgestellt wurden die ausgewählten Beiträge meist als Kopien, Microfiches oder Rollfilme. Man war also – nicht zeitgleich, nicht vollständig, nicht sehr bequem, so aber doch wenigstens im Prinzip – über die Forschungen und Entwicklungen in anderen Ländern und Einrichtungen einigermaßen informiert, und man durfte sich einer internationalen *community* von Computerlinguisten zugehörig fühlen.

Die Bedingungen für die Arbeit am Rechner, zuletzt dem R55, war für das wissenschaftliche Personal eher unterdurchschnittlich: Die KI-Mitarbeiter hatten nur in engen Zeitfenstern Zugang zur online-Programmierung im TSO-Regime, bis zu den letzten Tagen von ROBOTRON wurde auch noch die Stapelverarbeitung mittels Lochkarten intensiv genutzt. Rechenzeiten mit persönlicher Anwesenheit und dadurch eine semi-interaktiven Arbeit am Großrechner waren ebenso knapp bemessen und lagen oft in den späten Abend- und Nachtstunden. Selbst für den ersten importierten PC, den die KI-Mitarbeiter ab September 1987 mit nutzen durften, gab es ein strenges Rechenzeitregime und Nutzungszeiten nach 20 Uhr. Vorrang vor aller Forschungsarbeit hatten natürlich die Entwicklung und Testung kommerzieller Programmsysteme für PCs.

Die Forschungsarbeiten zu KI und CL unterlagen – wie alle anderen Arbeiten beim GFZ/ZFT/ROBOTRON-PROJEKT auch –, solange es eine selbständige Struktureinheit „Forschung“ gab, einem strengen inhaltlichen und zeitlichem Controlling: Am Ende einer Themenlaufzeit waren formale Abschlußberichte anzufertigen, es gab eine Verteidigung der Ergebnisse, es wurden Lob und Kritik ausgeteilt und die Hauptlinien der nächsten Arbeiten bestätigt. Die vorzulegenden Ergebnisse bestanden in der praktischen Präsentation der bis dahin erreichten Systemlösung, dem Themenabschlußbericht und allen während der Themenlaufzeit entstandenen Publikationen. Viele davon waren ROBOTRON-interne Forschungsberichte. In aller Regel waren die Arbeiten im Kontrollplan des Kombinates (sog. „Z-Themen“) enthalten, es gab sogar einen Fall, in dem das Ministerium für Elektrotechnik und Elektronik in Berlin die Prüfung und Abnahme beaufsichtigte („ZZ-Thema“).

4 Die Jahre 1980 bis 1990

Mit dem 1.1.1973 wurde das *Großforschungszentrum* (GFZ) in das *Zentrum für Forschung und Technik* (ZFT) überführt, das FG E1 wurde später aufgelöst, die softwareorientierten Forschungen wurden per 1.4.1978 in der Hauptabteilung E4 zu einer Abteilung E4F zusammengefasst, und seit 1.1.1984 hieß das ZFT dann *VEB Robotron Projekt*. Welche Gründe zu diesen Änderungen führten und welche Konsequenzen das auf die Strategie und Aufgaben des Kombinates ROBOTRON hatten, ist bei [JM05] ausführlich nachzulesen. Es wechselten teilweise die Führungskräfte, es gab Änderungen und Umbenennungen in den Strukturen, man benutzte die Anlässe, um Arbeitsgebiete und Verantwortungsbereiche neu zu organisieren, für die Arbeiten auf dem Gebiete der KI/CL hatte das aber praktisch eher wenig Folgen.

Eine Zäsur für die KI-Forschung bei ROBOTRON erfolgte jedoch im Spätherbst 1980: Aus der Struktur E4F wurden die Mitarbeiter herausgelöst, die mit KI, CL und informationslogischen Systemen befasst waren. Sie wurden als Arbeitsgruppe E4325 der Abteilung 432 zugeordnet, in der das Informationsrecherchesystem IRS AIDOS entwickelt und weiterentwickelt wurde. Die KI-Forschung fand damit nicht mehr wie bis dahin in einem „geschützten" Forschungsbereich statt, sondern sie wurde stringent auf AIDOS ausgerichtet, was aber von den meisten durchaus positiv aufgenommen wurde. Auf der Grundlage der bisherigen Kenntnisse, Erfahrungen und prototypischen Lösungen FAS-75 und FAS-80 wurde das Konzept entwickelt, eine Schnittstelle, ein *natürlichsprachiges Interface* (NLI) zu AIDOS zu schaffen. Damit sollten natürlichsprachige Abfragen zu großen Informationsbeständen in Form von „Sekundärdokumenten" [Lö08], die per AIDOS gespeichert und verwaltet wurden, also z.B. bibliographische Daten und Fakten, möglich gemacht und beantwortet werden. Wie das erreicht werden sollte, ist in [He91, S. 264 ff.] sowie im Beitrag von NAUMANN u.a. [Na08] dargestellt. Mit kleiner Kapazität wurden auch noch die Themen *Vorlauf Expertensysteme* und *Wissensverarbeitung* weitergeführt. In der Spätphase der Forschungsarbeiten erfolgte eine Fokussierung auf die PC-Anwendung, nämlich auf ein NLI für das System AIDOS/M. Mit dem 9.11.1989 und den nachfolgenden politischen Ereignissen änderten sich auch radikal die wirtschaftlichen Prioritäten, die Arbeiten am NLI-AIDOS wurden im Verlaufe des Jahres 1990 ganz eingestellt. Immerhin war das System konzeptionell zu Ende durchdacht, es wäre durch eine intensive Programmierung zu komplettieren gewesen.

Es gab weitere Leistungen, die parallel zum NLI in der Spätphase erbracht wurden: Einmal wurden die Untersuchungen zu Stand und Perspektiven von Expertensystemen weitergeführt [Bö86], und es wurde mit einem PROLOG-Verarbeitungssystem experimentiert. Dann wurde mit AIDOS/M eine umfangreiche *Datenbank KI-Literatur* aufgebaut, die als experimentelle Datenbasis für das NLI eingesetzt werden sollte, die aber auch einen „Wert an sich" darstellte. Zum Weiteren war ein Algorithmus zur Rekonstruktion von *Ellipsen* gefunden worden. Im natürlichsprachigen Dialog werden nur selten von den Diskursteilnehmern grammatisch vollständige Sätze ausgetauscht, sondern meist nur unvollständige Sätze mit erheblichen, aber regelgeleiteten Auslassungen, sog. „Ellipsen". Für ein „Verstehen" solcher elliptischer Sätze durch Rechenautomaten und seine Teilnahme als Dialogpartner sind aber grammatisch korrekte Sätze erforderlich. Wie man elliptische Sätze anhand des vorangegangenen Diskurses zu korrekten Sätzen rekonstruieren kann, das sollte dieser Algorithmus leisten. In der CL war ein solches Verfahren bis zu diesem Zeitpunkt sonst nicht bekannt. Und schließlich waren auf Anforderung des Herausgebers von den KI-Mitarbeitern eine Reihe von Beiträgen verfasst worden, die in einem *Lexikon der Künstlichen Intelligenz* des *Akademie-Verlages Berlin* (Hrsg. W. Kolbe) erscheinen sollten.

Im Spätherbst 1990 löste sich dann die KI-Arbeitsgruppe aus zuletzt 8 Mitarbeitern vollständig auf – durch Vorruhestand, Zuweisung anderer Aufgaben bei einer Nachfolgeeinrichtung SRS, Tätigkeiten für die SAP in Walldorf und bei SNI, München, usw. gingen das kollektiv erworbene know-how und die Ergebnisse einer mehr als 20jährigen wissenschaftlichen Arbeit großteils verloren. Ein Mitglied der KI-Arbeitsgruppe wurde noch in einem Eilverfahren zum Sprecher der GI, Sektion

Computerlinguistik, bestellt, um ihn zu legitimieren, Gespräche zur Rettung und Weiterführung der CL-Kompetenzen und -Arbeiten in Sachsen zu führen. Aber weder der damalige Wissenschaftsminister des Freistaates Sachsen, Prof. MEYER, noch der Gründungsdekan für die neue Volluniversität in Dresden, Prof. OBERREUTER, waren für diese Vorstellungen zugänglich. Auch dass danach noch drei Semester eine Gastvorlesung *Grundlagen der Computerlinguistik* im Rahmen des *studium generale* an der TUD, Fakultät Informationsverarbeitung, gehalten wurde, konnte daran nichts mehr ändern. Bis Mitte 1991 wurde im Auftrage von SNI noch das Thema *Expertensysteme* weiter bearbeitet, bis diese Arbeiten 6/1991 ebenfalls endgültig eingestellt wurden.

5 KI – ein Blick zurück und einer voraus

Man mag sich vielleicht fragen, was mit den bis heute fast 50-jährigen weltweiten Forschungsarbeiten zur KI und CL erreicht wurde, was davon Bestand hat, und wie eine Zukunft auf diesem Gebiet aussehen kann.

Mit *Intelligenz* bezeichnet man in der differentiellen und kognitiven Psychologie die Fähigkeit zum Erkennen von Zusammenhängen und zum Finden optimaler Problemlösungen. Intelligenz ist erst einmal primär die menschliche Fähigkeit, etwas zu verstehen, zu abstrahieren, Probleme zu lösen, Wissen und Sprache zu verwenden. Der Begriff *Künstliche Intelligenz* wurde erstmals 1955 in der Vorbereitung einer Konferenz durch McCHARTY geprägt. Diese später berühmt gewordene *Dartmouth Conference* begann am 13. Juli 1956 und war von JOHN McCHARTY, von MARVIN MINSKY, NATHAN ROCHESTER und CLAUDE SHANNON organisiert worden.

Im Begriff *Künstliche Intelligenz* spiegelt sich die schon in der europäischen Aufklärung entstandene Vorstellung vom „Menschen als Maschine“ wider, dessen Nachahmung sich die orthodoxen Vertreter einer *starken KI* zum Ziel setzen: eine Intelligenz zu erschaffen, die wie der Mensch kreativ nachdenken und Probleme lösen kann und die sich durch eine Form von Bewusstsein beziehungsweise Selbstbewusstsein sowie Emotionen auszeichnet. Das mag auch der Hintergrund für eine überlieferte frühe Prognose von MINSKY gewesen sein: „Wenn wir etwas Glück haben, werden uns die Roboter als Haustiere halten“. Solche Ziele der starken KI sind auch nach Jahrzehnten der Forschung weithin unerreicht und visionär, zumindest solange, wie man am derzeitigen Paradigma künstlich-intelligenter Systeme festhält.

Im Gegensatz dazu geht es den Vertretern einer *schwachen KI* um die Lösung konkreter Anwendungsprobleme, für die nach allgemeinem Verständnis eine bestimmte Form von „Intelligenz“ notwendig zu sein scheint. Es geht hier also eher um die Simulation intelligenten Verhaltens mit Mitteln der Mathematik und der Informatik, es geht nicht um Schaffung von Bewusstsein oder um tieferes Verständnis von Intelligenz. Dieser konstruktivistische Ansatz begnügt sich damit, intelligente Fähigkeiten von vorliegenden Anwendungen vom Vorhandensein gewisser Merkmale abhängig zu machen:

1. die Fähigkeit zur Verarbeitung beliebiger Symbole (nicht nur von Zahlen);
2. der Aufbau eines internen Modells der äußeren Welt;

3. die Fähigkeit zu einer zielorientierten Anwendung von Wissen.

(Es gibt darüber hinaus noch 9 weitere Merkmale, deren Vorhandensein eine jeweils höhere Art von Intelligenz ausmacht.) Für beide Lesarten gilt aber, dass die KI kein abgeschlossenes Forschungsgebiet darstellt. Vielmehr werden Techniken aus verschiedensten Disziplinen verwendet, ohne dass diese eine wissenschaftstheoretische Verbindung miteinander haben müssen.

Folgt man dieser zweiten, schwächeren Lesart, dann sind Erkenntnisse der Künstlichen Intelligenz mit der Zeit oft in die anderen Gebiete der Informatik übergegangen: Sobald ein Problem gut genug verstanden war, hat sich die KI neuen Forschungsaufgaben zugewandt. Zahlreiche heutige Anwendungen wurden auf der Grundlage von Techniken entwickelt, die einst Forschungsgebiete der KI waren oder es noch sind, z.B.:

- wissensbasierte und Expertensysteme;
- künstliche neuronale Netze;
- optische Zeichenerkennung für gedruckte Texte;
- Handschrifterkennung in PDAs;
- Computeralgebra-Systeme;
- Computer-Vision-Systeme;
- Algorithmen für Computerspiele;
- Gruppensimulationen für Sicherheitsplanung oder 3D-Computergrafik ;
- Entwicklung neuer, nichtprozeduraler Programmiersprachen und –stile: funktionale Programmierung (LISP), logische Programmierung (PROLOG), Grundlagen objektorientierter Sprachen (SIMULA, SMALLTALK).

Speziell in der Computerlinguistik haben solche Anwendungen ihren Ursprung wie z.B.:

- Textverarbeitung: Prüfung auf orthografische und grammatische Korrektheit;
- intelligente Autorensystemen (z.B. Finden eines treffenden Ausdrucks oder der richtigen Terminologie);
- Suche nach Textstellen anhand semantischer Kriterien, information retrieval, Suchmaschinen;
- interaktive oder vollautomatische Übersetzung;
- Erkennen gesprochener Sprache (Wörter, Texte), Sprachsynthese;
- Algorithmen des *data mining;*
- Verfahren des *content management* und der *business intelligence*;
- *summarizing*, Informationsabstraktion, Erstellen von *abstracts*;
- automatische Schlagwortvergabe für Texte oder andere sprachliche Informationen.

Die Liste solcher Ergebnisse und Anwendungen wird sich zukünftig sicher weiter verlängern, wenn man KI in der schwachen Lesart akzeptiert.

Man kann sich auch die Frage zu stellen, wie denn die KI-Forschungen bei ROBOTRON aus heutiger Sicht zu bewerten sind. ROBOTRON hat sich bewusst drei Jahrzehnte Forschungsarbeiten in kleinem Umfang geleistet, und zwar nicht nur die KI-Forschung.

Damit bestand einmal die Gewähr, an Ergebnissen partizipieren zu können, deren Erfolg schwer vorauszusagen war, aber im positiven Falle einen Qualitätssprung bzgl. des Einsatzes von Rechnern bedeutet hätte. Damit wurde andererseits auch die Reputation des Kombinates in der Öffentlichkeit außerordentlich gefördert, denn die in der Forschung tätigen Mitarbeiter hatten auf ihren Arbeitsgebieten einen festen, anerkannten Platz sowohl in der ostdeutschen Wissenschaftslandschaft, wie sie auch in diesen zwei Jahrzehnten mit vielen Publikationen und öffentlichem Auftreten an der sächsischen Wissenschaftsgeschichte mitgeschrieben haben. Das wissenschaftliche Niveau dieser Arbeiten wird beispielsweise auch dadurch belegt, dass aus diesem Kreis der Mitarbeiter drei von ihnen später auf einen Lehrstuhl berufen und fünf Dissertationen erfolgreich verteidigt wurden. Die relativ langen Themenlaufzeiten und der scheinbar zögerliche Fortschritt der KI-Forschungsarbeiten bei ROBOTRON waren v. a. dem Umstand geschuldet, dass hier praktisch keinerlei studentischen Hilfskräfte zur Verfügung standen, dass Moduln, Programmpakete oder Softwarewerkzeuge bestenfalls aus der DDR hätten hinzu erworben werden können, die es aber eben dort gar nicht gab, so dass alles von Grund auf „mit der Hand am Arm“ programmiert werden musste, und dass die Informationsmöglichkeiten besonders in den späteren Phasen der KI- und CL-Forschung immer mehr eingeschränkt waren. Das NLI-AIDOS wäre vielleicht nach etwa einem Jahr vollständig praktisch demonstrierbar gewesen, dann hätten sich Anpassung, Praxiseinsatz, Weiterentwicklung anschließen können. Infolge der Vorgänge nach dem Oktober 1989 blieben von all dem aber nur Kenntnisse, Dokumentationen und Reminiszenzen – ein Schicksal, das leider auch allen anderen Software-Produkten von ROBOTRON wie AIDOS, DBSR, REDABAS, … zuteil wurde. Dauerhafte wissenschaftliche Spitzenleistungen waren in einem Umfeld, das von Wirtschaftsembargo, Devisenmangel, Misstrauen und Geheimhaltung, Abgrenzung und kaltem Krieg geprägt war, letztlich nicht ernsthaft zu erwarten gewesen.

Wie kann man sich aber eine Zukunft der intelligenten Maschinen, eine Künstliche Intelligenz auf höherem Niveau denken? Ist ein qualitativer Sprung dadurch möglich, dass man das menschliche Gehirn und seine Leistungen besser nachbildet, dass man einen Paradigmenwechsel vornimmt und eine neue Hardwarearchitektur dafür entwickelt? Die neuronalen Netze der 80er Jahre und die dafür aufgestellten wissenschaftlichen Modelle der KI waren jedenfalls nicht geeignet, um wirklich intelligente Maschinen zu bauen. Technisch ist aber inzwischen ein Stand erreicht, wo der Rechner mühelos manche Leistungen des menschlichen Gehirns übernehmen könnte: Neuronen arbeiten im Millisekundenbereich, Siliziumbauteile operieren bereits heute im Nanosekundenbereich und werden immer schneller. Ein „Verstand“ auf Si-Basis wäre millionenfach schneller als das menschliche Gehirn. Der menschliche Cortex weist etwa 32 Billionen Synapsen auf. Wenn man für jede Synapse zwei Bits braucht und jedes Byte 8 Bits hat, braucht man für ein Gedächtnis mit menschlicher Kapazität etwa 8 Billionen Bytes. Die Festplatte eines heutigen PC hat eine Kapazität im 100-Gigabyte-Bereich, wäre also nur noch etwa um den Faktor 10^2 zu klein. Die theoretisch erforderliche Speicherplatzgröße ist wenn nicht schon heute, so doch ganz bestimmt in wenigen Jahren erreichbar. Und man muss auch nicht den gesamten Cortex nachbauen, bei einem „weichen“ Verständnis von KI genügt sicher viel weniger. Ein Problem scheint allerdings die Konnektivität zu sein: Menschliche Gehirne verfügen über große Mengen subcorticaler weißer Substanz, die aus Millionen von Axonen besteht, die kreuz

und quer verlaufen und die verschiedenen Regionen der corticalen Hierarchie miteinander verbinden. Diese massive Parallelverschaltung ist mit traditionellen Si-Herstellungstechniken nur schwer, vielleicht überhaupt nicht zu erreichen. Eine Lösung könnte darin bestehen, dass – wie beim Telefonnetz – auf einem Draht simultan mehrere Signale übertragen werden.

Technisch scheint also eine Simulation des menschlichen Gehirns denkbar, jedenfalls in absehbarer Zeit. Auch die Feinstruktur und Hierarchie der corticalen Regionen und deren Austausch von Signalen ist durch die Ergebnisse der Neurowissenschaften offenbar hinreichend aufgehellt. Bereits seit dem vergangenen Jahrhundert weiß man ziemlich genau, dass die unterschiedlichen kognitiven Funktionen des Gehirns in unterschiedlichen Cortexregionen angesiedelt sind, und wo sich diese Regionen befinden. Diese funktionellen Areale sind in einer verzweigten Hierarchie angeordnet. Ankommende Außensignale gelangen als individuelle Signalfolgen an die entsprechenden untersten, primär sensorische Areale, und die untergeordneten senden mittels eines bestimmten neuronalen Verknüpfungsmusters die erkannten Signalfolgen aus sich und assoziierten Arealen an übergeordnete Areale, die mittels anderer Verknüpfungsmuster ein feedback-Signal zurücksenden. Alle Leistungen des Gehirns beruhen allein auf dem Empfangen, dem Speichern, Weitersenden und Verifizieren solcher Signalfolgen oder -muster, die, je höher in der Verarbeitungshierarchie hoch gereicht und verarbeitet, desto spezialisierter und abstrakter werden. In der obersten Region des entwicklungsgeschichtlich jungen Neocortex, dem eigentlichen Zentrum der Intelligenz, entsprechen die ankommenden Muster unseren Begriffen, die als solche gespeichert werden und Grundlage des begrifflichen Denkens sind.

Was bisher fehlte, war eine schlüssige Theorie darüber, was die Intelligenz in einem solchen System des Gehirns ausmacht. Dafür haben HAWKINS und BLAKESLEE nun einen interessanten Vorschlag entwickelt – das *Gedächtnis-Vorhersage-Modell* [HB04]. Intelligentes Verhalten beruht auf Erfahrung, Wissen über die Umwelt i. w. S. und zweckmäßiges Verhalten. Aufgrund seiner riesigen Gedächtniskapazität kann nach HAWKINS der Cortex anhand eines neu ankommenden Signalflusses ständig voraussagen, welche äußeren Informationen als nächstes höchstwahrscheinlich zu erwarten sind, was man sehen, hören, tasten wird. Und das geschieht größtenteils unbewusst. Aber diese Vorhersagen machen unsere Gedanken aus, und zusammen mit dem motorischen Input konstituieren sie unsere Wahrnehmung. Und diese Kette von *elementarer Wahrnehmung über Sensoren - Abstraktion über mehrere Ebenen hinweg bis zur Begriffsbildung - Speichern von invarianten Mustern auf allen Ebenen – autoassoziatives Wiedererkennen/Identifizieren neu ankommender Signalfolgen auf den verschiedenen Niveaus - Erwartung/Vorhersage der nächsten Signalfolgen aufgrund früher gespeicherte Muster - zweckmäßiges Verhalten aufgrund von Erfahrung oder Zielsetzung* – gerade diese Abfolge macht nach den Autoren die menschliche Intelligenz aus. Also nicht Verhalten (wie etwa noch bei TURING), sondern Erwartung, Vorhersage der nächsten Wahrnehmungen aufgrund von Wissen und Erfahrung sei das Kriterium für Intelligenz. Intelligente Maschinen, die nach diesem Paradigma und mit einer anderen Hardwarearchitektur arbeiten, könnten damit vielleicht menschlichen Intelligenzleistungen deutlich näher kommen.

Literaturverzeichnis

[Al95] Allen, J.: Natural Language Understanding. The Benjamin/Cummings Publishing Company, Inc., Redwood City/CA, 1995

[Bo67] Booth, A.D. (ed.): Machine translation. John Wiley & Sons, New York, 1967

[Bö86] Böttger, H.: Expertensysteme in der 4. und 5. Rechnergeneration. In: .Roth, M. (ed.): Beiträge zur Mikrocomputertechnik. Verlag Technik, Berlin, 1986.

[Br06] Brosch; H.-J.: Informatik-Grundlagenforschung im Kombinat ROBOTRON – ein Erfahrungsbericht. In [NS06], S. 501 ff.

[Ca45] Cassirer, E.: Structuralism in modern linguistics, in: Word 1/2 (1945), S. 99ff.

[Gr84] Groß, B.: Japans fünfte Computer-Generation und die sozial-ökonomische Dimension der Informatik. Schriftenreihe Dokumentation/Information, 76 (1984), TH Ilmenau

[Ha00] Hausser, R: Grundlagen der Computerlinguistik, Mensch-Maschine- Kommunikation in natürlicher Sprache. Springer, Berlin, New York, 2000

[HB04] Hawkins, J.; Blakeslee, S.: On intelligence. Henry Holt & Co., New York, 2004

[HB83] Helbig, H.; Böttger, H.; String, P.; Wicas, R.; Zänker, F.; Koch D.; Heicking, W.: FAS-80 – Ein natürlichsprachiges Auskunftssystem. Wiss.Inf.u.Berichte, Nr. 18, Robotron ZFT, 1983

[He91] Helbig, H.: Künstliche Intelligenz und automatische Wissensverarbeitung. Verlag Technik, Berlin, 1991

[JM05] Junge, S., Merkel, G. (eds.): Sammlung von Beiträgen zur Geschichte der Zentralen Forschungs- und Entwicklungseinrichtung des VEB Kombinat Robotron., 2005

[Ju00] Jurafsky, D.; Martin, J.H.: Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, New Jersey, 2000

[Lö08] Löhr, H.: Die Verarbeitung unformatierter Daten am Beispiel der Beschreibung von wissenschaftlich-technischen Dokumenten. – in diesem Band

[Mi68] Minsky, M.(ed.): Semantic information processing. MIT Press, Cambridge/Mass., 1968

[Na08] Naumann, P.: Das Robotron-Informationsrecherchesystem (IRS) AIDOS: Software-Technologie von gestern? - in diesem Band

[NS06] Naumann, F.; Schade, G. (eds.): Informatik in der DDR – eine Bilanz. Reihe "Lecture Notes in Informatics", Vol. T-1, Köllen Druck + Verlag GmbH, Bonn, 2006

[St63] Steinbuch, K.: Automat und Mensch. Springer-Verlag, Berlin, 1963

[St77] Stoyan, H.: LISP-Programmierhandbuch. Akademie-Verlag, Berlin, 1977

[Tu50] Turing, A.: Computing Machinery and Intelligence. In "Mind", vol. LIX, no. 236, October 1950, S. 433–460

[We66] Weizenbaum, J.: „ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine"; in "Communications of the Association for Computing Machinery" 9 (1966), S. 36–45

[Wi08] Artikel "Computerlinguistik", "ENIAC", "Intelligenz", "Künstliche Intelligenz", "Strukturalismus" in: Wikipedia, Die freie Enzyklopädie. Abgerufen am 24.1.2008, 15:57 UTC

[Zä 81] Zänker, F.: Dialogrecherche in natürlicher Sprache? In "Schriftenreihe des Instituts INER", Heft 54, S. 76 ff., TH Imenau, 1981

[Zä87] Zänker, F.: Computersprachen und Sprachverarbeitungssysteme – Schritte auf dem Weg zur Künstlichen Intelligenz. In "Mikroelektronik und Künstliche Intelligenz", S. 62 – 86, Akademie-Verlag, Berlin 1987

[Zä89] Zänker, F.: Erste Ergebnisse mit einem NLI für AIDOS/M". In "Informatik", Heft 5/89, S. 171-173, Berlin 1989

Rechnernetzforschung und –entwicklung an der TU Dresden

Helmut Löffler

An der Rysselkuppe 2a
01326 Dresden
h.loeffler@online.de

Abstract: Seit Beginn der siebziger Jahre war die TU Dresden ein Zentrum der Rechnernetzforschung in der DDR. Die entsprechenden Arbeiten konzentrierten sich auf Grundlagenforschung, auf die Mitwirkung am Aufbau und der Nutzung des Rechnernetzes DELTA sowie auf die Schaffung und Anwendung lokaler Netze in Lehre, Forschung und industrieller Praxis. Hierbei spielte das LOTUNET-Konzept eine herausragende Rolle.

1 Historischer Überblick

An der TUD erfolgte Rechnernetzforschung und -entwicklung auf folgenden Gebieten:

- Grundlagenforschung zur Analyse und Synthese von Rechnernetzen seit 1971
- Mitwirkung am Rechnernetz DELTA (Einbindung Arbeitsrechner und netzgerechtes Terminalsystem; beginnend etwa 1972)
- LAN-Forschung, -Entwicklung und -Praxisüberführung (seit 1980)
- Management von Rechnernetzen (etwa ab 1980).

Darüber hinaus gab es praxisorientierte Arbeiten zur Kopplung von EDVA im Nahbereich, z. B. Kopplung BESM-6/ESER 1020/22 (am Rechenzentrum) und ESER 1040/KRS 4200 (Sektion 09).

2 Grundlagenforschung zur Analyse und Synthese von Rechnernetzen

Diese Arbeiten orientierten auf die folgenden Schwerpunkte:

- Grundlagenforschung zur Analyse und Synthese von Rechnernetzen (seit 1971), [Lö76a], [Lö77a] [Lö77b], [HL77], [Lö82], [Lö84];

- Berechnung der Verweilzeit von Aufträgen im Rechnernetz und Schaffung von Werkzeugen zum Entwurf und zur Optimierung von Paketvermittlungsnetzen [Lö76b],[JL77], [HL77], [Lö81], [LM81], [LM82];

- Flusssteuerung in Rechnernetzen [Ho84],

- Gestaltung der Mehrschichtenstruktur von Rechnernetzen [Lö85], [Me85], [Sa84], [Zs89];

- Berechnung der Reaktions- und Verweilzeit von Aufträgen, die über netzgerechte Terminalsysteme dem Rechnernetz übergeben werden [IL79], [Oe80];

- Protocol-Engineering [Kö87] [Kö90];

- Implementationen OSI-gerechter Protokolle [Zi89], OSI-Management-Framework [Ga91], OSI-Directory-Service [Neu08].

3. Mitwirkung am Rechnernetz DELTA und an dessen Nutzung

Das Rechnernetz DELTA war ein Gemeinschaftswerk der AdW der DDR und des Hochschulwesens im Rahmen der Hauptforschungsrichtung Informationsverarbeitungssysteme. Seitens der TU Dresden wurden neben den obengenannten Grundlagenarbeiten durch Wissenschaftler des Rechenzentrums und der Sektion Informationsverarbeitung folgende Hauptaufgaben erbracht:

- Schaffung eines DELTA-Rechnernetzknotens und Einbindung der TU Dresden [FW77], [Wu87];

- Entwurf und Entwicklung eines Terminal-Interfaceprozessors TIP [IL79], [Ir81a], [Ir81b],;

- Aufbau eines netzgerechten Terminalsystems NTS für den Zugriff zu entfernten Rechnernetzressourcen [Gü80].

Bereits im Versuchsstadium und durchgängig ab 1980 erlernten die Studenten den Umgang mit der Rechnernetztechnologie von DELTA und zwar in Kooperation mehrerer Wissenschaftsbereiche: Alle Informatik-Studenten sandten ihre Simulationsaufgaben des Pflichtfaches Prozess- und Fertigungssteuerung (Prof. Stahn) über das am Wissenschaftsbereich Rechnersysteme geschaffene Terminalsystem zum Rechnernetzknoten Adlershof und erhielten tags darauf die Ergebnisse in Dresden/Gebäude der Sektion Informationsverarbeitung in Form von Rechnerausdrucken zurück. Konfiguration des Terminalsystems siehe Abb. 1.

Das netzgerechte Terminalsystem NTS gewährleistete

- Ein- und Ausgabe von Nutzeraufträgen an intelligenten Terminals, in der für DELTA-Arbeitsrechner (BESM-6, ESER) festgelegten Netzwerksteuersprache.

- Anschluss von Terminals unterschiedlicher Leistungsfähigkeit. Aufgabe des TIP: Steuerung des multiplexen Terminalanschlusses und der Verbindung zum lokalen Verarbeitungsrechnersystem oder zum Rechnernetz, Anschluss von Robotron-AP4-Terminals sowohl für autonomen als auch rechnernetzabhängigen Einsatz

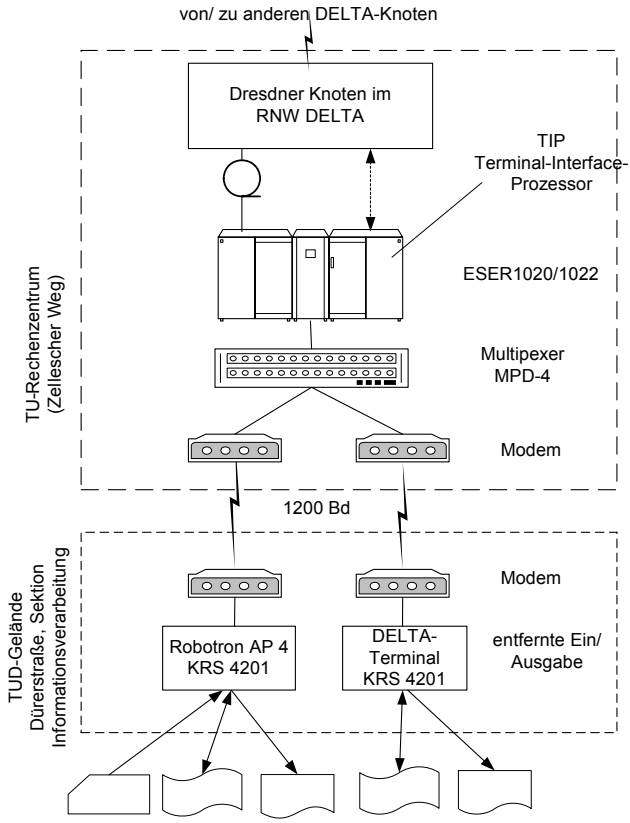


Abbildung 1: Konfiguration des netzgerechten Terminalsystems NTS

4. Lokale Netze

Die entsprechenden Arbeiten an der Sektion Informationsverarbeitung der TU Dresden (bzw. ab 1986 am Informatik-Zentrum) hatten folgende Zielstellungen:

- Forschung zur Wirkungsweise lokaler Netze und zur Gestaltung der entsprechenden Basissoftware [Lö87a], [Ha91];

- Implementation und Erprobung verschiedener Medienzugriffsmethoden auf der in der DDR erzeugten/verfügbaren Hardware [We85], [Oe86], [So87], [Ko89];

- Schaffung des LAN LOTUNET und Überführungen in die Praxis,

- Gestaltung der LAN-basierten Kommunikationstechnologie in CIM-Umgebungen (Projekt Fabrik der Zukunft).

Das LOTUNET-Konzept und seine Realisierungen sind in wissenschaftlichen Publikationen im In- und Ausland sowie u. a. in Habilitationen, Dissertationen und zahlreichen studentischen Abschlussarbeiten dokumentiert (siehe z. B. [Lö85], [Lö87a], [Lö89b], [Kl89], [Mö89], [WB90]). Bemerkenswert ist, dass LOTUNET wegen der angespannten Embargosituation ausschließlich auf in der DDR hergestellter Hard- und eigener Software entwickelt werden musste und so auch entstand. Netzwerkinterface-Einheiten wurden in Eigenregie entworfen, erprobt und in Kleinserien gefertigt. Bis 1990 wurden rd 3000 LOTUNET-Netzwerkkarten im Wissenschaftlichen Gerätebau der TU Dresden gefertigt und verkauft.

Herausragende Installationen von LOTUNET befanden sich außer an der TU Dresden im VEB Nagema Dresden [Lö87a], [Lö89b], [Mö89] und in der Bauakademie der DDR. Die streng verordnete Kontingentierung beispielsweise von Personalcomputern war in der DDR ein sehr großes Hindernis für die LAN-Forschung, -Entwicklung und -Erprobung. Man muss jenen wenigen Leitern von Kombinaten und Institutionen noch im Nachhinein dafür Anerkennung zollen, dass sie uns Gerätetechnik für die anwendungsorientierte Forschung leihweise bereit stellten.

In den 80-iger Jahren wurden an der TU Dresden sektionsübergreifend Ideen und Konzeptionen für künftige moderne Fertigungsweisen und –stätten entwickelt. Integraler Bestandteil dieses interdisziplinären Gedankengutes waren sowohl die Informatik als Gesamtheit wie auch die Informationstechnologie auf der Basis lokaler Netze. Einen Teil jener Vorstellungen bezeugen die Publikationen [Lö87a], [Lö87b], [Lö89c], [LS89].

Theoretische und praxisrelevante Forschung mit Orientierung auf lokale Sprach- und Datenkommunikation wurde auch an der Sektion Informationstechnik durchgeführt, die u. a. zu sechs Dissertationen und zahlreichen Veröffentlichungen führten [Neu08].

5. Zur Nachhaltigkeit jener Rechnernetzforschung und –entwicklung

Im Gegensatz zu den Verhältnissen an den Universitäten in der damaligen BRD, welche moderne Rechnernetztechnik durch Kauf erwerben konnten, musste diese im Hochschulwesen der DDR durch Eigenentwicklungen erst geschaffen werden. Dieser Nachteil hatte jedoch auch den Vorzug, dass z. B. in den Einrichtungen der TU Dresden praxisorientiertes Wissen mit der Maßgabe erarbeitet wurde, dieses unmittelbar in die rechentechnische Nutzung an der Universität (z. B. DELTA-Mitwirkung, LOTUNET), in die Lehre und industrielle Praxis zu überführen. Auf diese Weise entstand ein Erkenntnis- und Erfahrungsschatz der Rechnernetztechnologie, welcher auch heute noch von den zahlreichen ehemaligen Mitgestaltern getragen und weitergegeben wird. Hierzu zählen viele erfolgreiche Fach- und Führungskräfte und Unternehmer im In- und Ausland, welche seinerzeit die entsprechenden Arbeiten mit voran brachten. Seit Beginn der 80-iger Jahre erhielten alle Informatik-Studenten der TUD auch fundierte theoretische und praktische Kenntnisse der LAN-Technologie. Auch dies wirkt heute noch nach.

Zur Nachhaltigkeitswirkung gehört letztlich auch die Tatsache, dass aus dem Kreis der ehemaligen TU-Rechnernetzforscher und -entwickler heute ein halbes Dutzend als Universitäts- und Hochschulprofessoren der Rechnernetztechnologie und Kommunikationstechnik erfolgreich ihren Mann steht.

Literaturverzeichnis

[Lö76a] Löffler, H.: Probleme der Entwicklung von Rechnernetzwerken. Wiss. Z. TU Dresden, 25 (1976), 1193-1199.

[Lö76b] Löffler, H.: Optimalitätskriterien für die Informationsüberragung in Rechnernetzwerken. msr 19(1976), 128-130

[Lö77a] Löffler, H.: Nachrichtenverkehrstheoretische Analyse eines Modellrechnernetzes. - In: Vorträge auf dem Problemseminar Systemanalyse (Weißig, 29.11.-3.12.76). - Dresden; (1977) 25 - S. 47 – 53

[Lö77b] Löffler, H.: Analytische Rechnernetzmodelle. TU-Informationen, Sektion Informationsverarbeitung 08-06-77, 1-15

[HL77] Hofmann, U., Löffler, H.: Strategien im Kommunikationssystem von Rechnernetzwerken. Nachrichtentechnik/Elektronik 27(1977)7, 293 – 296

[Lö82] Löffler, H. Rechnernetzwerke. - 1. Aufl. - Berlin: VEB Verlag Technik, 1982. - Reihe Automatisierungstechnik, Band 199

[Lö84] Löffler, H.: Rechnerverbundsysteme. Berlin: Akademieverlag 1984

[Lö81] Löffler, H.: Study of computer network models. Proc. IFIP-UNESCO-Conf. COMNET `81, Budapest 1981, Working Papers 6/1-6/7.

[LM81] Löffler, H.; J. Mertins: Dynamisches Verhalten zeitvarianter Bedienungssysteme. msr 24 (1981), 9-12

[LM82] Löffler, H.; J. Mertins: Verhaltensmodellierung von Rechnernetzwerken bei zeitvarianter Auftragslast. Wiss. Z. TU Dresden, 31 (1982), 123-127

[Ho84] Zur Theorie der Flusssteuerung in Rechnernetzen. Diss. (B) TU Dresden 1984

[Me85] Meier, I.: Zur Gestaltung des Transportdienstes in lokalen Rechnernetzen. Diss. TU Dresden 1985

[Sa84] Sadowski, A.: Modellierung von Transportstationen in einem Rechnernetz durch geschlossene Bedienungsnetze. Diss. TU Dresden 1984

[Zs89] Zschiesche, M.: Konzeption und Implementation portabler Komponenten des OSI-Referenzmodells, dargestellt an der ISO.gerechten Transportschicht. Diss. TU Dresden 1989

[JL77] Jacob, H.-J., Löffler, H.: Analytische Beschreibung und Bewertung eines Modell-Rechnernetzwerkes. Nachrichtentechnik/Elektronik. - Berlin; 27(1977) 1, 14 – 16

[Oe80] Oehme, W.: Beitrag zur Verhaltensanalyse von Terminalsystemen in Rechnernetzen auf Grundlage der Theorie blockierender Systeme. Diss. TU Dresden 1980

[Kö87] König, H.: Kommunikationsprotokolle. Prinzip und Entwicklung. Diss. B. TU Dresden, 1987

[Kö90] König. H.: Kommunikationsprotokolle. Berlin: Akademie-Verlag 1990

[Ga91] Garbe, K.: Management von Rechnernetzen. Teubner-Verlag. Stuttgart 1991.

[Ne08] Neubert, P.: perönliche Information von Prof. Dr. Neubert an den Autor, 2008

[FW77] Frank, M. und N. Wulst: Probleme der Einordnung eines ERER-Rechners in ein Rechnernetz. ZfR-Informationen ZfR-77.04, 42-48.

[Wu87] Wulst, N.: Anwendungsorientierte Schichten am Beispiel der DELTA-Dienste und Beiträge zur Entwicklung offener Systeme. Diss. (B) AdW Berlin 1988 – siehe auch iir-Reporte 9/1987, 1-107

[IL79] Irmscher, K.; H. Löffler: Design and Performance Evaluation of Control Programs for Terminal Access in Computer Networks. TU-Informationen, Sektion Informationsverarbeitung, Part I: 08-08-79, Part II: 08-09-79 (1979)

[Ir81b] Irmscher, K.: Funktion, Struktur und Leistung von Terminalinterface-Prozessoren. INFO`81. ZfR-Informationen ZfR-80.13, 170-179.

[Ir81a] Irmscher, K.: Terminal-Interface-Prozessor von Rechnernetzen. Wiss. Z. TU Dresden, 30 (1981); 107-114.

[Gü80] Gütter, D., Irmscher, K., Löffler, H., Pohl., A., Schledz, H.-U.: Ein netzgerechtes Terminalsystem an der Technischen Universität Dresden. TU-Informationen, Sektion Informationsverarbeitung 08-05-80 (Dresden 1980).

[ZL89] Zschiesche, M.; Löffler, H.: Developing and Structure of LAN-Transportstations. - In: Proceedings Winterschool DSA'88 (Distributed System Architecture '88), Ed.: M. Baziewicz. - TU Wroclaw; (1989), 261-271

[Lö87a] Lokale Netze. Berlin: Akademie-Verlag, 1987. -München, Wien: Carl-Hanser Verlag 1988

[Ha91] Zur Gestaltung von LAN-Basissoftware für Personalcomputer. Habil.-Schr. TU Dresden 1991

[We85] Weidhase, F: et all.: Netzwerkinterfaceeinheit NIU/K1520. Nachrichtentechnik Elektronik. 35 (1985), 454-456

[Oe86] Oehlschaegel, W.: Realisierung von CSMA/CD-Buszugriffsmethoden für lokale Netze. Diss. TU Dresden 1986

[So87] Sokolowski, J.: Entwurf, Implementation und Erprobung des lokalen Ringnetzes LOTUNET 8.3 mit Tokenzugriffsprotokoll. Diss. TU Dresden 1987

[Ko89] Kolodziejczyk, T:. Entwurf und Realisierung des Tokenbusverfahrens für das lokale Netz LOTUNET. Diss. TU Dresden 1989

[Lö85] Löffler, H.: LOTUNET – ein modular strukturiertes lokales Rechnernetz der TU Dresden. Nachrichtentechnik Elektronik **35** (1985) 442-444

[Lö89b] Löffler et al.: Das lokale Netz NAGEMA-LOTUNET. rechentechnik/datenverarbeitung 26(1989) 2; 23-25

[Kl89] Rechnergestützte Inbetriebnahme und Qualitätsüberwachung von LOTUNET-Hardware-Komponenten. Diss. TU Dresden 1989

[Mö89] Mögling, G.: Entwurf und Anwendung von Konfigurations-, Test- und Messeinrichtungen im lokalen Netz NAGEMA-LOTUNET: ein Beitrag zur Verwaltung in lokalen PC-Netzen. Diss. TU Dresden 1989

[WB90] Weidhase, F.; Boytschenko, E. W.: LOTUNET-Kurzbeschreibung und Datendurchsatz. In: Proceedings of the International Conference on Local-Area Networks, Riga, Oktober 1990, pp. 166-170 (in russ.)

[Lö87b] Löffler, H.: Kommunikationsmodell und -technologie des künftigen Maschinenbaubetriebes auf der Basis lokaler Rechnernetze. - In: Wissenschaft Theorie und Praxis. - Kolloquium der TU Dresden 15. 4. 1987.- TU Dresden, Der Rektor. (1987). 39 – 53

[Lö87c] Löffler, H.: Neue Rechnergenerationen - CIM und LAN. - In: Wissenschaftliche Beiträge zur Informatik: Informatik-Zentrum des Hochschulwesens. - Dresden, l(1987)1, 31 – 34

[Lö89c] Löffler, H.: Lokale Netze für Büro- und Fertigungsautomatisierung. - In: Wissenschaft Theorie und Praxis Vortragsband zum CIM-Kolloquium zur Fabrik 2000 an der TU Dresden. Dresden (1989), 26-33

[LS89] Löffler, H.; Schubert, D.: Rechnerintegrierte Fertigung aus technologischer Sicht - Kommunikationsmodell/ Datenverwaltung - . In: H.-J. Jacobs: Interdisziplinäre Ansätze zur Rechnerintegrierten Fertigung (CIM). Dresden: Technische Universität, 1989, S. 103 - 117

Virtuelle Maschinen für Anwendungssoftware-Kompatibilität und zukunftsorientierte Lehre

Dieter Jungmann

Leiter Technologischer Vorlauf
Robotron Datenbank-Software GmbH
Stuttgarter Str. 29
01189 Dresden
dieter.jungmann@robotron.de

Abstract: Es wird die Entwicklung der Computer in der DDR vom ersten Kleinrechner bis hin zu Mainframes und Superminicomputern aus der Sicht der Kompatibilität ihrer Rechnerarchitekturen dargestellt. Sie war insbesondere bei den Kleinrechnern dadurch gekennzeichnet, dass jedes Nachfolgesystem nicht kompatibel zu seinem Vorgängersystem war. Am Rechenzentrum der Ingenieurhochschule Dresden wurde eine Vielzahl von Kompatibilitätslösungen auf der Grundlage der Implementierung virtueller Maschinen realisiert. Sie dienten zunächst der Gewährleistung einer experimentell gestützten Lehre für fortgeschrittene Rechnerarchitekturen und deren Sprachverarbeitungssysteme. Mit leichtem Zeitversatz wurden aber auch Simulatoren und Emulatoren entwickelt, die die uneingeschränkte und produktive Weiternutzung geschaffener Software auf nicht kompatiblen Nachfolgerechnern gewährleistet. Später lösten gezielte Forschungsarbeiten im Auftrag des Kombinates Robotron zu Rechnerarchitektur und Sprachverarbeitungssystemen die aus eigener Sicht betriebenen Arbeiten ab. Die fachliche Entwicklung des Autors war eng verbunden mit diesen Arbeiten von Anbeginn an.

O Vorbemerkungen

Meine fachliche Entwicklung stand in enger Verbindung zur Entwicklung von elektronischer Rechentechnik und Informatik in der DDR. Das 1963 begonnene 3-jährige Studium „Hochfrequenztechnik“ an der Ingenieurschule für Maschinenbau und Elektrotechnik wurde im 2. Studienjahr zu „Elektronischer Rechentechnik“ als Konsequenz aus dem „Ersten Datenverarbeitungsprogramm“ vom 3.7.1964 [Me64] umgestaltet. Fachrichtungsleiter Wilhelm Leupold spielte dabei eine Pionierrolle hinsichtlich der Profilierung des Ausbildungsprogramms, der Absicherung der rechentechnischen Basis für Praktika und der ersten Lehrveranstaltung „Elektronische Rechentechnik“. Die rechentechnische Ausstattung bestand aus einem Kabinett mit mechanischen Tischrechenmaschinen, einem programmgesteuerten Modellrechen-automaten auf Relaisbasis, einem Analogrechner des Institutes für Datenverarbeitung (IDV) und einem ersten digitalen Kleinrechner SER2a. Erste ingenieurmäßige Arbeiten führte ich in der Analog- und Hybridrechentechnik durch:

- Rekonstruktion des Analogrechners zum Modellregelkreis für die Simulation technischer Systeme

- Entwicklung eines Spezialrechners für die Simulation von Projektabläufen mittels CPM, den ich mit zwei weiteren Kommilitonen zu einem Wirtschaftspatent anmeldete.

Mein Fachrichtungsleiter überzeugte mich, an der Ingenieurschule als Laboringenieur für Elektronische Rechentechnik zu bleiben. Durch Wartungs- und Programmierarbeiten für den geschenkten D4a war mir die Gelegenheit gegeben, tiefer in die Materie einzudringen. Wenig später kam die industrielle Version C8205 hinzu. Die intensive Beschäftigung mit Methoden der Simulation führte sehr bald dazu, den C8205 als Basis für die CPM-Simulation, aber auch für die Simulation des wesentlich leistungsschwächeren SER2 einzusetzen.

1 Motivation

Basierend auf meinen ersten Simulationserfahrungen wurde dann am Rechenzentrum der Ingenieurhochschule Dresden eine lückenlose Folge von Simulatoren bzw. Emulatoren für Kleinrechner vom SER2 über den C 8205/C 8205Z zum KRS 4200/4201 und K1600 realisiert. Der KRS-Prozessor für die K 1630 wurde in Kooperation mit dem VEB ZFT Robotron in die Produktion überführt.

Die zur damaligen Zeit relativ hohe Operationsgeschwindigkeit des PRS 4000 bzw. KRS 4201 ermöglichte sogar die Architektur der EDVA R300 nahezu uneingeschränkt auf diesen Kleinrechnern zu simulieren und so mittels der R300 Hardware-Testprogramme zu verifizieren.

Zur Gewährleistung einer modernen Ausbildung wurden aber auch Assembler, Compiler und Simulatoren für Computer neuer Rechnergeneration auf Computern älterer Generationen implementiert und für Praktika in der Lehre eingesetzt.
Im Vortrag werden die theoretischen Grundlagen, eine Systematik realisierter Anwendungen und einige Fallbeispiele dargestellt. Abschließend wird die Aktualität der damaligen Arbeiten am Beispiel des Produktes VMware für die Desktop- und Server-Virtualisierung gezeigt.

2 Theoretische Grundlagen

In historischen Abhandlungen ist es sinnvoll auf einheitliche Begriffe zu verzichten, weil zu bestimmten Zeitabschnitten und in verschiedenen Ländern unterschiedliche Begriffe für das gleiche Objekt verwendet wurden. Das trifft insbesondere für die DDR zu. Ähnlich wie heute noch in Frankreich, wurden international anerkannte Begriffe in englischer Sprache durch Begriffe der Landessprache ersetzt.

Einschränkend ist zu bemerken, dass die politische Doktrin der DDR dazu Wandlungen unterworfen war. In diesem Beitrag werden die Begriffe Rechner, Computer, Rechenmaschine, Rechenanlage, EDVA für reale Objekte synonym verwendet. Der Begriff Maschine, eng. *machine,* wird dagegen für abstrakte Konstrukte verwendet.

Das Prinzip der Virtualität wurde in der elektronischen Rechentechnik für die Realisierung aller Komponenten eines Computers verwendet, um die physische Begrenztheit realer Komponenten zu überwinden:

- Hauptspeicher als virtueller Speicher
- Zentraleinheit (Prozessor) als virtuelle Maschine (Virtual Machine, VM)
- Periphere Geräte als Virtual Devices
- Computer als Computer Grid.

Zur Lösung des Problems der Lauffähigkeit von Programmen eines Computersystems auf einem anderen gibt es folgende Ansätze:

- Kompatibilität zwischen Generationen einer Computerfamilie (IBM 360, …)
- Kompatibilität mittels expliziter virtueller Maschinen (Interpreter):

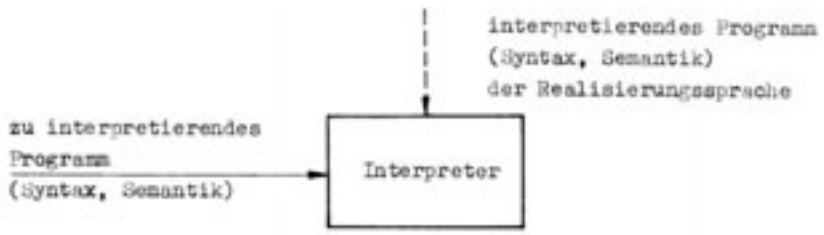


Abbildung 1: Blockschaltbild eines interpretierenden Sprachverarbeitungssystems

- Portabilität mittels impliziter virtueller Maschinen (Compiler):

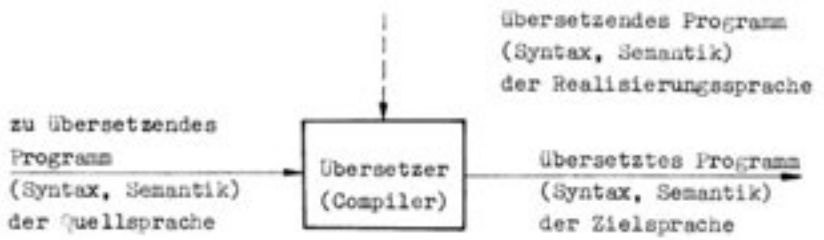


Abbildung 2: Blockschaltbild eines übersetzenden Sprachverarbeitungssystems

Der Klassiker auf dem Gebiet der Realisierung von Interpretern zum Zwecke Kompatibilität war V. Risak mit seinem Buch „Simulation von Digitalrechnern“ [Ri71].

Die darin enthaltenen Methoden mit Beispielanwendungen wurden in meiner Dissertation A mit dem Titel: „Simulation von Digitalrechnern – Technik und Technologie der programmtechnischen Implementierung von Digitalrechnern“ [Ju77] auf der Basis der DDR-Rechentechnik erweitert angewendet. Die nachfolgende Definition entspricht der experimentellen Basis dieser Arbeit und deckt auch die aus der Literatur bekannten Anwendungen.

In beiden Fällen ist Datenträgerkompatibilität eine notwendige Voraussetzung. Dieses Kriterium ist aus heutiger Sicht besonders kritisch zu bewerten, weil die meisten früheren Datenträger mit heutigen Computern nicht mehr lesbar sind.

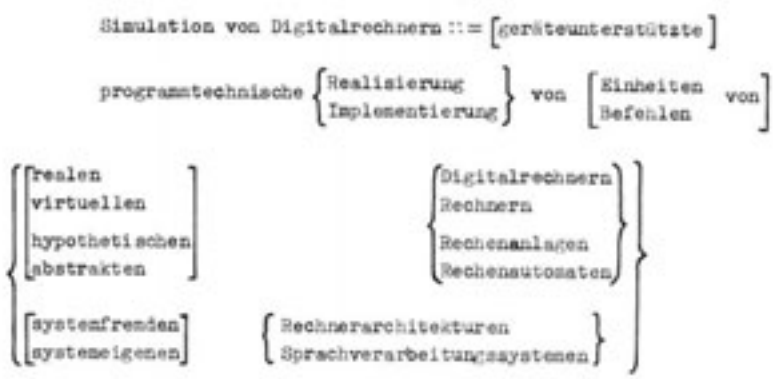


Abbildung 3: Definition Simulation von Digitalrechnern

3 Anwendung virtueller Maschinen

3.1 Systematik der Anwendung

In der Zeit von 1970-2004 wurden zunächst im Rechenzentrum der Ingenieurhochschule Dresden, dann ab 1986 am Lehrstuhl Rechnerarchitektur des Informatikzentrums und ab 1993 am Lehrstuhl Entwurf von Rechnersystemen eine Vielzahl von F/E-Arbeiten zum Thema virtuelle Maschinen durchgeführt. Etwa 1975 waren es zunächst nur Überführungsleistungen an das Kombinat Robotron, die Ende der 70er Jahre zu Auftragsarbeiten geführt wurden.

Es wurden folgende Anwendungsgebiete verfolgt:

1. Praktikumssysteme für die **Informatiklehre**
2. Kompatibilität für Anwendungsprogramme auf nichtkompatiblen Nachfolgerechnern (**IT-Anwendung**)
3. Experimentalsysteme für die **Informatik**-, insbesondere aber Rechnerarchitektur**forschung.**

Eine Veröffentlichung aus dem Jahr 1982 [Ju82] zeigt eine Übersicht in Form einer Kompatibilitätsmatrix zum Stand der Kompatibilität zwischen den häufig genutzten Computern. In [Ju77] ist eine weit umfangreichere Darstellung enthalten, die weitere Computer beinhaltet, die ich z. B. während meines mehrfachen Studienaufenthaltes am LETI und am Informatik- und IT-College in Budapest kennenlernte. Abbildung 4 aus [Ju82] zeigt in jeder Zeile einen Wirtsrechner und in jeder Spalte einen Zielrechner. Die Kennzeichen Y haben folgende Bedeutung:

Y = A In Anwendung befindlich
R In Realisierung befindlich

P Geplante Vorhaben
S Sinnvolle Anwendung
M Mögliche Anwendung
(Y) Eingeschränkte Anwendbarkeit
Y. Von IH Dresden realisiert

	SER 2	C 8205	C 8205 Z	d 1840	KSR 4100	KRS 4201	RS 4000	MRES 20	BC 5100	URS 5000	K 2063	SKR	R 300	ESER I	ESER II	BESM 6
SER 2																
C 8205	A.	A				A.										
C 8205 Z	A	A	A			A.										
d 1840	S															
KSR 4100		A														
KRS 4201	A.	A.	A.	S		A.	A.	A	S	(A)	R.	A.	A.	P	M	M
RS 4000	A.	A.	A	M		A.	A.	A	S	(A)	S	S	A.	(A).	M	M
MRES 20	M	M	S	S				A	S	A	S	S				
BC 5100	M	M	R.	(P)					A			(P).		(P).		
URS 5000						S		A			S	S				
K 2063	R.	R.	R.	S	S	R(-)					A.		R.	R.	S	S
K 2662			R					R.		R.	A.	A				
K 2663			R					R.			A.	A				
R 300		(A).	(A).			(A).	(A).						A.	(A).		
R 22						(A)		(A).				R	M	A		
R 40						(A)		(A).				R	A	A		
R 55 M						M		M.				R	M	A	A	S

Abbildung 4: Gesamtübersicht zum Stand der Kompatibilität im Jahr 1982

Diese Matrix veranschaulicht, dass die meisten Anwendungen an der IH Dresden entwickelt wurden (mit A. markiert). Es werden auch die verwendeten Wirtsrechner ersichtlich, das sind Zeilen mit Symbol Y. In den nachfolgenden Abbildungen werden die Kompatibilitätsbeziehungen zwischen Computern ersichtlich. Meistens sind es Computer einer Größenklasse. An den Bögen steht K für Kompatibilität, Kt für Teilkompatibilität oder eine Software-Produktbezeichnung.

Softwareprodukte sind Editoren, Cross-Assembler/Compiler, interpretierende Testsysteme, Konvertierungsprogramme, Interpreter (Simulatoren), Software-Support des KRS-Emulator-Prozessors für den K1630.

- **C 8205 (Z):** Das war der erste Rechner mit einem Prozessor, der Bitketten verarbeiten konnte und eine für die damalige Zeit akzeptable Operationsgeschwindigkeit hatte (etwa 10fach höher als SER 2). Ausführliche Informationen sind in [KJ04] enthalten.

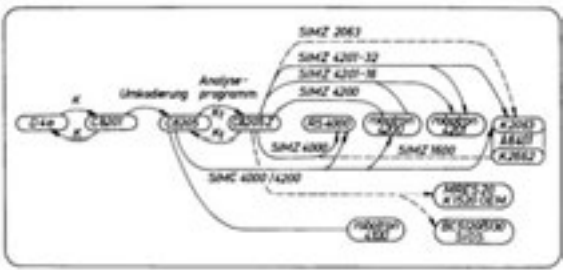


Abbildung 5: D4a-C8205 (Z)-Kompatibilitätsbeziehungen

- **R 300:** Das war der erste Computer mit einer leistungsfähigen Peripherie. Er wurde deshalb insbesondere als Host für Cross-Assembler/Compiler der Kleinrechner mit eingeschränkter Peripherie eingesetzt.

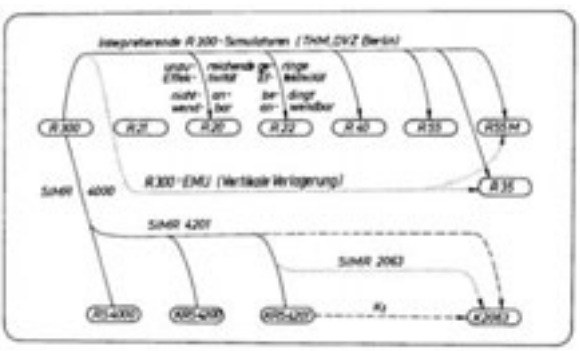


Abbildung 6: R 300-Kompatibilitätsbeziehungen

- **KRS 4201/RS 4000:** Diese Computer hatten ein günstiges Preis/Leistungs-Verhältnis, verfügten über einer relativ hohe Operationsgeschwindigkeit (<= 1 MIPS) und einen Hauptspeicher von 64 KByte. Mit diesen Eigenschaften konnte die Rechnerarchitektur aller Vorgänger programmtechnisch implementiert werden.

- **K1630:** Die Hauptspeicherkapazität war zwar höher als die des KRS 4201, doch die Operationsgeschwindigkeit war wesentlich geringer durch die Verwendung von N-MOS-Prozessorschaltkreisen an Stelle von TTL-Schaltkreisen. Die Kompatibilität zum Vorgänger KRS 4201 war programmtechnisch nicht realisierbar.

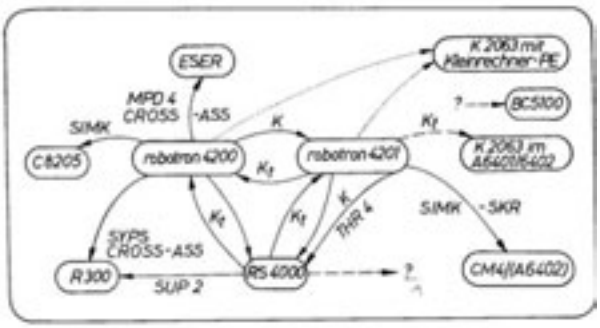


Abbildung 7: KRS 4201/RS 400-K1630-Kompatibilitätsbeziehungen

- **ESER I/II:** Bis auf das schlechtere Preis/Leistungsverhältnis, vergleichbar mit RS 4000.

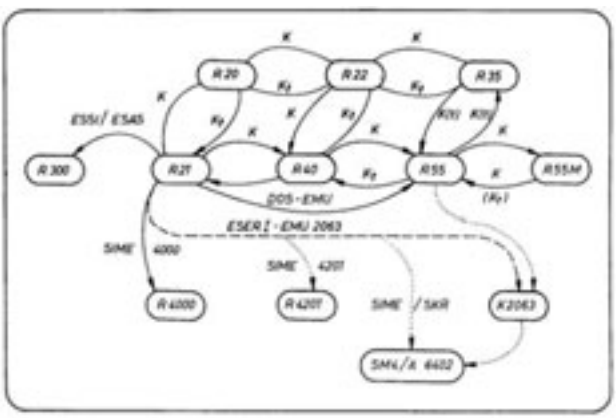


Abbildung 8: EDVA (Mainframe)-Kompatibilitätsbeziehungen

3.2 Informatiklehre

Das Rechenzentrum der IH Dresden wurde im Jahr 1969 primär als Ausbildungsrechenzentrum gegründet. Zunächst war ich Leiter des Einsatzstabes und später stellv. Leiter des RZ und Leiter des Softwarebereichs. Die Mitarbeiter waren bis auf Ausnahmen Absolventen: Am Anfang von der Ingenieurschule für Maschinenbau und Elektrotechnik, später von der Ingenieurhochschule. Ein großer Vorteil war, dass bereits 1966 mit der Ausbildung von Programmieringenieuren begonnen wurde. Maßgeblich beteiligt waren Wilhelm Leupold und Helmut Jenke.

Bedingt durch limitierte Bilanzen und fehlende finanzielle Mittel bekamen wir 1970 zunächst nur ausgesonderte Computer des Kombinates Robotron: Ein R 300 mit Seriennummer 1, der später an das Technische Museum Dresden übergeben wurde und ein Prozessrechner PR 2100, der später in das Museum folgte.

Das war in einer Zeit, in der in der Wirtschaft bereits IBM 360-kompatible Mainframes und Prozessrechner DDP 516 bzw. die Eigenentwicklung RS 4000 im Einsatz waren.

Mit viel Kreativität und Engagement wurde in einem erheblichen Umfang maschinenorientierte Software (damals Systemunterlagen genannt) für neue Computergenerationen entwickelt. Unser Entwicklungspotential konnte durch den Einsatz von Diplomanden und Kooperationspartner der Hochschulrechenzentren des Hochschulwesens bedeutend erweitert werden.

Folgende größere Softwarepakete wurden entwickelt und in Praktika eingesetzt:

- **PL/1-Compiler mit dem IBM/OS-Sprachumfang für den R 300:** Der Compiler wurde in der R300-MACRO-Assembler-Sprache als 8-Pass-Compiler im Wesentlichen von Peter Hoffmann und Walter Schiller entwickelt. Die Übersetzung des Compilers dauerte 24 Stunden. Es kam oft vor, dass die Maschine abstürzte und neu begonnen werden musste. So wurden Änderungen weitestgehend im Maschinencode ausgeführt, um Rechen- und Entwicklungszeit zu sparen. Als schließlich das erste minimale PL/1-Programm generiert war, stellten wir fest, das für die Übersetzung des generierten MACRO-Assembler-Programms eine Stunde Rechenzeit erforderlich war. Peter Hoffmann entwickelte in nur 6 Monaten einen neuen MACRO-Assembler, der nur 1/3 der Anzahl der Befehle aufwies und 3mal schneller war. Damit war die Durchführung von PL/1-Praktika auf dem R 300 gesichert. Bei einem Hauptspeicher heutiger PCs von 1 GByte kann man sich nicht mehr vorstellen, solch komplexe Software mit einem Hauptspeicher von 40.000 Zeichen zu 6 Bit implementieren zu können.

- **ESER-Assembler/Simulator für den R300:** Ein R300-Assembler-Praktikum entsprach nicht mehr den Praxisanforderungen. Basierend auf umfangreichen Erfahrungen in der Realisierung von Sprachverarbeitungssystemen konnte in nur einem Jahr der volle ESER-Assembler-Sprachumfang implementiert werden. Die geringe Operationsgeschwindigkeit des ESER-Simulators war für den Testbetrieb völlig ausreichend. Eine besondere Herausforderung bestand darin, einen binär organisierten Prozessor auf einem dezimal organisierten Prozessor zu simulieren.

- **Crossassembler Klein- und Prozess- und Mikrorechner:** In diesem Fall wurden die Vorteile des R 3000 mit Magnetbandtechnik für die Quellcodeverwaltung und leistungsfähigem Paralleldrucker für das Drucken der Assemblerlisten genutzt. Ein schneller Lochbandstanzer stellte den Maschinencode für den Test auf Computern wie KRS 4201, Prozessrechner PR 2100 und Mikrorechner K1520 bereit. Mittels PDP 11-Simulator konnten mit dem KRS 4201 bereits SKR- bzw. K1600-Programme getestet werden.

- **BASIC-Timesharing-System für den R4000:** Mit dieser Entwicklung von Peter Hoffmann konnte erstmalig der Stapelverarbeitungsbetrieb eines geschlossenen Rechenzentrums verlassen werden. Die Studenten konnten interaktiv ihre Programme entwickeln und testen.

3.3. IT-Anwendung

Die Entwicklung der Kleinrechentechnik in den früher selbständigen Betrieben Karl-Marx-Stadt und Zella-Mehlis sowie später im Kombinat Robotron war dadurch gekennzeichnet, dass die Architektur jedes Nachfolgerechners nicht kompatibel zum Vorgänger war. Die Ursache dafür war zunächst von technischer Natur, weil bis zur Verfügbarkeit von TTL- und Halbleiterschaltkreisen ein Top-down-Entwurf ausgehend von einer Rechnerarchitektur hin zu einer Rechner-Implementierung und -Realisierung nicht möglich war. Das trifft zu für die Serien SER2 und C8205. So wird die Architektur des C8205 (Z) eindeutig durch die Struktur des Trommelspeichers bestimmt. Unter Verwendung eines Kern- oder Halbleiterspeichers wäre man niemals auf eine Wortlänge von 33 Bit gekommen.

Danach beim RS 4000 und damit auch beim KRS 4201 war die Orientierung an dem Industrie-Standardrechner DDP 516 bestimmend. Wegen der geringen Abweichung in der Befehlssatz-Architektur (*Instruction Set Processor, ISP*) und vollständig abweichenden E/A-Architektur war die Verwendung des Original-Betriebssystems verhindert. Eine umfangreiche Eigenentwicklung von Betriebssystemen und Systemsoftware war die Konsequenz.

Diese Vorgehensweise wurde von Kombinaten wie Zeiss Jena nicht akzeptiert. Sie entwickelten und produzierten einen sogenannten Kleinsteuerrechner KSR 4100, der 100 %-kompatibel zu dem auf der Welt als anerkannter Standard geltenden PDP 8 war. Die entwickelte Anwendungssoftware war auf Originalrechnern lauffähig. Der KSR 4100 war aus vielen Gründen in das sogenannte NSW nicht exportierbar. Später musste Robotron diesen Rechner auch produzieren.

Die Beibehaltung der KRS/PRS-Architektur war aus vielen Gründen für die Familie K1600 nicht fortsetzbar. Wichtigster Grund war, dass im RGW auf die PDP 11-Architekur orientiert wurde. Diese Architektur ist unvereinbar mit der Vorgängerarchitektur. Wichtigste Anforderung war, dass das Originalbetriebssystem und damit auch die Anwendungssoftware lauffähig sind. Abweichungen gab es aber im Bussystem und in den metrisch ausgerichteten Steckverbindern.

Inzwischen wurde von führenden Kombinaten mit dem Kombinat Mikroelektronik an der Spitze durchgesetzt, dass genau wie bei Schaltkreisen Steckerkompatibilität zu gewährleisten ist. Auch die bisher geringe Zuverlässigkeit von Computersystemen wurde nicht länger akzeptiert. So wurde auf höchster Ebene durchgesetzt, dass abweichend von der TGL das Zoll-System für den K1600-Nachfolgerechner K1840 zu Grunde gelegt wurde.

Auf höchster politischer Ebene musste entschieden werden, dass künftig Gold für Steckverbinder eingesetzt werden durfte. Ein Gesetz zur Rückgewinnung des eingesetzten Goldes wurde durch die Volkskammer beschlossen. Durch die strenge Orientierung am Vorbild VAX 11/780 mit einem PDP 11-Kompatibilitätsmodus war auch die Programmkompatibilität vom K1830 zum K1600 gesichert.

Nachfolgend werden Lösungen zu offenen Kompatibilitätsproblemen zwischen den Kleinrechnergenerationen dargestellt. Eine allgemeine Übersicht wurde dazu bereits im Abschnitt 3.1 vermittelt.

SER 2-Serie: Die Anwendungsprogramme waren durch die SER 2-Simulatoren für C 8205 und KRS 4201, insbesondere aber beim KRS 4201 mit sehr hohem Performancegewinn weiter nutzbar.

C 8205 (Z): In so manchem Unternehmen war die Anwendungsentwicklung für C 8205 Z noch nicht abgeschlossen, als die Produktion der Serie eingestellt wurde. Die Simulatoren SIMC für C 8205 und SIMZ für C8205 Z sicherten eine uneingeschränkte Kompatibilität für alle Anwendungsprogramme. Die Gewährleistung der Abwesenheit von Fehlern wurde zunächst durch Anwendung aller Hardwaretestprogramme in Angriff genommen. In der mindestens 800fachen praktischen Anwendung in der DDR und im SW traten noch 3 weitere sehr spezielle Fehler auf, die nach gründlicher Analyse behoben wurden. Danach war Fehlerfreiheit gesichert. Die Effektivität der Anwendung konnte bedeutend gesteigert werden: Mindestens 7fache Steigerung der Operationsgeschwindigkeit und wesentlich vereinfachte Bedienung durch die Simulation der Lochbandtechnik auf Magnetband (virtuelle Gerätetechnik).

R 300: Das Problem der weiteren Nutzung von R300-Anwendungsprogrammen stand erstmals auch für unser Rechenzentrum. Der riesengroße Rechnerraum des R 300 wurde für die geplante ESER-Anlage EC 1022 aus Minsk gebraucht. Es war klar, dass für eine Simulation der R 300-Architektur mindestens die Operationsgeschwindigkeit der EDVA R 40 aus der Robotron-Produktion erforderlich ist, der uns aber nicht zur Verfügung stand. Zu einem relativ frühen Zeitpunkt wurden uns Anlagen vom Typ RS 4000 durch Robotron geschenkt. Mit unseren Erfahrungen aus anderen Simulationsverfahren kamen wir zu der Einschätzung, dass der RS 4000 mit ESER-Peripherie als R 300-Similator bestens geeignet ist. Die Entwicklung wurde unverzüglich begonnen und durch Lösung vieler Detailprobleme zu einem guten Ende geführt. Aus der Praxis kam die Forderung, diesen Simulator auch für einen ausgebauten KRS 4201 verfügbar zu machen. Beide Versionen wurden an Robotron zum Zwecke des Vertriebs übergeleitet. Später ergab es sich, dass sowohl das Rechenzentrum der Universität Magdeburg als auch das LfA Berlin einen R 300-Simulator für den R 40 entwickelten.

KRS 4200/4201: Die Situation beim Übergang zum System K1600 war vergleichbar: viele Entwicklungsvorhaben waren noch nicht abgeschlossen. Bei der Planung des Nachfolgesystems hatte sich Robotron darauf verlassen, dass die KRS-Assembler-Programme in K1600-Assemblerprogramme konvertiert werden können, und dass inzwischen auch Anwendungen in höheren Programmiersprachen entwickelt wurden, die durch einen äquivalenten Compiler nur neu zu übersetzen sind (Portabilität).

Beide Annahmen stellten sich sehr bald als Trugschluss heraus. Ich hatte vom Chefentwickler den Auftrag erhalten, zu diesen Annahmen eine detaillierte Studie zu verfassen. Nach kurzer Zeit stand fest, dass den Anwendern eine Alternative geboten werden musste: Einsatz eines Emulationsprozessors auf der Basis eines schnellen Bit-Slice-Microprocessor-Systems analog Intel 3000, hergestellt in der Sowjetunion.

Der Auftrag zur Entwicklung eines Funktionsmusters ließ nicht lange auf sich warten. Parallel zum Architekturentwurf wurden Simulatoren für die Ebenen Befehlssystem- und Mikroprogrammebene entwickelt, um Programme entwickeln zu können, noch bevor der Hardwareprozessor verfügbar war. Die Gesamtarchitektur entstand durch eine optimierende Verteilung auf die verschiedenen Ebenen: Elementare Funktionen mit größter Häufigkeit der Verwendung in die Hardware, Funktionen mit mittlerer Häufigkeit der Verwendung in die Firmware (Mikroprogramme) und Funktionen mit geringer Häufigkeit der Verwendung in die Software. Das Befehlssystem bestand aus den Originalbefehlen und selbst definierten Befehlen zur Optimierung der Performance der in Software implementierten Funktionen des Originalsystems. Früheren Erfahrungen folgend, wurden die Hardware-Testprogramme des KRS 4201 verwendet. Dazu musste die Architektur des KRS 4201 einschließlich Interrupt- und E/A-System originalgetreu nachgebildet werden. Das Ergebnis konnte sich sehen lassen: Die Performance war besser als die des K1630-Prozessors, und es war sogar das Betriebssystem für kommerzielle und wissenschaftlich-technische Anwendungen lauffähig [KS82]. Auch die früher für den KRS entwickelten Simulatoren konnten weiter genutzt werden. Der Prozessor wurde in einer größeren Stückzahl produziert.

3.4 Informatikforschung

Eigene experimentelle Arbeiten, unterstützt durch eine Vielzahl von Diplomarbeiten, wie z. B. die Simulation der ESER-Architektur auf dem RS 4000 sowie die Verfügbarkeit der weltweiten Rechnerarchitekturliteratur an der TU Dresden und bei Robotron, wurden zur Gestaltung von Rechner-, insbesondere Prozessorarchitekturen durchgeführt. Die Ergebnisse sind in [Ju84] zusammengefasst. Das finale Ergebnis waren eine Universalarchitektur UNIA und ein Virtual Instruction Set Prozessor (VISP).

Durch meine Tätigkeit bei Robotron auf dem Gebiet der Entwurfssoftware für Schaltkreise und komplette Rechnersysteme wurde an mein damaliges Team der Auftrag vergeben, ein universelles Simulationssystem für den Rechnersystementwurf zu entwickeln. Uwe Crenze erstellte im Rahmen seiner Dissertation dafür das Konzept und implementierte auch den größten Teil des ISPS-Simulationssystems. Verwendet wurde die Spezifikation der Hardwarebeschreibungssprache ISPS der CMU.

Ende der 80er Jahre war das Problem der Auswahl eines RISC-Prozessors zu lösen [SJTM94]. Mit RISC-Prozessoren waren höhere Taktfrequenzen und eine größere Anzahl von Befehlen pro Takt zu erzielen. Durch ergänzende Marktbetrachtungen waren die Prozessoren MIPS und SPARC in die engere Wahl gefallen.

In einem weiteren Forschungsauftrag war ein Simulator zu entwickeln, der Metriken für die Auswahl eines der beiden Alternativen lieferte. Die Ergebnisse wurden in der Dissertation von Wasja Suworow beschrieben. Eine spezielle Frage war, ob anstelle dieser Universalprozessoren der Ein-Chip-Computer „Transputer“ für massiv-parallele Systeme als nächster Chip in die Entwicklungspläne von Robotron und Mikroelektronik einzuordnen ist.

Aus Ergebnissen einer parallelen Forschungsgruppe des Informatikzentrums und internationalen Tagungsberichten war bekannt, dass noch keine praxiswirksamen Ergebnisse einer universellen Parallelisierbarkeit von Software vorlagen. Teilweise wurden diese Auseinandersetzungen ohne Berücksichtigung einer soliden fachlichen Basis auf politischer Ebene ausgefochten.

In der Diplomarbeit von Wolfgang Schwarz wurde untersucht, ob man durch Simulation eines Intel x86-Prozessors problemfrei schneller Viren erkennen kann. Aus Kapazitätsgründen konnte dieser Ansatz nicht weiter verfolgt werden.

Es war inzwischen klar, dass der Performance-Vorteil der RISC-Prozessoren mit dem Nachteil einer geringeren Codedichte erkauft werden muss. Bei universellen Systemen ist dies bei dem schnell wachsenden Integrationsniveau von Speicherschaltkreisen kein Problem. Bei Embedded Systems und Ein-Chip-Computern sieht es aber anders aus. Im Rahmen eines Förderprojektes entstand dazu die Dissertation von Gerd Markwart.

Vor meinem Ausscheiden aus der Technischen Universität Dresden hatte ich gundlegende Ideen zu einem universellen Parallelprozessor entwickelt, der rekonfigurierbar ist. Der Compiler einer höheren Programmiersprache entscheidet in Abhängigkeit von dem analysierten Quellcode, in welchem Modus der Parallelprozessor betrieben wird. Patentstudien zeigten, dass Anteile dieser Ideen bereits patentiert wurden, aber Alleinstellungsmerkmale noch offen blieben.

4 Virtuelle Maschinen aus heutiger Sicht

Analysiert man heute Anwendungen virtueller Maschinen kommt man zu folgenden Ergebnissen:

- Unternehmen, die RISC-Prozessoren in ihren PCs und Workstation einsetzen, simulieren die CISC-Architektur Intel x86.
- IBM nutzt inzwischen seinen sehr leistungsfähigen PowerPC-Chip für die Emulation der Mainframearchitektur.
- AMD hat in Dresden ein Kompetenzzentrum gegründet, dass mit Mitteln der Simulation und Compilertechnik die Prozessorarchitektur optimiert. Aus der Sicht der Methodik entspricht die Vorgehensweise genau der unserer früheren Arbeiten.
- Die Technologie Virtueller Server hat durch die hohe Prozessorleistung eine enorme Bedeutung erlangt. Auf einer realen Maschine können mehrere virtuelle Maschinen mit völlig eigenständigen Softwarekonfigurationen laufen.

5 Schlussbemerkungen

Mit diesem Beitrag soll vermittelt werden, dass die begrenzte Verfügbarkeit moderner Computertechnik und Software für Lehre und Forschung in der DDR eine Herausforderung war, international fortgeschrittene Rechnerarchitekturen und Programmiersysteme auf veralteten Klein- und Prozessrechnern sowie Datenverarbeitungsanlagen für eine fortgeschrittene Lehre zu implementieren.

Dadurch war es im begrenzten Umfang möglich, den oft beklagten zeitlichen Abstand von 8-10 Jahren zu verkürzen.

Auf der Grundlage des dabei erworbenen Knowhows wurde das Problem der Praxis gelöst, die Investitionen in umfangreiche Anwendungssoftware beim Übergang zu nichtkompatiblen Nachfolgrechnern zu schützen. Die hohe Zahl von ca. 1.000 Lizenzabschlüssen für die SER2- und C8205 (Z)-Simulatoren für KRS 4200/4201 waren der Beweis dafür, dass die These: „Kompatibilitätslösungen zu neuen Rechnergenerationen verhindern die Ausnutzung neuer technischer Möglichkeiten“ nicht haltbar war. Genau das Gegenteil war der Fall: Erst durch diese Möglichkeit war der Zeitdruck der Einsatzvorbereitung genommen und der Entwurf sowie die Implementierung von Anwendungssoftware mit neuen funktionalen und nichtfunktionalen Eigenschaft möglich. Unter den Bedingungen einer freien Marktwirtschaft wäre die Ausgründung einer Firma die Konsequenz gewesen. So kamen einige weitere Kompatibilitätslösungen nicht über das Konzeptstadium hinaus. Produkte zur Server-Virtualisierung wären die logische Fortsetzung gewesen.

Eine gesicherte Finanzierung von F/E-Arbeiten durch das Ministerium für Hoch- und Fachschulwesen und die Absicherung der materiell-technischen Basis durch die Kooperation mit dem Kombinat Robotron war die Basis für eine langfristige Profilierung von Forschungsarbeiten zur Rechnerarchitektur und Sprachverarbeitungssystemen, die bis in die ersten Jahre nach der politischen Wende hinein reichten.

Literaturverzeichnis

[Ju77] Jungmann, D.: Simulation von Digitalrechnern – Technik und Technologie der programmtechnischen Realisierung von Rechnerarchitekturen, Dissertation A, Ingenieurhochschule Dresden, 1977.

[Ju83a] Jungmann, D.: Zur Programm- und Datenkompatibilität zwischen Rechnern, rechentechnik datenverarbeitung 10/1982, S. 5-9.

[Ju84] Jungmann, D.: Tendenzen zur Gestaltung der Befehlssysteme von Rechnerarchitekturen, INFO84, Vorträge Rechnerarchtektur,1984, S.23-32.

[KJ04] Krause, Ch.; Jacobs, D.: Von der Schreibmaschine zu Mikrorechnersystemen, LNI – Informatik in der DDR, Symposium in Chemnitz, 2004, S. 116-139.

[KS82] Kofer, C.; Schöne, S.: Entwicklung eines robotron-4200/4201-Emulators für K1600, rechentechnik datenverarbeitung 10/1982, S. 10-13.

[Me64] Merkel, G.: Computerentwicklungen in der DDR – Rahmenbedingungen und Ergebnisse; LNI – Informatik in der DDR, Symposium in Chemnitz, 2004, S. 40-54.

[SJTM94] Suworow, W., Jungmann, D., Tavangarian, D., Monjau, D.: Klassifikation von RISC-Prozessoren, Wissenschaftliche Beiträge zur Informatik 8(19994) H. 2, Technische Universität Dresden, Fakultät Informatik, 1994

Forschungsarbeiten zur Simulation Diskreter Prozesse und zu objektorientierter Simulationssoftware an der IHD

Martin Frank

Gerichtsstraße 20
01069 Dresden
mfrank.dd@t-online.de

Abstract: Wesentlicher Gegenstand der Darstellung ist ein an der Ingenieurhochschule (IHD) bzw. ab 1986 am Informatikzentrum Dresden entwickeltes, auf die unmittelbare Nutzung am Arbeitsplatz des Ingenieurs zielendes Simulationssystem für technologische Prozesse.

1. Einleitung

Arbeiten zur Simulation diskreter Prozesse bildeten einen Schwerpunkt der Forschung an der Ingenieurhochschule Dresden (IHD). Vor allem vom Lehrkollektiv „Modellierung und Simulation“ initiiert und getragen, beteiligten sich daran auch die Lehrgebiete „Technologische Vorbereitung“ und „Echtzeitverarbeitung“. Den Anfang bildeten Studien an Teilprozessen der Produktion im VEB Plastikart, für die auf der elementaren Basis von FORTRAN getaktete Simulationsprogramme entworfen und experimentell analysiert wurden. Die Ergebnisse ermutigten zur Fortsetzung dieser Arbeitsrichtung, wobei im Weiteren das inzwischen im VEB Robotron entwickelte Simulationssystem SIMDIS – ein GPSS-Derivat –, auch im Sinne seiner Erprobung mit mehreren industriellen Partnern an unterschiedlichen Prozessen der Produktion und Informationsverarbeitung, genutzt werden konnte. Trotz der ausschließlich über Lochkarten erfolgenden Eingabe, erlaubte es eine wesentlich komfortablere Modellentwicklung und Experimentgestaltung und erreichte eine breite Anwendung in der DDR. Jedoch, Anwendungen dieser Softwarekategorie, das bestätigten unsere Erfahrungen mit SIMDIS, bedürfen eines Mittlers zum eigentlichen Nutzer, eines Simulationsexperten. Andererseits implizierten die erzielten Ergebnisse die Methode der Simulation für eine Anwendung unmittelbar am Arbeitsplatz z. B. des Projektierungsingenieurs als prädestiniert anzusehen. Das bedingt jedoch ein wesentlich einfacher zu handhabendes Simulationswerkzeug, worauf die weiteren Arbeiten zur Entwicklung eines eigenen, diesen Ansprüchen genügenden Simulationssystems für technologische Prozesse ausgerichtet wurden. Dessen Charakteristik soll hier dargestellt werden soll. Um seine wesentlichen Unterschiede im Vergleich zu Systemen wie GPSS bzw. SIMDIS verdeutlichen zu können, sollen zuvor wesentliche Merkmale des letzteren skizziert werden.

2. Charakteristik des Simulationssystems SIMDIS

Für die simulative Analyse eines Prozesses bedarf es stets eines mathematischen Modells, im Falle diskreter Prozesse in Form einer algorithmischen Beschreibung. Für die Entwicklung eines solchen Modells bietet das System eine Reihe vordefinierter Elemente und darauf bezogene Ereignisalgorithmen.

Elementeklassen:
Aktivatoren (temporär, mobil wie Werkstücke, Chargen, Jobs, Nachrichten usw.)
Permanente Elemente (Einrichtung, Speicher, Warteschlange u. a.)
Berechnungselemente (z. B. Zufallszahlengeneratoren., Verteilungs- u. a. Funktionen)

Aktivatorbezogene Ereignisalgorithmen in SIMDIS (eine Auswahl):

Bezeichnung	Parameter	Kommentare
GENERATE (Erzeugung von Aktivatoren)	A, B, C, D, E, F ... I	A und B: Ankunftsabstand C: Zeitpunkt für 1. Aktivator D: Maximalzahl Aktivatoren E: Prioritätsklasse F bis I.: Aktivator-Parameter
ADVANCE (Verzögerung)	A, B, C	A und B: Verzögerungsdauer C: Schranke für Ereigniszeitpunkt
ASSIGN (Änderung von Parametern)	A(+/-), B, C, D	A(+/-): Nr. des Aktivator-Parameters B: Wert, (zuzuweisen () , zu addieren (+) oder zu subtrahieren(-)

Ereignisalgorithmen (Blöcke) bezüglich permanenter Elemente z. B. :

Element	Bezeichner	Parameter	Funktion
Einrichtung	SEIZE RELEASE	A A	Einrichtung "A" belegen/ freigeben
Speicher	ENTER LEAVE	A, B A, B	In "A" B Plätze belegen bzw. freigeben
Warteschlange	QUEUE DEPART	A, B A, B	B Plätze in A belegen bzw. freigeben
Skalar (Variable)	SAVEVALUE	A(+/-), B, C	Wertänderung des Skalars "A" vom Typ "C"

Datenstruktur eines Permanenten Elements :

Als eine vorab definierte Datenstruktur sei die des Elements Speicher angegeben.

Merkmale des Speichers	Numerische S:	Zugeordnete logische S:
Nr. oder Name Aktueller Inhalt Zeitintegral Maximaler Inhalt Aktivatoreintritte	 SNj SAj, STj, SLj SMj, SCj	 SEN/SNE: leer/nicht leer SF/SNF: (voll, nicht voll) SV/SNV: (verfügbar, nicht verfügbar)

Die zugeordneten numerischen und logischen Symbole ermöglichen den Zugriff auf diverse Daten während des Simulationslaufs bzw. zur Auswertung.

Die Bezeichner der Algorithmen bilden die Elemente der Modellbeschreibungssprache. Das heißt, die Modellformulierung erfolgt durch eine Sequenz derartiger Benennungen und deren Parametrisierung. Zur Veranschaulichung sei ein simples Beispiel (der Übersichtlichkeit wegen tabellarisch) angegeben: Das stark vereinfachte Modell einer Tankstelle mit 3 Säulen.

Nr. / Name	Bezeichner	Parameter	Kommentar
	SIMULATE	15	Max. Rechenzeit
	INITIAL	SK#TANKS,3	Kapazität: 3 Tanksäulen
TANK	FUNCTION	RN3,C4 0,5/.1,10/.8,30/1,45	Verteilungsfunktion für Tankfüllung (Empirisch 4 Stützstellen)
1	FVARIABLE	PL1*3	Tankdauer: Füllmenge * 3
	GENERATE	DX1(30),,,,,PL1	Gleichverteilte Fahrzeug-ankünfte
	ASSIGN	1,FN#TANK,,,,,1PL	Tankfüllung → PL1
	QUEUE	TANKS	Einreihen in WS
	ENTER	TANKS	Tanksäule belegen
	DEPART	TANKS	Verlassen der WS
	ADVANCE	AV1	Verzögern um PL1*3
	LEAVE	TANKS	Tanksäule freigeben
	SAVEVALUE	LITER+, PL1, XL	Verkaufter Treibstoff
	TERMINATE	1	Aktivatoren-Senke
	START	5000	Abbruch nach 5000
	END		Aktivatoren.in der Senke

Der Parameter 1 von TERMINATE bewirkt, dass bei jedem die Senke erreichenden Operanden "1" vom "Startzähler" 5000 subtrahiert wird. Dieses Zusammenspiel der Blöcke TERMINATE und START zeigt, dass die Elementeklasse AKTIVATOR in SIMDIS auch zur Steuerung des Simulationsexperiments genutzt wird. Insgesamt umfasst das System 23 Blocktypen zur Modellbeschreibung und ca. 50 verschiedene Definitions- und Steueranweisungen.

3. Charakteristik des System TOMAS

3.1 Das Formalisierungskonzept

Wenngleich mit SIMDIS der Simulation in der DDR das Tor zu einer breiten Anwendung geöffnet wurde, blieb diese an die Mitwirkung eines mit dem System bestens vertrauten Mittlers gebunden. Andererseits erwiesen sich die Experiment-resultate z. B. für die Projektierung von Fertigungsabschnitten, Hochregallagern u. a. Objekten als sehr aussagefähig für deren Gestaltung, so dass die Verfügbarkeit eines einfach zu nutzenden Simulationswerkzeugs am Arbeitsplatz z. B. des Projektierungs-ingenieurs wünschenswert zu sein schien.

Es lag der Gedanke nahe, eine vereinfachte Modellierung auf der Grundlage eines dialogfähigen Systems zu erreichen, das über eine relativ geringe Anzahl von Bausteinen mit wesentlich komplexeren Funktionen verfügt. Sie sollten nicht elementare Ereignisse sondern an reale Objekte, wie Einzelmaschinen Maschinengruppen, Lager usw., gebundene Teilprozesse modellieren.

Es war klar, dass die Einfachheit derartiger Bausteinsysteme nur mit Verlust an Anwendungsbreite, der Beschränkung auf bestimmte Prozessklassen erkauft werden muss. Deshalb erschien es wichtig, der Bausteinentwicklung ein uniformes formales Konzept zugrunde zu legen, um eine Systemergänzung um neue, nutzerspezifische Bausteine jederzeit ohne Kompatibilitätsprobleme vornehmen zu können. Analysen anderer neben GPSS zu jener Zeit bekannt gewordener Simulationssoftware ergaben in dieser Richtung keine verwertbaren Aussagen. Eine gewisse Orientierung z. B. an der Bedienungstheorie war mitunter erkennbar aber kein homogenes formales Basiskonzept. Der Anregung einer 1974 erschienenen Publikation zu Problemen der algorithmischen Modellierung folgend[1], wandten wir uns der Automatentheorie zu und fanden in dem Begriff des Allgemeinen Stochastischen Asynchron-Automaten ein geeignetes Konzept zur uniformen formalen Bausteindarstellung.

Die nachfolgende Skizzen (Bild 1 und Bild 2) verzeichnen die Bestimmungsgrößen und das Reaktionsschema dieses Automatentyps auf ein eintreffendes Eingangssignal.

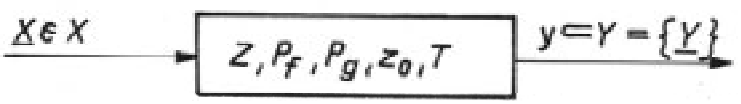


Bild 1: Bestimmungsgrößen

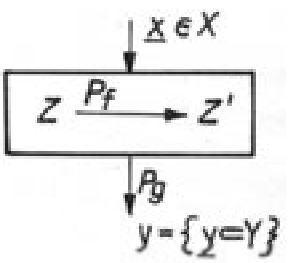


Bild 2: Reaktion des Automaten

[1] Bakajew; Kostina; Jarowizki: Algorithmische Modellierung ökonomischer Probleme, Akademie Verlag Berlin 1974

Es bedeuten:
X: Eingangssignalmenge, Y: Ausgangssignalmenge, Z: Zustandsmenge,
P_f: die Überführungsfunktion, Pg: die Ergebnisfunktion, z_0: Anfangszustand,
T: Zeitmenge, (die x, y, und z sind Vektoren)

Die Wirkung der Funktionen P_f und P_g beschreiben die folgenden Formeln, sie bilden zusammen den Automatenalgorithmus. P_f bewirkt einen zeitlosen Zustandsübergang.

$$\underline{z}(t+\varepsilon) = P_f(\underline{z}(t), \underline{x}(t)); \quad \underline{z} \in Z, \quad \underline{x} \in X, \quad \varepsilon \to 0$$

Bild 3: Überführungsfuntion

$$y(t+\varepsilon') = P_g(\underline{z}(t+\varepsilon), \underline{x}(t)); \quad \varepsilon' > \varepsilon, \quad \varepsilon' \to 0, \quad \varepsilon \to 0, \quad y \subset Y$$

Bild 4: Ergebnisfunktion

Pg erzeugt eine Menge von Ausgangssignalen (Teilmenge von Y)

3.2. Operatoren und Operanden

Die formalen Merkmale des Automaten nehmen für unterschiedliche Bausteintypen, in TOMAS Operatoren (Or) genannt, eine spezifische Ausprägung an. Beispielsweise hat ein „Bedienungsoperator“ einen Parameter- und einen Zustandsvektor Z = (Z1, Z2), (s. Tabelle u.). Neben den als Operatoren bezeichneten Modellbausteinen existiert in TOMAS eine Operanden (Od) genannte Klasse zur Abbildung "passiver" Prozesselemente. Sie existieren analog den SIMDIS-Aktivatoren als temporäre Datenstrukturen, aber ohne Steuerungsaufgaben.

Die Kopplung der Bausteine eines konkreten Modells erfolgt über die beim Modellentwurf gemäß dem Operandenfluss anzugebenden Nachfolgeroperatoren. Somit stellt ein TOMAS-Modell ein in Abhängigkeit vom Operandenfluss strukturiertes Operatorennetz dar.

Parameter und Hauptattribute Z_1:	Zusatzattribute Z_2: Zur lokalen WS
- Name - Anzahl der Bedienungskanäle - Kapazität der lokalen WS - Bedienungsdauer (i. a. eine Funktion) ..... - Anzahl der Operandeneintritte - Warteschlangen-Zeitintegral - Bedienungskanal-Zeitintegral..... - Unterbrechungszeitintegral	- Nr. des 1. Operanden - Typ desselben - nächste Ereigniszeit ..... - Nr. des 2. Od - Typ desselben - Nächste Od-Ereigniszeit .....

Operatoren:
- sind Instanzen/Objekte (Inkarnationen) vordefinierter Operatorentypen,
- modellieren die Einwirkung der realen permanenten Objekte des zu simulierenden Systems wie Maschinen, Transportgeräte, Prozessoren, Übertragungskanäle usw. auf dessen temporäre Objekte,
- realisieren signalabhängig den Operandenfluss.

Operanden
- modellieren vorrangig die temporären Objekte des betrachteten realen Systems wie Werkstücke, Chargen, Jobs, Nachrichten usw.
- gegebenenfalls aber auch die für bestimmte Operationen benötigten, beschränkt verfügbaren mobilen Ressourcen wie Vorrichtungen, Werkzeuge usw.

Die Modellentwicklung besteht in der Komposition eines systemadäquaten Operatorennetzes über die Nachfolgerbenennung der Bausteine und deren Parametrisierung durch den Nutzer.

3.3 Das Baustein- und Funktionsangebot

Das Standardangebot an Modellbausteinen zeigt die folgende Tabelle.

Benennung	Kurzbez.	Abbildungsfunktion
Operandengenerator I	GENO	Autonome Erzeugung von Od
Operandengenerator II	GENA	Abhängige Erzeugung von Od
Bedienoperator	BEMM	Zeitverbrauchende Teilprozesse
Zuordnungsoperator	ZUOR	Verknüpfung von Od (Montage)
Dispositionsoperator	DISP	Verwaltung/Zuteilung speziell. Ressourcen
Packetieroperator	PACK	Bildung von Od-paketen / -Senke
Kopieroperator	KOPI	Vervielfachung von Od (Klonen)
Vereinigungsoperator	VERE	Vereinigung von Od-strömen
Verzweigungsoperator	VERZ	Verzweigung eines Od-stroms
Speicheroperator	SPEI	Speicher für Od (Lager)
Störsignalgenerator	GENS	Determinierte bzw. stochastische Unterbrechungen

Der 1. Schritt der Modellentwicklung ist die Definition der Modellstruktur. Z. B. von einem Prozessgraphen ausgehend (Bild 5) sind im Dialog für die einzelnen Teilprozesse der Bausteintyp zu wählen, zu benennen und der Nachfolger anzugeben.

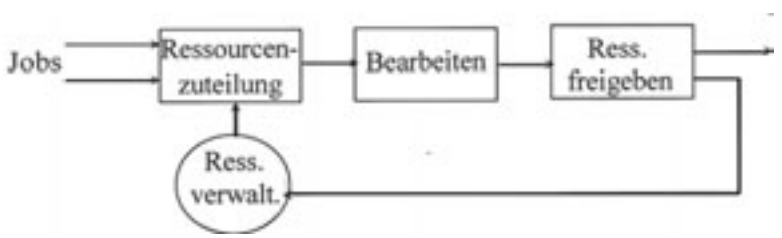


Bild 5: Strukturgraph

Das Ergebnis ist ein Modellgraph wie im Bild 6 für den skizzierten einfachen Prozess.

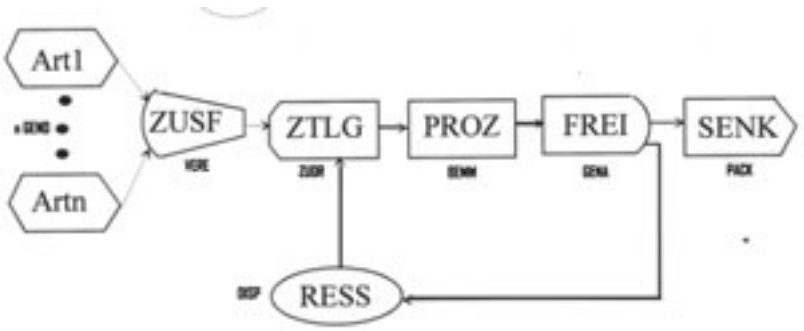


Bild 6: Modellgraph

Für die Belegung der Parameter der aus dem Bausteinmenü gewählten Operatoren, die auf der Basis von Formularen erfolgt, wird ein Menü vordefinierte Funktionen angeboten, und zwar
- Theoretische Verteilungsfunktionen
- Tabellarische Funktionen z. B. für empirische Verteilungen
- Verzweigungsfunktionen
- Kopierfunktionen
- Prioritätsfunktionen
- Unterbrechungsfunktion

Auch der Definition dieser Funktionen wurde ein formales Schema zugrunde gelegt und zwar in Form weniger abstrakter Datenstrukturen.

3.4. Interne Steuerung des Simulationslaufes

Wie in SIMDIS und den meisten anderen Systemen wird der Ablauf der Simulation ereignisorientiert auf der Grundlage einer Modelluhr gesteuert, allerdings nicht über die den Aktivatoren analogen Operanden sondern mittels der im Modell von den verwendeten Bausteinen (Automatenobjekten) erzeugten Signale.

Auf der Vereinigungsmenge X U Y wurden folgende Signalklassen definiert:

1. Kalendersignale

- Sie dienen der Nachbildung Zeit verbrauchender Operationen durch Planung zukünftiger Ereignisse (nächste Ankunft eines Operanden, nächster Störbeginn, Ende einer Bearbeitung oder Störung usw.).
- Sie werden im „Kalender“ notiert, in dem je Zeit verbrauchendem Operator ein Platz vorhanden ist. Später liegende, schon erzeugte Kalendersignale werden lokal im Operator verwaltet. (Beschleunigung der Simulation durch partiell dezentrale Signalverwaltung!)

Dem Signal aufgeprägte Informationen sind Absender, Empfänger, Ereigniszeit, Ereignistyp.

2. Momentane Ereignissignale
- Sie werden zum aktuellen Modellzeitpunkt einer „Signalkette“ (SK) genannten Liste übermittelt.
- Ihre Einordnung in die Signalkette erfolgt prioritätsabhängig. Blockierungssignale haben die höchste , Übergabesignale die zweithöchste Priorität, alle übrigen momentanen Signale werden dahinter, jeweils am Ende der SK eingeordnet werden.

Insgesamt wurden 7 Typen momentaner Signale definiert:
- Übergabesignale, bewirken die Übergabe eines Operanden an einen Nachfolger.
- Blockierungssignale, an Vorgänger gerichtet, verbieten weitere Übergabe von Od.
- Störsignale, vom zugeordneten GENS kommend, zeigen Störungsart, Anfangs- und Endzeitpunkt der Störung an.
- Markierungssignale, von Operandensenken (PACK oder DISP) an GENO oder GENA gerichtet, zeigen das Ende der Existenz eines Operanden der Quelle an.
- Bestellsignal, von ZUOR an DISP (Anforderung von Hilfsoperanden / Ressourcen) für einen Hauptoperanden.
- Liefersignale, von DISP an ZUOR gerichtet (Anzeige der Lieferung).
- Sondersignale, für spezielle Zwecke, z. B. Losbildung, verwendet.

Arbeitsweise des Aktivierungsalgorithmus (Bild 7): Die Steuerung der Simulation des durch das Modell abgebildeten Prozesses, verdeutlicht diese Schrittfolge:

1. Das erste Signal in der Signalkette wird an dessen Empfänger übermittelt.
2. Der Empfänger wird aktiv und erzeugt gegebenenfalls neue Signale, die in die Signalkette oder in den Kalender eingetragen werden.
3. Solange die Signalkette nicht leer ist, folgt Schritt 1, andernfalls Schritt 4.
4. Aktivierung des Kalenders,

4.1. Suche des frühesten Kalendersignals, Aktualisierung der Modelluhr und Übermittlung an dessen Empfänger, der damit aktiviert wird. und gegebenenfalls momentane oder zukünftige Ereignisse erzeugt und übermittelt.

4.2. Warten auf erneute Aktivierung bei leerer Signalkette

Der Simulationslauf endet, wenn keine Signale mehr anliegen, spezielle Steueranweisungen sind nicht erforderlich.

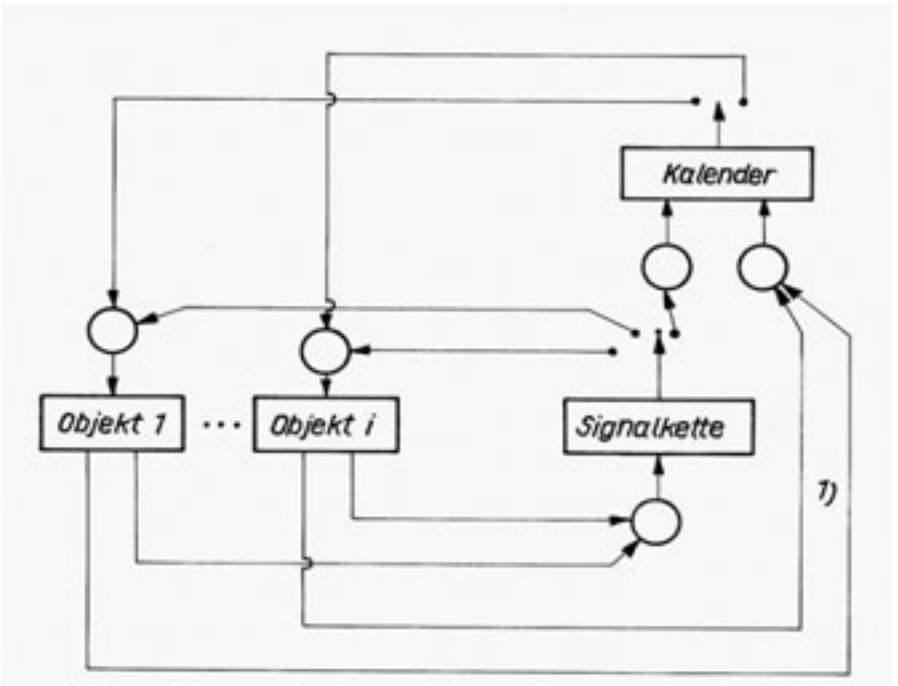


Bild 7: Aktivierungsalgorithmus

1) Kalendersignale kommen nur von aktiv verzögernden Objekten, andere, nur mit der SK korrespondierende Objekte sind in der Skizze nicht verzeichnet.

3.5 Systemarchitektur, Implementierung und Anwendung

Mit Bild 8 soll die Architektur des Gesamtsystem TOMAS verdeutlicht werden. Dazu ist, wie schon angedeutet, anzumerken, dass sowohl die Benutzer- wie die Administratoroberfläche anfangs auf einem programmierten Dialog über Formulare mit nur quasigraphischer, also per Drucker realisierter Darstellung der Modellstruktur und ausgewählter Resultate basierte. Graphikfähige Technik stand nicht zur Verfügung.

Die Implementierung erfolgte aus pragmatischen Gründen auf ESER-Computern mittels der Programmsprache FORTRAN. Compiler einer objektorientierten Programmiersprache, deren Nutzung - wie auch ein Versuch mit SIMULA auf einem nur singulär zugänglichen Computer BESM 6 zeigte - nahe gelegen hätte, waren allgemein nicht verfügbar. Die genannten Beschränkungen bedingten zwangsläufig einen immensen Aufwand für die Realisierung des Konzepts.

Um die Mitte der 80-er Jahre war die Standardversion so weit gediehen, dass sie im Rahmen von Praktika zum Lehrfach „Modellierung und Simulation“ einem intensiven Test unterzogen werden konnte. Danach wurde TOMAS einer ganzen Reihe von interessierten industriellen Anwendern und einigen Hochschulen zur Erprobung bzw. Nutzung mit teilweiser Unterstützung durch Diplomanden übergeben.

Partner waren z. B. die Unternehmen Planeta Radebeul, Zentrum für Mikroelektronik Dresden (ZMD), Schaltgerätewerk Oppach, Elektromotorenwerk Dessau, Armaturenwerk Prenzlau, Sektion Elektroniktechnologie der Humboldt-Universität. Das DVZ Neubrandenburg übernahm das System TOMAS zur kommerziellen Aufbereitung und zum Vertrieb.

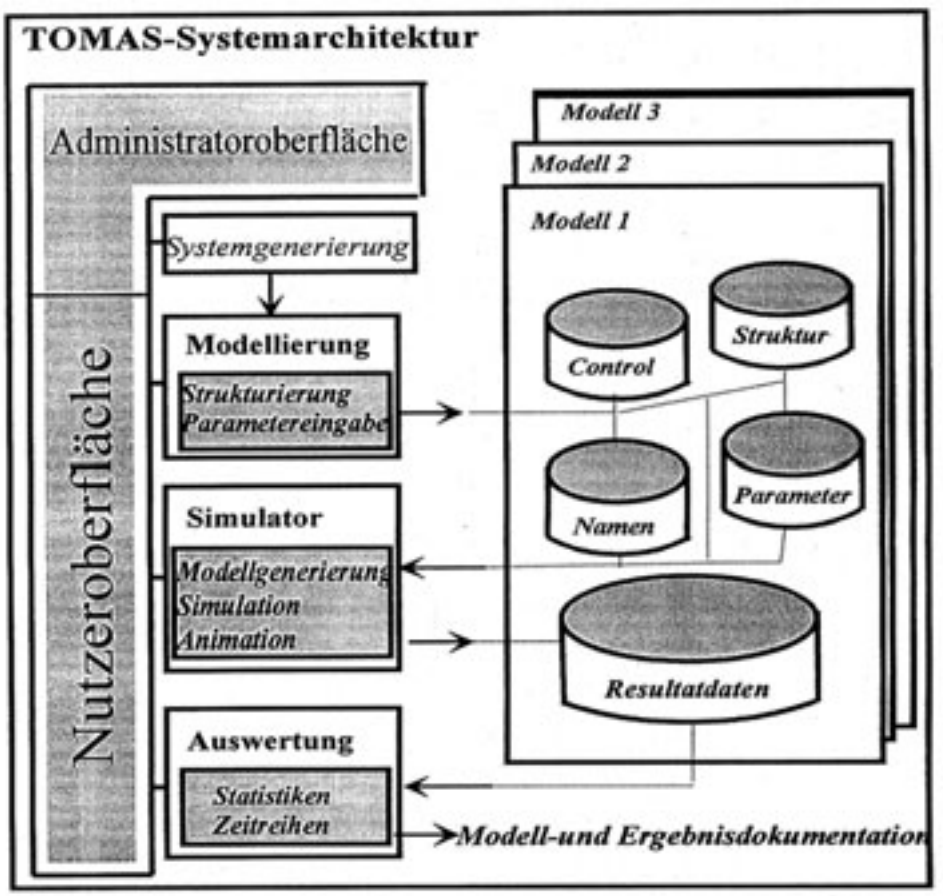


Bild 8: Systemarchitektur

Das Urteil der beteiligten Ingenieure der Partnerunternehmen war außerordentlich positiv, vor allem die Transparenz der Modellstrukturentwicklung und die leichte Erlernbarkeit des Umgangs mit dem System wurden hervorgehoben.

Ein nahe liegender Gedanke kam allerdings auch zu Tage: Die Parametrisierung des Modells durch Kopplung an die betriebliche Datenbasis (Arbeitsplan-Stammdaten z. B.) weiter zu vereinfachen und zu beschleunigen. In Zusammenarbeit mit ZMD wurden entsprechende Arbeiten zur (Teil-)Automatisierung der Modellentwicklung, einschließlich der Modellstruktur, in Angriff genommen. Diese zielten zugleich dahin, TOMAS-Modelle auch für die Steuerung von Abschnitten der Schaltkreisfertigung einzusetzen, konnten aber nicht mehr zu Ende geführt.

Schon 1990 begann der Überlebenskampf der industriellen Forschungspartner in dessen Folge dieser Teil der Finanzierung unserer Arbeiten und jegliche Zusammenarbeit entfielen. Wenig später folgte die radikale Personalreduzierung an Universitäten und Hochschulen und deren Einordnung in das BRD-System der Forschungsförderung. Die beantragte Aufnahme unserer Thematik in die Förderung durch die DFG wurde abschlägig beschieden. Ein maßgeblicher Gutachter der DFG und Fachkollege empfahl „wohlwollend“ ein Subthema in seiner Forschungsrichtung zu übernehmen. Infolge dieser Umstände reduzierten sich die Arbeiten an TOMAS auf die Umgestaltung der Nutzer- und Administratoroberfläche in echt graphische sowie die Entwicklung einer weiteren Generierungskomponente für die Anpassung der Oberfläche selbst an spezielle Nutzeranforderung. Auch die Einbindung von Möglichkeiten der Animation als Mittel der Modellvalidierung wurde begonnen. Schließlich kamen diese Forschungsarbeiten vollständig zum Erliegen.

4. Resümee

Die zuletzt geschilderten Umstände verhinderten den endgültigen Nachweis der möglichen Integration der Simulation in die "Werkzeugkiste" des komplexe Systeme projektierenden oder steuernden Ingenieurs mittels eines adäquaten, einfach zu nutzenden Simulationswerkzeugs. Bemühungen in dieser Richtung hat es begrenzt auch anderweitig gegeben. Sie begnügten und begnügen sich im Wesentlichen mit der Einfügung parametrisierbarer Modelle (z, B. für Hochregallager, Engpassanalysen von Maschinengruppen, Halbleiterfertigung) in Fabrikplanungssysteme zum Beispiel. Neu bzw. deutlich verbessert erscheinen die auf die Modelle bezogenen Visualisierungs- bzw. Animationsmöglichkeiten sowie die Methoden der statistischen Auswertung und Darstellung der Resultate. Die Integration flexiblerer komplexer Simulationskomponenten in die Werkzeuge der Computer unterstützten Ingenieurarbeit steckt anscheinend weiterhin in den Anfängen.

Auf dem Gebiet der Simulationssoftware allgemein sind in den letzten 15 Jahren auf Basis der rechentechnischen Entwicklung bedeutende Fortschritte in Richtung graphischer Oberfläche und Animation, statistischer Auswertung und Darstellung der Simulationsergebnisse sowie der Nutzung objektorientierter Sprachen zur Implementation erzielt worden. Auf der Grundlage ihrer objektorientierten programmierungs-technischen Umsetzung verfügen einige Bausteinsysteme auch über Möglichkeiten zur Modifikation der modellierenden Elemente, bzw. der Programmierung neuer. Neben dieser, die Nutzerfreundlichkeit der Simulationssysteme erheblich steigernden, hat sich eine drastische quantitative Entwicklung vollzogen.

Neben den damals schon bekannten sind etliche allgemeine und fachgebietsorientierte, in konzeptioneller Hinsicht konservative Simulationssysteme auf dem Markt, mit denen Simulation weiterhin im Sinne von Dienstleistung durch eine Vielzahl (meist kleinerer) Unternehmen angeboten und realisiert wird. Bemühungen in Richtung einer theoretisch fundierten abstrakten Beschreibung der modellierenden Konstrukte hat es offensichtlich kaum gegeben. In einem Fall kann man vermuten, dass die Idee, Simulationssoftware auf Basis eines wohl definierten, aus der Automatentheorie abgeleiteten formalen Bausteinkonzepts zu entwickeln, Nachahmung gefunden hat. Zu erwarten gewesen wäre wohl in dieser Zeit auch die breitere Nutzung echter Parallelverarbeitung anstelle der quasi parallelen rechnerischen Prozessnachbildung, wozu es im Rahmen der hier dargestellten Arbeiten bereits erste Ansätze gab.

Wenngleich die Ziele der Forschungsarbeiten zur Simulation an der IHD und ab 1986 im Informatikzentrum der TUD nur partiell erreicht wurden, dürften sie nicht ohne Nachhaltigkeit gewesen sein. Durch ein Lehrbuch und eine Vielzahl von Publikationen trugen sie zur Publizität der Computersimulation und zur Wissenschaftsentwicklung bei. Von unschätzbarem Wert war die Tatsache, mit dem eigenständig entwickelten Simulationssystem, eine bis aufs letzte Bit transparente Quelle für Diplomthemen und andere Graduierungen zur Verfügung zu haben. Etwa 100 Diplomthemen wurden vergeben u. a. zur Erprobung an praktischen Aufgaben sowie zu Problemen der algorithmischen Modellierung, der Oberflächengestaltung, der Beschleunigung des Simulationslaufs, einer effektiven Implementierung, einer aussagefähigen Ergebnisdarstellung usw. Darüber hinaus entstanden im Rahmen dieser Arbeitsrichtung 18 Dissertationen und 3 Habilitationsschriften. Somit wurde gewiss ein nennenswerter Beitrag zur Heranbildung hoch qualifizierter, überwiegend noch heute in Deutschland tätiger Informatiker geleistet.

Literaturverzeichnis

[1] Frank, M.: Ein automatentheoretisches Modellkonzept für ein fachgebietsorientiertes Simulationssystem. Dissertation B, TH Ilmenau 1979

[2] Frank, M.: A Process-oriented Modelling Concept for Special Simulation Systems. In: Discrete Simulation and Related Fields. North-Holland Publishing Company, Amsterdam - New York - Oxford 1982

[3] Frank, M.; Neuthe, S.: Fertigungssysteme modellieren und simulieren. Zeitschrift für wirtschaftliche Fertigung und Automatisierung. 85(1990)6, S. 326 - 330

[4] Strohmeyer, S.: Probleme der Gestaltung und Implementation "offener" Simulationssoftware. 15 Jahrestagung - Grundlagen der Modellierung und Simulationstechnik; Rostock 1986, Tagungsbericht S. 5 - 8

[5] TOMAS-Anwenderdokumentation. Teil I: Handbuch für die Modellentwicklung, Teil II: Hantierung, TU Dresden 1991

[6] Bersiner, S.: Neukonzipierung der Generierungskomponente des Modellierungs- und Simulationssystem TOMAS. Diplomarbeit, TU Dresden 1994

[7] Frank, M.: Ein Simulationswerkzeug für die Ausbildung. Proceedings zum 5. Symposium: Simulation als betriebliche Entscheidungshilfe - Neuere Werkzeuge und Anwendungen aus der Praxis. Braunlage 1995, S. 223 - 240

Forschungen zum computergestützten Unterricht am Forschungs- und Rechenzentrum der Akademie der Pädagogischen Wissenschaften der DDR (FRZ der APW) und an der TU Dresden aus Sicht einer tätigkeitsorientierten Modellierung

Joachim Oelschlegel Horst Kreschnak

CAD-Systemhaus Dr.Oelschelgel
GadelsdorferWeg 18 Einsteinstr. 6
01328 Dresden 01445 Radebeul
JOelschlegel@CAD-Systemhaus.de horstkreschnak@arcor.de

Abstract: Das FRZ der APW beschäftigte sich seit 1972 mit der Modellierung dispositionell erklärbarer Verhaltensweisen, die sowohl für die methodische Unterrichtsgestaltung als auch für den damals beginnenden Einsatz im Computergestützten Unterricht geeignet waren. Mit großem Aufwand wurden Resultate von Leistungskontrollen analysiert, wissenschaftlich erklärt und experimentell bestätigt. Invarianten von Tätigkeitsresultaten konnten auch in der Grundlagenausbildung Mathematik an der TU Dresden ermittelt werden. Damit wurden Grundlagen für eine Verbesserung von Inhalt und Methodik des Lernstoffes geschaffen. Die Ergebnisse der Forschung sind mit der Auflösung des FRZ der APW 1990 verloren gegangen.

1 Zur Historie und Einordnung des Themas

Das FRZ der APW wurde 1970 gegründet. Direktor war bis 1990 Prof. Horst Kreschnak. Das FRZ hatte zwei Kernaufgaben:

Eine der beiden Kernaufgaben des FRZ war die Grundlegung für inhaltlich-strategische Entscheidungen zum Einsatz von Rechentechnik in der Unterrichtsforschung und später auch im Unterricht selbst (CUU).

Die Ziele zum CUU weisen

a) in eine technisch-praktische Richtung: Entwicklung einer Programmiersprache zur Beschreibung der Interaktionen von Schüler, Lehrstoff und Lehrkraft und

b) in eine inhaltlich-theoretische Richtung: der Modellierung des schulischen Lernens.

1972 wurde im Ergebnis der Programmiersprachenentwicklung am R300 ein LISP-Interpreter (Herbert Stojan) erfolgreich zum Einsatz gebracht. Joachim Oelschlegel modellierte mit diesem Interpreter erste Lehrprogramme zum Lehrfach Physik, gemeinsam mit Dr. Helmut Menschel.

Der Anspruch der Modellierung des schulischen Lernens (der noch heute offen ist) wurde durch einen scheinbar elementaren Schritt in Angriff genommen: dispositionelle Erklärung und Voraussage am Beispiel von Leistungskontrollergebnissen im Fach Mathematik von Schülern der 5. und 6. Klasse. Diese Modellierung entwickelte Kreschnak mit einer Gruppe des FRZ und der PH Dresden mittels eines mehrwertigen und mehrstufigen prädikatenlogischen Ansatzes. Empirische Basis waren tausende von Leistungskontrollen, die rechentechnisch erfasst und ausgewertet worden sind. Die dispositionellen Erklärungen stellten Erklärungen schriftlicher Schülerergebnisse auf der Grundlage von gesetzmäßigen Zusammenhängen zwischen Dispositionen und Symptomen bzw. zwischen Fehldispositionen und Symptomen dar. Zutreffende dispositionelle Erklärungen ermöglichen zutreffende Voraussagen entsprechender Ergebnisse bei erneuten Leistungskontrollen.

Dieses Vorgehen konnte Joachim Oelschlegel durch den Wechsel 1977 vom FRZ der APW an das Forschungszentrum für technische Lehr- und Lernmittel an der TU Dresden (Ltr. Prof. Bannwitz), an komplexeren Inhalten bestätigen. Tausende von Klausuren des Grundlagenstudiums Mathematik wurden analysiert. Die fehlerhaften Ergebnisse der Studenten konnten, analog zu den Schülern, durch dispositionelle Verhaltensmuster zu über 85% dispositionell erklärt und vorausgesagt werden.

Damit sind empirische Grundlagen gelegt worden, die fernab der damals kausalistischen (behavioristischen) Erklärung und Beschreibung des Lernens lagen. Es war der Eintritt über die dispositionelle Erklärung von Tätigkeiten hin zu einer Modellierung des Handelns und Lernens.

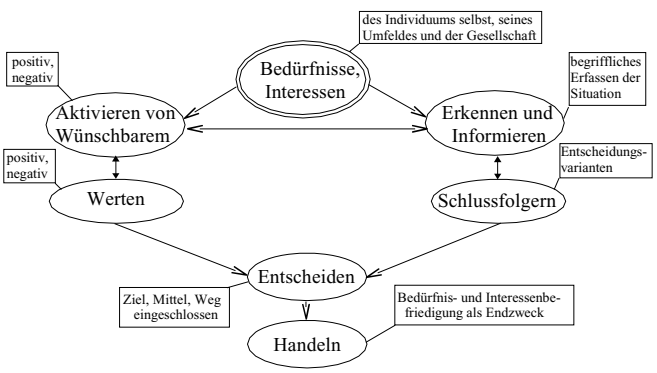


Abbildung 1: Zyklus – Bedürfnisse-Erkennen-Entscheiden-Handeln

2 Besonderheiten von Diagnose-Therapie-Modellen

2.1 Historisches zu den Grenzen des Experiments

In dem Wissenschaftsverständnis, das lange Zeit vorherrschte und beginnend mit der europäischen Renaissance unter starkem Einfluss solcher Naturwissenschaften wie der Physik sich herausbildete, nimmt das Experiment als gezielter Eingriff in die Natur einen hervorragenden Platz ein.

In einer experimentell erzeugten Situation werden der Beobachtung unmittelbar zugängliche Größen systematisch variiert, um den Einfluss dieser Größen auf andere beobachtbare Größen genau registrieren zu können. Mögliche andere Einflussfaktoren sind dabei konstant zu halten. Das Experiment als eine Art praktischer Tätigkeit, die zu den Aktivitäten theoretischer Vernunft bzw. zu wissenschaftlichen Aktivitäten gehört, ist klar abgrenzbar von einer anderen Art praktischer Tätigkeit, die als Entscheidungsrealisierung zu den Aktivitäten praktischer Vernunft zu rechnen ist (ausführlich in [KRE, 2006/ 2007] dargestellt und logisch-mathematisch modelliert).

Experimente dienen eindeutig dem Vordringen theoretischer Vernunft zur vollkommeneren Wahrheit. Entscheidungsrealisierung dagegen ist darauf gerichtet, die Erwartungen, die das Entscheidungssubjekt mit der von ihm getroffenen Entscheidung verbindet, zu erfüllen. Diese Erwartungen betreffen vermittelt oder unvermittelt der Befriedigung menschlicher Bedürfnisse oder Interessen, sind also kein Vordringen zu immer vollkommenerer Wahrheit, sondern sind auf Nutzen für das menschliche Entscheidungssubjekt oder – wie es in der Theorie rationalen Entscheidens heißt – auf das Erreichen von positiv Wünschbarem oder das Abwenden von negativ Wünschbarem gerichtet.

Das Experiment hielt auch Einzug in solche empirische Wissenschaften wie Biologie, Soziologie, Psychologie usw. Zwar hat es dort zu beachtlichen Erkenntnisfortschritten beigetragen, stieß aber zugleich an Grenzen. Erst als in der Physik, an der sich lange Zeit andere Wissenschaften orientierten, kritische Fragen zur Wechselwirkung zwischen Beobachtetem und Beobachtenden auftraten, wurden die mögliche Begrenztheit auch des Experiments sichtbar.

Seit den letzten Jahrzehnten des zwanzigsten Jahrhunderts gewinnen Probleme bei Erforschung komplexer Systeme, speziell auch Probleme bei Erforschung sich selbst organisierender Systeme immer größere Bedeutung. Dabei wird deutlich, dass die Möglichkeiten, komplexe Systeme mit experimentellen Methoden zu erforschen, relativ rasch an Grenzen stoßen. Zwei Umstände, durch die Fragen nach genauerer Erforschung komplexer Systeme auf nicht-experimentellen Wegen immer dringlicher werden, seien hier genannt:

- Vielfach ist es unmöglich, bei Erforschung von Prozessen, die sich innerhalb komplexer Systeme vollziehen, nur die unmittelbar interessierenden Faktoren bzw. Größen zu variieren und andere mögliche Einflussfaktoren konstant zu halten. Gründe sind: Die verschiedenen Faktoren sind zu eng miteinander verknüpft, ihre Isolierung voneinander faktisch unmöglich. Bei sich selbst organisierende Systeme kennzeichnender Nicht-Linearität gilt, dass weder das schwache Kausalprinzip – gleiche Ursachen haben gleiche Wirkung – noch das starke Kausalprinzip – ähnliche Ursachen haben ähnliche Wirkungen – durchgängig zutrifft, vielmehr können kleine Abweichungen durchaus große Abweichungen zur Folge haben. Schließlich ist oftmals eine der interessierenden Größen oder einer der interessierenden Faktoren nur von Daten über unmittelbar beobachtbare Symptome ausgehend nicht zugängig. Als Beispiel seien Dispositionen genannt, auf deren Existenz nur von Daten über unmittelbare beobachtbare Symptome ausgehend zu schließen ist.
- Handelt es sich bei den zu erforschenden komplexen Systemen um Menschen oder um Objekte des gesellschaftlichen Lebens, ist Experimentieren oft als unmenschlich vollkommen abzulehnen.

2.2 Diagnose-Therapie-Modelle als ein Weg bei Erforschung komplexer Systeme

Ohne Zweifel spielt die Trennung theoretischer von praktischer Vernunft, die zugleich klare Trennung experimentellen Vorgehens von auf positiv Wünschbares oder Abwenden des negativ Wünschbaren gerichtete Entscheidungsrealisierung ist, geschichtlich eine kaum zu überschätzende positive Rolle. Wenn sich die Menschen vom Zutreffen der auf Erkenntnissen über Gesetzeszusammenhänge beruhenden Voraussagen getrennt vom zweckmäßigen praktischen Handeln überzeugen, erweitert das die Möglichkeiten dieses praktischen Handelns beträchtlich; denn: Eine Entscheidung muss innerhalb einer zeitlich begrenzten – oftmals sogar stark begrenzten – Entscheidungssituation getroffen und verwirklicht werden, wenn der beabsichtigt Zweck tatsächlich erreicht werden soll. So fehlt in Entscheidungssituationen die Zeit, Zweifeln gründlich nachzugehen, bessere Lösungsmöglichkeiten umfassend zu erwägen und zu erproben. Aufgrund dessen ist zu fragen, wie bei komplexen Systemen, bei denen sich Experimente verbieten, trotzdem die großen Vorteile, die Experimente bieten, weitgehend erhalten werden können. In die medizinische Forschung eingeschlossene Tierversuche stellen z.B. eine Teilantwort auf diese Frage dar. Problematisch ist an diesen Versuchen nicht nur, ob die damit verbundene Art, mit Tieren umzugehen, immer verantwortbar ist. Tierversuche bieten weiterhin nur unvollkommene Teillösungen, da die Möglichkeit der Übertragung der Erkenntnisse auf den Menschen nicht einfach zu beantworten ist. Vor allem aber kommen bei Erforschung komplexer Systeme Tierversuche als Ausweichlösung oftmals überhaupt nicht in Betracht.

Der Entwicklung der Mathematik, besonders auch der mathematischen Logik, und die Entwicklung moderner Informationsverarbeitungstechnik ist zu danken, dass es eine umfassend positive Antwort zur Frage nach dem Experimentersatz bei Erforschung komplexer Systeme gibt. Praktisches Handeln, das als Realisieren einer Entscheidung unmittelbar oder mittelbar dem Befriedigen menschlicher Bedürfnisse oder Interessen dient, kann so mit detailliertem Erfassen von Daten und logisch-mathematischer Modellierung verbunden werden, dass sich die gewonnenen Daten ähnlich wie experimentelle Befunde zu Vervollkommnung oder Korrektur wissenschaftlicher Einsichten nutzen lassen. Mit Hilfe ein und derselben praktischen Tätigkeit werden so Aktivitäten praktischer Vernunft realisiert und Probleme theoretischer Vernunft bewältigt.

Bei der folgenden Erläuterung der Einheit von praktischer und theoretischer Vernunft beim Entwickeln und Korrigieren von Diagnose-Therapie-Modellen soll nicht verschwiegen werden, dass sich schriftlich fixierte Schülerresultate aus Mathematik-, Physik, Chemie- und Sprachunterricht zur Entwicklung von Diagnose-Therapie-Modelle besonders eignen. Begriffliche Darstellungen empirischer Daten erweisen sich u. U. als außerordentlich schwierig. Das ist dann der Fall, wenn erforderliche Korrekturen der mathematischen Modelle zugleich Korrekturen in sich einschließen, die das begriffliche Erfassen der empirischen Daten betreffen. Dadurch können sich früher erfasste Daten für weitere Modellüberprüfungen als ungeeignet herausstellen. Indem die Schülerresultate faktisch unverändert als empirische Daten dienen, kann der genannten Schwierigkeit aus dem Weg gegangen werden. Das erleichterte die Modellentwicklung, besonders auch das Konfrontieren korrigierter Modelle mit empirischen Daten, auch mit empirischen Daten zurückliegender Schülerergebnisse.

Die Fragen des Verhaltens und der Tätigkeiten von Lehrer und Schülern, Lehren und Lernen während des Unterrichts sind unmittelbar Fragen praktischer Vernunft. Das Lehren erweist sich als rational, wenn es dem erfolgreichen Lernprozess der Schüler dient. Dabei sind alle zu rationalem Entscheiden und Handeln gehörenden Aspekte zu beachten. Der Lehrer muss u. a. von seinem pädagogisch-psychologischen Wissen, von seinen Erfahrungen im Umgang mit den einzelnen Schülern und der gesamten zu unterrichtenden Klasse Gebrauch machen, um die Schüler zum gewünschten Unterrichtsziel zu führen. Für die Schüler muss es für den Lernprozess einen hinreichend starken Handlungsgrund geben, sie sollen ihr bisheriges Wissen im betreffenden Unterrichtsfach aktivieren usw. usf. Kurz gesagt: Der Unterricht einschließlich der schriftlich zu lösenden Schüleraufgaben ist ganz normaler Unterricht wie jeder andere Unterricht auch. Die Daten aus den Schülerarbeiten werden dazu genutzt, Diagnose-Therapie-Modelle aufzustellen und diese immer wieder mit Daten aus neuen Schülerarbeiten zu konfrontieren. Bei den Diagnosen handelt es sich um Aussagen über Fehldispositionen, die bei oft auftretenden Schülerfehlern als Grund angenommen werden. Wird eine Fehlerdisposition eines Schülers zutreffend erfasst, so führt die der betreffenden Diagnose zugeordnete Therapie zum Erfolg.

Diese Verknüpfung von Diagnose und Therapie und die Erprobung der Therapie ist der Schritt im Modellierungsprozess, der die größte Ähnlichkeit mit einem Experiment aufweist. Hier ist besonders der Vertreter der Didaktik des betreffenden Unterrichtsfachs und eventuell seine Arbeit mit einzelnen Schülern gefragt.

Hier ist auch computergestützter Unterricht unter direkter Kontrolle des Didaktikers möglich. Hat sich das Wissen über den Zusammenhang von Diagnose und Therapie hinreichend bewährt, ist damit zugleich der Weg zur Empfehlung prophylaktischer Maßnahmen geebnet, durch die künftig der bisher häufig auftretende Schülerfehler weitgehend vermieden wird. All die eben aufgeführten Aktivitäten – beginnend mit Datenerfassung und ersten Modellierungsversuchen und endend mit der Empfehlung für prophylaktischer Maßnahmen – sind Aktivitäten theoretischer Vernunft, speziell Aktivitäten der Didaktikforschung im betreffenden Unterrichtsfach.

Vermutlich erkennen auch Laien auf den Gebieten von Wissenschaftstheorie und Medizin, dass es in der Medizin bei Entwicklung von Diagnose-Therapie-Modellen ein in mancher Hinsicht ähnliches Vorgehen wie in Didaktiken der Unterrichtsfächer gibt. Ohne dass dies hier näher ausgeführt werden kann, sei auch auf erste bescheidene, trotzdem aber vergleichbare Versuche vorsichtigen Diagnostizierens und eines gewissermaßen tastenden Therapierens im größeren Maßstab auf politisch-wirtschaftlichem Gebiet verwiesen.

Schließlich sei ein Problem dargestellt, bei dem es nicht nur darum geht, dem Pädagogen Hilfen zu bieten, durch die er Prophylaxe, Diagnose und Therapie während des Unterrichtens – eventuell durch Computer im Schüler-Computer-Dialog unterstützt – verbessern kann. Vielmehr handelt es sich dabei um ein grundsätzliches Problem, bei dem Fragen folgender Art Antworten verlangen:

- Ist der Handlungsgrund, ohne den der einzelne Schüler sein Lernpensum nicht zu bewältigen vermag, tatsächlich stark genug?
- Lässt sich in solchen Fällen, in denen einzelne Schüler durch zu große Misserfolge im Lernprozess demotiviert werden, eine gegenseitige Hilfe unter Schülern organisieren, bei der die Motivierung des zurückbleibenden Schüler dank der Unterstützung durch seinen Mitschüler wieder wächst, während der Handlungsgrund des Mitschülers einen neuen, gesellschaftlich besonders positiven Aspekt erhält?
- Sind Fälle, in denen Schüler durch Misserfolge im Lernprozess demotiviert werden, so häufig, dass sich Konsequenzen für die Lehrplangestaltung ergeben?

In der Volksbildung der DDR gab es derartige Fragestellungen, wenn auch ihre Lösung durch ideologische Engstirnigkeit nicht zustande kam.

Während sich auf Grundlage sehr umfangreichen empirischen Materials Diagnose-Therapie-Modelle in den Unterrichtsfächern Mathematik, Physik, Chemie, Muttersprache erfolgreich entwickeln ließen, war das bei dem empirischen Material aus dem Russischunterricht schwieriger. An dem Material war zu erkennen, dass es eine nicht geringe Anzahl von Schülern gab, die den Anschluss in diesem Fach bereits völlig verloren hatten, dem Unterricht faktisch nicht mehr folgen konnten.

Erfolgreiche therapeutische Maßnahmen wären nur als sehr umfangreicher nachholender Unterricht möglich gewesen. Die zuständigen Fachvertreter der Pädagogischen Akademie in Berlin trauten sich nicht zu, die Ministerin für Volksbildung mit dem Problem zu konfrontieren. Aus Erfahrung wussten sie, dass auf sie die Schuld wegen angeblich viel zu schwerer Aufgabenstellung für die Schüler abgewälzt worden wäre.

Wieder sind Analogien mit der Medizin als einer anderen Wissenschaft, in der die Komplexität des Untersuchungsgegenstandes ein vergleichbares Ineinandergreifen von Aktivitäten praktischer und theoretischer Vernunft verlangt, nicht zu übersehen. Auch in der medizinischen Forschung kann Häufung von Negativem oder zumindest Fragwürdigem die Lösung umfassenderer – so sozialer, sozialhygienischer, gesundheitspolitischer – Probleme erfordern.

3 Zur Bedeutung der Modellierung für den Mensch-Maschine-Dialog

Es ist sicher eine Binsenweisheit, dass der Computer textsprachliche, akustische und visuelle Information verarbeitet und mit dem Menschen auf informeller Ebene kommuniziert. Der Computer/die Rechentechnik (IT) ist eines der wesentlichsten Hilfsmittel des Neuzeitmenschen geworden. Informelle Tätigkeiten werden der Maschine anvertraut.

Wir behaupten, dass sich aber nur die Tätigkeiten lohnen, formal und damit IT-tauglich aufzubereiten, die sich wiederholen, invariant sind und damit gesetzmäßige Züge tragen. Die invarianten, informellen Prozesse der IT sind, falls sie nicht ausdrücklich Forschung oder Spiel betreffen, Hilfsmittel in menschlichen Entscheidungsprozessen. Dementsprechend ist, abgesehen von den genannten Ausnahmen, Modellierung informeller Tätigkeiten stets im Kontext von Entscheidungsprozessen zu betrachten.

Dieser Zusammenhang ist keine Binsenweisheit und ist aus Sicht der Autoren in Software nur in Ausnahmen bewusst umgesetzt.

Menschliche Entscheidungen können rational, nichtrational oder ausgesprochen irrational sein. Entscheidungsprozesse können sich auch aus Komponenten zusammensetzen, die sich hinsichtlich Rationalität, Nichtrationalität oder Irrationalität unterscheiden. Von Zufällen abgesehen, sinken die Chancen, dass durch Realisierung einer Entscheidung die in sie gesetzten Erwartungen erfüllt werden, umso mehr, je näher deren Komponenten der Irrationalität sind (KRE. 2006/2007). Demnach kann es sich beim Bemühen, Entscheidungen mit Hilfe von IT zu unterstützen, nur um das Bemühen handeln, rational zu entscheiden. In diesem Zusammenhang ist auch nicht zu übersehen: Die praktischen Zwecke, denen Forschungsergebnisse dienen sollen, sind auf rationales Entscheiden gerichtet.

Die Basis, damit der erste Schritt im Gesamtprozess (s. Grafik), ist die Modellierung informeller Tätigkeiten des Menschen. Dies setzt die Erkenntnis der Invarianten der Tätigkeiten und die Erklärung und damit Vorhersage dieser Tätigkeiten, entsprechend der mit zu modellierenden Rahmenbedingungen, voraus. Genau dieser Ansatz an der APW, sauber zwischen Beobachtbarem (Resultaten der Tätigkeiten) und Theoretischen (nicht beobachtbare dispositionelle Eigenschaften des Menschen) unterschieden zu haben, ist heute aktueller denn je.

IT-Pflichtenhefte beginnen in der Regel mit der Geschäftsprozessanalyse und den berühmten Frage an den Nutzer „Was tun Sie eigentlich wie und warum?“. Wenn dann der normale Nutzer dem IT-Fachmann verständnislos und hilflos gegenübersteht, könnte genau das Vorgehen beginnen, was in den 1970-iger Jahren mit der Analyse von Leistungskontollergebnissen des Mathematikunterrichtes in 5. und. 6. Klassen mit erheblichem Aufwand vorgenommen worden ist.

Stark vereinfacht wurden folgende Arbeitsschritte gegangen:

1. Herstellung von Beziehungen zwischen Klassen von Anforderungen (mathematische Aufgabenstellungen) und Klassen von Ergebnissen (korrekte Ergebnisse und invariante Fehler; Ergebnistypen)
2. Dispositionelle Erklärung der Ergebnistypen durch eine Menge möglicher geistiger Tätigkeiten (mathematische Lösungs- und systematisch-invariante Irrwege)
3. Herstellung von Versuchsbedingungen (geänderte mathematische Aufgabenstellungen) und Voraussage der Ergebnistypen
4. empirische Analyse der Ergebnistypen und Verifizierung der Voraussagen
5. methodische Empfehlungen an Lehrbücher und Lehrer zur didaktischen Umsetzung im Normalunterricht
6. Erste Schritte zur Umsetzung der tätigkeitsorientierten Methodik in computergestütztes Lernen

Übertragen auf heutige Gegenstände in Applikationen (ERP, BI, BPM, CRM, DMS, CAD, CAFM, und viele weitere Kürzel, die sicher allen Anwesenden geläufig sind), sollte man weniger das Modellieren, was der Nutzer sagt oder meint, sondern beobachten, was er tut. Damit ist der Aufwand zur Modellierung um ein mehrfaches höher, aber resistenter gegenüber Veränderungen der Anpassung von Prototypen.

Einschränkend sei gleich gesagt, dass der Aufwand des empirisch-theoretischen Vorgehens sich nicht an allen Stellen lohnt (deshalb wurde an der APW bewusst die 5. und 6. Klassenstufe gewählt, weil hier nach Auffassung der Lehrer wesentliches mathematisches Grundverständnis vermittelt wird).

IT-Programme, die Invariantes abbilden und invariante Prozesse unterstützen, repräsentieren Standards. Standards sind ein wesentlicher Weg zur effektiven IT-Nutzung und aus unserer Sicht neben intuitiver Erfahrung auch durch systematische Tätigkeitsanalysen erreichbar. Jegliche Begriffsbildung beruht bewusst oder unbewusst auf invarianten Merkmalen (invarianter Semantik), wenn auch manchmal – wie beispielsweise in der Biologie – diese invarianten Merkmale erst in langwierigen Forschungen und dazu noch als nicht unmittelbar beobachtbar ermittelt werden.

Leider sind die bis in die 1990iger Jahre betriebenen Forschungsmethoden der APW der DDR mit deren Untergang auch verloren gegangen und haben, ob des gewaltigen analytischen Aufwandes, auch wenig Chancen, wieder belebt zu werden.

Andererseits wird zu Recht beklagt, dass es zwar wesentlich verbesserte technische Hilfsmittel für Computerunterstütztes Lernen gibt, aber die theoretischen Grundlagen seit Jahren keine wirklichen Fortschritte haben [D21, 2007]. Tätigkeiten erfolgreich zu führen, wird aber nur funktionieren, wenn man deren Gesetzmäßigkeiten erkannt hat.

4 Empirisch-theoretisches Vorgehen

4.1 Theoretische Grundlagen Diagnose-Therapie-Modell

These 1: Logisch-mathematische Modellierung und Computernutzung sind ursprünglich nur wesentlicher Bestandteil eines Forschungsprozesses. Ziel dieses Forschungsprozesses ist das Gewinnen von Diagnosen und Therapien zu Schülerfehlern, die relativ häufig bei Auswertung sehr umfangreicher Daten – mehrere Tausend – in Gestalt schriftlicher Schülerergebnisse vorliegen. Dabei sind einige theoretische Probleme zu bewältigen:

- Ausschlaggebend für den Erfolg der Modellierung ist das Ermitteln zutreffender Diagnosen. Indem diese Diagnosen Aussagen über Fehlerdispositionen von Schülern darstellen, handelt es sich um Aussagen mit theoretischem Termini; denn für Dispositionen gilt, dass sie nicht unmittelbar beobachtbar sind.
- Wenn die interessierende Disposition bekannt ist, so sind Erklärungen oder Voraussagen über Symptome auf Basis deduktiver Schlüsse möglich. Allgemein gilt: Liegt eine bestimmte Disposition D1 bei einem Schüler x1s vor und wird diese Schüler vor eine Situation S1 gestellt – beispielsweise einige Aufgaben in Mathematik zu lösen –, so lässt sich Resultat R1, zu dem er gelangt ist, erklären, bzw. es lässt sich das Resultat R1, zu dem er gelangen wird, vorhersagen. Die Situation S1 und das Resultat R1 stellen dabei zusammengenommen das Symptom dar. Sortenlogisch lässt sich die Gesetzmäßigkeit, die das ermöglicht, mit dem Alloperator – $\forall x_1^s$ heißt: von allen x_1^s gilt – und der Implikation – $\rightarrow$ steht für ‚wenn .., so' – etwas vereinfacht, da auf Zeitterme verzichtet wird, so ausdrücken: $\forall x_1^s(D_1(x_1^s) \rightarrow (S_1(x_1^s) \rightarrow R_1(x_1^s)))$.

- Während dann, wenn die Disposition bekannt ist, mit Hilfe der hier vereinfacht dargestellten Gesetzesaussage Symptome erklärt oder vorhergesagt werden können, ist ein entsprechender Schluss von Aussagen über die Symptome auf die Aussage über die Disposition nicht möglich. Die bei Schülern auftretenden Symptome sind nicht mit manchen Indikatoren in der Technik gleichzusetzen, bei denen ein deduktiver Schluss vom Beobachtbaren auf das nicht unmittelbar Beobachtbare unter besonders günstigen Bedingungen statthaft erscheint. So spricht das Auftreten dieses oder jenes Symptoms lediglich mit mehr oder weniger großer Wahrscheinlichkeit, die durch empirische Untersuchungen genauer zu ermitteln ist, für die jeweilige Disposition.
- Da beim Vorliegen dieser oder jener Symptome gewöhnlich mehrere Aussagen über Dispositionen – also mehrere Diagnosen – als zutreffend in Betracht kommen, muss bei einer gewählten therapeutischen Maßnahme immer noch mit Misserfolg gerechnet werden. In solchen Fällen ist dann eventuell nach einer neuen therapeutischen Maßnahme zu fragen.

4.2 Praktische Ergebnisse in Grundschulen der DDR und an der TU Dresden

These 2: Die Diagnose und Therapie betreffenden Forschungsergebnisse können zuerst einmal zur Verbesserung des Unterrichts genutzt werden, in dem keine Computer eingesetzt sind. Dazu dienten in der DDR Unterrichtshilfen, d.h. schriftliche Anleitungen für den Lehrer zur Gestaltung seines Unterrichts. Ausgehend von den fundierten Diagnose-Therapie-Modellen war es möglich, in die Unterrichtshilfen Empfehlungen prophylaktischer Natur aufzunehmen. Durch diese prophylaktischen Empfehlungen ließ sich solchen Schülerfehlern vorbeugen, die in den vorangegangenen empirischen Untersuchungen relativ häufig festgestellt werden mussten. Die ersten positiven Auswirkungen waren im Mathematikunterricht zu registrieren. Auch nach 1990 wurde von einem für die Bildung verantwortlichen CDU-Minister der sächsischen Landesregierung (Herrn Rößler) das hohe Niveau des Mathematikunterrichts in der DDR anerkannt. Ein Faktor, der zu diesem Niveau beitrug, war die soeben beschriebene Prophylaxe. Es zeigte sich, dass prophylaktisches Vorgehen auf der Grundlage empirischen Materials gewonnener und erprobter Diagnose-Therapie-Modelle auch im naturwissenschaftlichen und im muttersprachlichen Unterricht möglich ist.

These 3: Bei den Diagnose-Therapie-Modellen gibt es mehrere Gründe dafür, dass an ein Ersetzen des Lehrers durch den Computer bei Stimulierung derjenigen Schülertätigkeit nicht zu denken ist, durch die sich der Schüler von seinem fehlerhaften Vorgehen befreit. Die beiden wichtigsten Gründe, die im Zusammenhang mit These 1 unter 4.1. zu sehen sind, seien hier genannt:

- Häufig kommen aufgrund erster empirischer Befunde über fehlerhaftes Verhalten eines Schülers mehrere sich gegenseitig ausschließenden Diagnosen über Dispositionen in Betracht. Das können zwei, drei, eventuell auch vier sein. Es zeigt sich: Auch im Falle hoher Wahrscheinlichkeit für das Zutreffen einer der in Betracht kommenden Diagnosen ist niemals völlig auszuschließen, dass sich eine bisher nicht beachtete Diagnose beim nächsten Schüler als wahr erweist. In dieser Hinsicht sind die Diagnose-Therapie-Modelle grundsätzlich

unvollkommen. Die Summe der Wahrscheinlichkeiten für das Zutreffen der positiven Diagnosen ist kleiner als 1.

- Diagnose- und Therapieprozesse sind ebenso wie durch prophylaktische Maßnahmen bewirkte Veränderungen durch innere und äußere Einflüsse störanfällig, die in den Diagnose-Therapie-Modellen unberücksichtigt bleiben müssen. Beispielsweise kann es wegen äußerer Ablenkung, wegen einer bisher unbeachtet gebliebene Erkrankung, wegen Ermüdungserscheinungen oder wegen ähnlicher Faktoren zu einem Verhalten des Schülers kommen, das sich eventuell stark von seinem normalen Verhalten unterscheidet.

Werden also in absehbarer Zeit Computer in Diagnose-Therapie-Prozessen genutzt, so grundsätzlich nur als Mittel zur Entlastung des Lehrers, nicht zur Ersetzung des Lehrers.

4.3 Einsatz rechentechnischer Mittel (Hard- und Software)

Für die Auswertung von Massendaten wurde zuerst ein R300 genutzt. Die Schüler-Ergebnisse wurden auf Lochkarten übertragen und ausgewertet. Einfache Auswertungen wurden mit RPG umgesetzt; komplexere Analysen wurden durch die Entwicklung eines Fachinterpreters (Siegried Zwiebel) umgesetzt. Der Interpreter wurde auf Basis der Maschinensprache MOPS erstellt.

Für erste Versuche im Computerunterstützten Unterricht (CUU) wurde ein LISP-Interpreter (Herbert Stoyan, Joachim Oelschlegel) entwickelt und genutzt. Die alphanumerischen Terminals für die Schülerarbeitsplätze gestatten nur sehr eingeschränkte Programminhalte und Programme. Die polnische Notation und das rekursive Verhalten von LISP waren für diese ersten Versuche sehr gut geeignet. Man konnte leicht didaktische Funktionen erstellen und dem Pädagogen zur Umsetzung seiner fachlichen Inhalte geben. Erster Nutzer des Systems war Dr. Helmut Menschel, der mit physikalischen Lehrabschnitten die physikalische Grundausbildung von Schülern der 6. Klasse versuchsweise gestaltete. Die zeichenkettenorientierte Arbeitsweise von LISP erlaubte damals schon recht komplexe Erkennung alphanumerischer Tastatur-Eingaben der Probanten.

Von einer breiten Nutzung konnte keine Rede sein, weil die 5 Bildschirmarbeitsplätze und die Grenzen des zu vermittelnden Stoffes doch erheblich waren. Übertragbar aus heutiger Sicht war aber der Ansatz, über die Modellierung des Verhaltens, die Erklärung der dispositionellen Persönlichkeitseigenschaften und den schrittweisen Zugang zu einer Fachsprache zur Modellierung des didaktisch-methodisch aufbereiteten Schulstoffs zu gelangen.

Aus heutiger Sicht sind die didaktisch-methodischen Inhalte wesentlich eleganter und grafisch aufbereiteter umzusetzen. Fehlt allerdings der theoretisch-empirische Hintergrund, dann fehlt der Aufbereitung auch die Basis. Dazu ist interdisziplinäre Forschung von Psychologen, Didaktikern, IT-Fachleuten und Mathematikern erforderlich, die auf einer breiten empirischen Basis aufzusetzen haben. Dieser Ansatz wird nach unserem Kenntnisstand bis heute nicht gegangen und scheint mit den Forschungen des FRZ der APW und deren Auflösung auch untergegangen zu sein.

5 Konsequenzen für IT-Einführung aus heutiger Sicht

IT-Nutzung ist in der Regel ein Mensch-Maschine-Dialog. Jeder dieser Dialogpartner hat Stärken und Schwächen. Darüber ist schon viel geschrieben und diskutiert worden. Was aus unserer Sicht fehlt, ist die komplexe Modellierung dieses Dialoges vor dem realen Dialog. Je nach Komplexität der abzubildenden realen Prozesse scheitern IT-Projekte in Deutschland zwischen 10-30%. Hauptursache ist aus unserer Sicht die Spanne von fehlendem Pflichtenheft bis zu überzogenem Pflichtenheft. Modellierung ist heute keine mathematische Disziplin, die anwendungsbereit in der Breite verfügbar ist. Die große Dynamik der realen Prozesse wird durch die Abbildung in Applikationen wie CAFM, GIS, CRM usw. nur teilweise befriedigend erfüllt. ERP-Prozesse, die weitgehend standardisiert ablaufen, scheinen am ehesten und erfolgreichsten effektiv umsetzbar zu sein. Allerdings verwundert es oftmals, mit welchen Einschränkungen der Nutzer zu kämpfen hat, so dass sich der Mensch-Maschine-Dialog mehr in Richtung Maschine als den zum Menschen bewegt.

Menschliches Verhalten hat immer einen Anteil dispositioneller Erklärungsversuche, da Dispositionen eben keine Ursachen sind. Wenn diese prinzipiellen Zusammenhänge auf einfache Determinismen reduziert werden, sind dann die vorher genannten Probleme gut zu erklären.

Eine Lösung könnte eine Modellierung mit entscheidungstheoretisch-empirischem Hintergrund sein. In diesen Kontext könnten auch Service Orientierte Architekturen (SOA) als die Invarianten von Tätigkeiten interpretiert werden. Der Tätigkeitsorientierte Ansatz ist aber wertlos, wenn er nicht, wie oben im Bild 1, in den Zusammenhang einer Modellierung des Gesamtprozesses gestellt wird. Dies ist eine Aufgabe, die noch viel Zeit erfordern wird.

Literaturverzeichnis

[KRE, 2006] Kreschnak, Horst; Rationales Entscheiden in Geschichte und Gegenwart; Europäischer Verlag der Wissenschaften Peter Lang; Frankfurt/Main 2007, Band 1

[KRE, 2007] Kreschnak, Horst; Rationales Entscheiden in Geschichte und Gegenwart; Europäischer Verlag der Wissenschaften Peter Lang; Frankfurt/Main 2007, Band 2

[D21, 2007] Unterlagen Jahreskongress Initiative D21; Symposium „Web 2.0 Neue Strategie für die Bildung“

Computergraphik an der Universität Rostock von 1969 bis 1990

Adolf Kotzauer

Dorfstraße 63c
17139 Basedow
adolf.kotzauer@web.de

Karl-Heinz Kutschke

Annette-Kolb-Ring 41
18106 Rostock
kh.kutschke@arcor.de

Abstract: Es wird das Forschungsgebiet „Computergraphik“ an der Universität Rostock von den Anfängen an der Sektion Mathematik bis zum tragenden Wissenschaftsbereich nach der Gründung der Sektion Informatik behandelt. Wichtige Forschungsthemen und Anwendungen der Computergraphik in Rostock werden in ihrem Zeitbezug und ihrer Relevanz aus heutiger Sicht vorgestellt und gewertet.

1 Einleitung

Der Startschuss für die Computergraphik in Rostock, damals unter der Bezeichnung Digitalgraphik, war ein Forschungsseminar, das im Sommersemester 1969 unter Leitung von Prof. Helmut Kiesewetter gemeinsam von Mathematikern der Universität Rostock und Kollegen des Instituts für Schiffbau (Forschungszentrum des DDR-Schiffbaus) durchgeführt wurde. In diesem Seminar wurden Themen behandelt, die den internationalen Stand der Computergraphik aufarbeiteten und die in Rostock vorhandenen rechentechnischen Möglichkeiten vorstellten.

Als Ausgangspunkt für die Untersuchungen auf dem Gebiet der Digitalgraphik lassen sich zwei Fakten nennen.
Auf der einen Seite wurde in der DDR im Jahre 1964 von der Regierung ein “Programm zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR” beschlossen. Weiter ergaben sich aus dem Staatsratsbeschluss vom April 1969 über die “Weiterführung der 3. Hochschulreform und die Entwicklung des Hochschulwesens bis 1975” Empfehlungen, Wissenschaft und Praxis sollten feste Kooperationsbeziehungen eingehen, die sich sowohl für die Wissenschaft als auch für die Praxis als vorteilhaft erweisen.
Auf der anderen Seite begann sich in diesem Zeitraum die Computergraphik weltweit als eigenständiges Gebiet zu entwicklen. Das erste interaktive computergraphische System, das System *Sketchpad* von Ivan Sutherland wurde 1963 fertiggestellt. 1965 wurde das erste Vektordisplay kommerziell vertrieben. Drei Jahre später wurde die Firma Evans Sutherland mit dem Ziel gegründet, spezialisierte Hardware für Zeichenroutinen zu entwickeln. Ebenfalls in den 60-er Jahren wurden vielfältige Algorithmen für die Computergraphik entwickelt. Dazu gehörten z. B. Algorithmen zur Modellierung von Kurven und Flächen (de Casteljau, Bezier, Coons) oder Sichtbarkeitsalgorithmen (Appel, Bouknight, Warnock), um nur einige zu nennen.

Beide Einflussfaktoren und der Bedarf der Schiffbauindustrie der DDR an graphischen Verfahren in Konstruktion und Produktionsvorbereitung führten dazu, dass an der Sektion Mathematik der Universität Rostock die Forschungsgruppe Digitalgraphik gegründet wurde. Parallel dazu wurde am Institut für Schiffbau eine Arbeitsgruppe mit einer analogen, jedoch stark auf die Belange des Schiffbaus orientierten Thematik gegründet. Beide Arbeitsgruppen arbeiteten auf der Grundlage fester Kooperationsvereinbarungen viele Jahre außerordentlich eng und ergebnisorientiert zusammen. In dieser Zusammenarbeit wurde auch die Erfahrung gesammelt, die eine praxisorientierte Forschungsarbeit in der Computergraphik über 30 Jahre ermöglichte, ohne dabei eine universitäre Grundlagenforschung zu vernachlässigen. Ende der 70-er und in den 80-er Jahren wurde das Kombinat Robotron zum Hauptkooperationspartner.

2 Die ersten 15 Jahre von 1969 bis 1984

Nach Abschluss des ersten Forschungsseminars zur Computergraphik im Sommer 1969 begann sofort die konkrete Forschungsarbeit. Die Existenz der beiden Arbeitsgruppen „Digitalgraphik“ im Schiffbau und an der Universität Rostock und die exakte Abstimmung der Arbeitsprogramme untereinander bei durchaus unterschiedlich ausgerichteten Arbeitszielen lieferten die Grundlage für eine außerordentlich fruchtbare wissenschaftliche Zusammenarbeit von Universität und Industrie. Die Arbeiten im Schiffbau waren ausgerichtet auf die Entwicklung einer sicheren, anwendungsreifen und schiffbauspezifischen Software. Die Entwicklungen an der Universität waren primär auf Grundlagen- und zielgerichtete Vorlaufforschung ausgerichtet. Der Schiffbau gestattete der Forschungsgruppe Digitalgraphik auch die Nutzung der im Schiffbau vorhandenen Technik (CD1604 und Zeichenautomat vom Typ KONGSBERG, später CD3300 als Hostrechner und ein System Kleinrechner RAYTHON gekoppelt mit einem interaktiven Vektordisplay vom Typ SINTA). Ohne die Nutzung dieser Technik war eine Computergraphik an der Universität Rostock zum damaligen Zeitpunkt nicht möglich.
Die wesentlichen Forschungsthemen dieser 15 Jahre (s. [KK85]) waren:

- 1969 bis 1974 Entwicklung der Graphischen Systeme DIGRA70 und DIGRA73
- 1975 bis 1979 Anwendungen des DIGRA73-Systems
- 1978 bis 1980 Entwicklung von Kernkonzepten
- 1980 bis 1981 Testversion GKS und Struktur von Dialogsystemen
- 1981 bis 1984 GKS1600 und Nutzer-Maschine-Interface.

2.1 Fachsprachenbasiertes interaktives graphisches System DIGRA73

Das erste Ziel der Arbeiten zur Computergraphik war die Entwicklung einer Menge von Unterprogrammen zur Unterstützung der graphischen Ein- und Ausgabe. So entstand an der Universität Rostock das DIGRA70-System. Es war eine Menge von FORTRAN-Unterprogrammen für die Ausgabe 2- und 3-dimensionaler graphischer Objekte auf der Grundlage eines vom Institut für Schiffbau entwickelten Unterprogramm-Pakets GIPS (Interaktives Graphisches Programm-System) für die Ein- und Ausgabe graphischer

Elemente. Zur Anwendung des DIGRA70-Systems war ein hoher Schreibaufwand für die Programmierung notwendig. Die Konsequenz daraus war die Entwicklung einer graphischen Programmiersprache [Ko72].
Das Nachfolgesystem des DIGRA70-Systems war das DIGRA73-System. Es hatte folgende Eigenschaften:

- Vorhandensein einer Fachsprache DIGRA als Erweiterung von FORTRAN zur Definition und Manipulation graphischer Objekte sowie zur Spezifizierung und Initialisierung der graphischen Ausgabe
- Vielfalt der Objekttypen, wie Linien- bzw. Drahtgitter, Freiformflächen (Coonspatches) und Körper
- Vielfalt der Ausgabeprozesse und wahlweise Berücksichtigung von Hidden Linie und Hidden Surface
- Graphischer Dialog mit einem umfangreichen, fest vorgegebenen Menübaum auf der Basis eines interaktiven graphischen Vektordisplays. Besonders hervorzuheben ist die interaktive Gestaltung der graphischen Darstellung von Objekten für Bildschirm und Zeichenmaschine.

Weitere Merkmale des DIGRA73-Systems waren eine strikte Trennung von Dialoggestaltung und Verarbeitung sowie eine dreistufige Datenbehandlung:

- Datenstrukturen für die Modellierung und Bildbeschreibung
- Datenstrukturen für die Ausgabe und interaktive Arbeit
- Datenstrukturen für die Bilddaten.

Die klare Struktur und starke Modularisierung des DIGRA73-Systems gestattete bereits ein Austauschen von Programmmoduln und lieferte eine sehr gute Grundlage für die spätere Entwicklung der Kernkonzepte.
Die Autoren sind davon überzeugt, dass das DIGRA73-System mit großer Wahrscheinlichkeit eines der ersten fachsprachenbasierten interaktiven graphischen Programmsysteme in Gesamtdeutschland war. Die eingeschränkte technische Basis grenzte allerdings die Einsatzmöglichkeiten erheblich ein. Die Arbeiten zum DIGRA73-System wurden 1975 mit dem Universitätspreis und 1978 mit einer hohen staatlichen Auszeichnung, dem Banner der Arbeit, anerkannt.

2.2 Anwendungen der Systeme DIGRA70 und DIGRA73

Eins der bekanntesten anwendungsorientierten Systeme, das auf dem DIGRA73-System aufbaute, war das System ROHR77 zur graphischen Darstellung, interaktiven Kontrolle und Korrektur von Rohrleitungsplänen. Dieses System wurde im Rahmen eines Staatsplanthemas und in enger Zusammenarbeit mit dem Kombinat „Rohrleitungen und Isolierung“ Leipzig entwickelt. Begleitet waren diese Arbeiten mit der Entwicklung und Implementierung von Algorithmen zur automatischen Wegfindung von Rohrleitungen in Räumen mit Hindernissen.

Eine andere Anwendung war der Entwurf von Maschinenaufstellungsplänen. Das System hatte die folgende Funktionalität: Definieren einer Werkhalle, Anordnen von Maschinen, Verschieben und Löschen von Maschinen, Erzeugen eines Werkhallenausschnitts, Verbinden von Maschinen und Erzeugen von Zeichnungen.
Weitere Anwendungen beschäftigten sich mit dem Anbringen von Bemaßungen, der Konstruktion und Darstellung von Baukörpern sowie der Planung von Moordränungen. Einzelne Systemteile des DIGRA73-Systems wurden auch in weiteren Anwendungen bei Kooperationspartnern genutzt.

Bei der breiteren Nutzung des DIGRA73-Systems taten sich aber sehr bald Probleme auf. Zur damaligen Zeit existierten in der DDR nur zwei Vektordisplay-Geräte, eins in Karl-Marx-Stadt und eins am Institut für Schiffbau in Rostock. Letzteres war die Basis für die Entwicklungsarbeiten im Wissenschaftsbereich Digitalgraphik, gleichzeitig aber auch das einzige Gerät, das den Nutzern für die Anwendung der entwickelten Programme zur Verfügung stand. Daraus ergaben sich massive zeitliche Engpässe, ganz abgesehen von den teilweise sehr langen Anfahrtswegen der Anwender.

Die Anwendungen des DIGRA73-Systems lieferten auch eine Reihe von Erfahrungen, Hinweisen und Problemen. Dazu gehörten:

- Handhabung großer Datenmengen, Strukturierung der graphischen Daten und Aufbau zweckmäßiger Datenstrukturen
- Modellierung graphischer Objekte und ihre Behandlung in Datenbanken
- Strukturierung großer Systeme, modularer Aufbau, Skalierung von Systemen in Abhängigkeit vom Bedarf der Anwendungen
- Gestaltung des Dialogs, effiziente Dialogführung, Gestaltung der Benutzungsoberfläche, woraus sich der Begriff des Dialogkerns entwickelte
- Werkzeuge zur rechnerunterstützten Systemgenerierung, insbesondere des Dialogteils.

2.3 Kernkonzepte

Basierend auf den Erfahrungen mit dem DIGRA73-System begannen unter dem Aspekt der Modularisierung, Übertragbarkeit und Generierung weiter Grundlagenuntersuchungen, die zunächst unter der Orientierung der Konzipierung eines neuen Systems DIGRA80 vorgenommen wurden. Von einer Realisierung musste auf Grund der rechentechnischen Bedingungen, des Aufwands und der Effektivität Abstand genommen werden. Die generelle Stoßrichtung waren Untersuchungen zur Realisierung einer modularen Systemarchitektur sowie des Erzielen einer hohen Übertragbarkeit und Anpassungsfähigkeit von einzelnen Systemteilen für den Aufbau von CAD-Systemen. Dabei konnten anwendungsunabhängige Systemteile, sogenannte Kerne, mit klaren Schnittstellen identifiziert werden, woraus dann die Kernkonzepte entstanden. Ein Kern war ein System von Prozeduren, der eine Teilaufgabe innerhalb eines graphischen Dialogsystems löst und über problemorientierte Schnittstellen zweckmäßig aktiviert werden kann. Folgende Kerne wurden entwickelt:

- **Dialogkern** zur Unterstützung der Kommunikation Nutzer-System
- **Strukturkern** zur Speicherung von graphischen Datenstrukturen sowie zur Realisierung des Zugriffs auf Elementen der Datenstrukturen und deren Manipulation
- **Ein- und Auslagerungskern** zur Realisierung eines virtuellen Speichers für Datenstrukturen
- **Datenbasiskern** zum dialogorientierten Aufbau und zur dialogorientierten Manipulation von Datenbeständen in Datenbasen
- **Ausgabekern** zur graphischen Darstellung und Ausgabe von 2- und 3dimensionalen Objekten unter Beachtung von Transformationen.

2.4 Entwicklung graphischer Kernsysteme und Standardisierung

Die Arbeiten an universell nutzbaren Kernen lenkten die Aufmerksamkeit der DIGRA-Forschungsgruppe in Rostock auf den 1976 mit dem SEILLAC 1 Workshop „Methodology in Computer Graphics“ beginnenden Prozess der Standardisierung in der Computergraphik. 1979 entschied die Arbeitsgruppe DIN-NI/UA-5.9 das GKS (Graphical Kernel System) zu einem internationalen Standard weiterzuentwickeln (vgl. [Ur90]).

Bereits 1982 wurde gemeinsam mit dem Kombinat Schiffbau eine erste Implementation des Standardentwurfs GKS (Version 6.4) unter Nutzung des GIPS (s. 2.1.) des Kombinats Schiffbau realisiert. Hierauf aufbauend wurde dann in Kooperation mit dem Kombinat Robotron und dem Leitzentrum für Anwenderforschung Berlin, BT Schwerin, das GKS1600 entwickelt. GKS1600 wurde auf der SKL-Technik (System der Kleinrechentechnik) entwickelt. Hervorzuheben sind die speziell von der Universität Rostock entwickelten Algorithmen zur Realisierung der Rasterfunktionen in der vollen Funktionalität, wie es der Standard vorschrieb. Mit dem Arbeitsplatz für Konstruktion und Technologie AKT A6454 aus dem Kombinat Robotron erfuhr GKS1600 innerhalb der DDR eine breite Anwendung.
Nach Abschluss der Arbeiten zum GKS1600 wurde eine Implementierung für die PC-Technik realisiert und unter der Bezeichnung GKSU vertrieben. Zur Gewährleistung einer Aufwärtskompatibilität wurde zum GKS1600 eine Verarbeitung des GKS1600-Metafiles geschaffen.
1984 wurde dann das System GKS1600 fertiggestellt mit nur wenigen Abstrichen gegenüber der GKS-Beschreibung des Standards und ein Jahr vor der Bestätigung des GKS als internationaler Standard angenommen wurde.
GKS wurde 1988 unter Mitwirkung der Forschungsgruppe der Universität Rostock Fachstandard in der DDR [TGL].

2.5 Nutzerinterfacegestaltung

Bereits mit der Realisierung und den Anwendungen des DIGRA73-Systems rückten Fragestellungen zur Gestaltung der graphischen Interaktion, d. h. der Gestaltung von User Interfaces oder Nutzerinterfaces, in das Blickfeld und in den Mittelpunkt einer nutzerorientierten Handhabung des Systems. Infolgedessen wurden Bausteine zur Gestal-

tung von Nutzerinterfaces konzipiert und implementiert. Charakteristisch waren einerseits die Bemühungen und der Zwang (sehr schwache rechentechnische Basis) zur konsequenten Modularisierung und andererseits die strikte Trennung zwischen Dialogbeschreibung und Dialogführung und die Orientierung auf den CAD–Bereich. Neben den Implementierungsarbeiten erfolgten vielfältige Untersuchungen einer theoretischen Fundierung der praktischen Arbeiten auch mit dem Ziel, den kaum zu überwindenden technischen Einschränkungen der verfügbaren technischen Basis zu begegnen. Die Suche nach machbaren Lösungen insbesondere bei der Gestaltung von Nutzerinterfaces führte zu vielfältigen methodischen Erfahrungen, die, wenn auch nur teilweise veröffentlicht, bei Kooperationspartnern und Mitarbeitern der Forschungsgruppe bei vielen anderen Projekten zur sehr erfolgreichen Arbeiten führten. Dazu gehörten auch Fragen der Generierung von Dialogsystemen. Sie waren auch Gegenstand von Untersuchungen des Wissenschaftsbereichs „Programmierungstechnik“ [Lä84].

Der Zwang zur modularen Struktur bei der Gestaltung der Mittel und Methoden für den Aufbau von Nutzerinterfaces war naturgemäß mit der Einführung bestimmter Begriffe verbunden und eine konsequente Anwendung des Kernkonzepts auf die Gestaltung von graphischen Dialogsystemen. Zwei zentrale Begriffe waren:

- Dialogkern für die Dialogführung, Laufzeitmodul für die Realisierung der Interaktion
- Dialogkern für die Dialogbeschreibung (Tool für die Beschreibung der Interaktion).

Sie sind ein Ausdruck der Trennung zwischen Dialogbeschreibung und Dialogführung.

In die Untersuchungen wurden sehr frühzeitig ergonomische Anforderungen und Richtlinien zur Gestaltung von Benutzungsoberflächen eingeschlossen. Ein wesentlicher Punkt der Untersuchungen war stets die Verbindungen mit den Graphischen Standards. Die Implementation der Konzepte erfolgte auf dem AKT A6454.

3 Die Entwicklung von 1984 bis 1990

Die Entwicklung der ersten 15 Jahre Computergraphik wurde an der Universität Rostock in den folgenden 7 Jahren von 1984 bis 1990 im neuen organisatorischen Rahmen der 1984 gegründeten Sektion Informatik fortgesetzt (s. [KS99]). Sie erfolgte unter den Bedingungen der erhöhten Forderungen nach leistungsfähigen interaktiven graphischen Systemen in der Praxis und einer völlig unzureichenden rechentechnischen Basis sowie schlechten Arbeitsbedingungen. Der Hauptauftraggeber wurde das Kombinat Robotron Dresden. Die Hauptzielrichtung war die Schaffung von Basissoftware und dazu notwendige Grundlagenuntersuchungen.

Der Auftrag des Ministeriums für Hoch- und Fachschulwesen zum Aufbau eines „Computergraphikzentrums-CDA/CAM des Hochschulwesen der DDR“ (CGZ) an der Universität Rostock sollte bessere Bedingungen für die Entwicklung der Computergraphik und eine praxisrelevante Anwendungsforschung schaffen. Die ersten konkreten Schritte

zum Aufbau des CGZ ergaben zweifelsfrei verbesserte Bedingungen für die Computergraphik und darüber hinaus für die gesamte Sektion Informatik. Dazu gehörten:

- die Aufstockung des wissenschaftlichen Personals. Zeitweise arbeiteten mehr als 40 Computergraphiker an der Sektion Informatik.
- eine Verbesserung der materiellen technischen Basis. Es erfolgte eine Ausrüstung mit einer für DDR-Verhältnisse sehr guten Hard- und Software. Das waren eine VAX 780 und eine VAX Microstation II (VT220, VMS, ULTRIX), ein interaktiver graphischer Bildschirm von TEKTRONIX und PC-Technik sowie CAD-Systeme EUCLID, MEDUSA, CADdy.
- ein Neubau, das heutige Gebäude des Instituts für Informatik. Er wurde als sogenannter Initiativbau, d. h. außerhalb der Planvorhaben, errichtet, wobei die Mitarbeiter der Sektion Informatik viele freiwillige Arbeitsstunden leisteten.

Der Aufbau eines CGZ wurde 1990 nicht weiter fortgeführt, und alle Arbeiten dazu wurden eingestellt.

In den Jahren 1985 bis 1990 sind folgende Themen als Schwerpunkte der Computergraphikforschung zu nennen:

- Graphik-Nutzerinterface
- Graphische Kernsysteme
- CAD/CAM und Sondervorhaben Softwareentwicklung.

3.1 Graphik-Nutzerinterface

Das Thema Graphik-Nutzerinterface ist eine Kurzbezeichnung für User-Interface-Management-Systems (UIMS) oder für die Gestaltung von Nutzerinterfaceverwaltungssystemen (NIVS) unter besonderer Orientierung auf interaktive graphische Systeme. Es bedeutet die konsequente Fortsetzung der Arbeiten zu den Kernkonzepten bei einer klaren Schwerpunktsetzung auf den Aufbau von Datenbanken für die Verwaltung der Interaktionsbeschreibung zur Realisierung der Interaktion. Ein weiterer Aspekt war die Bereitstellung von Werkzeugen für die Gestaltung und Generierung von Nutzerinterfaces bzw. von Dialogsystemen. So wurden in [Va85] für die Phasen Spezifikation, Implementation, Testung, Dokumentation, Änderung und Simulation im Lebenszyklus von Dialogsystemen Hilfsmittel entwickelt und Autorensysteme für Dialogsysteme als Prototyp realisiert. Weitere Arbeiten lieferten Ergebnisse zu einer effektiven Verwaltung von Nutzerinterfacebeschreibungen in Nutzerinterfaceverwaltungssystemen sowie zur effektiven Steuerung des Datenaustausches über interne Schnittstellen in graphischen Dialogsystemen.

Die Untersuchungen von ergonomischen Anforderungen und Richtlinien zur Gestaltung von Benutzungsoberflächen wurden fortgesetzt. Hier kam es auch zu einer Zusammenarbeit mit der Humboldt-Universität Berlin.

Softwaretechnologisch waren auch aus heutiger Sicht sehr gute Konzepte entstanden. Ein Beleg für diese Aussage waren die umfangreichen Kooperationen mit der Industrie und die vielfältigen wissenschaftlichen Veranstaltungen zum Gegenstand, deren Ergebnisse weitestgehend in den Rostocker Informatik-Berichten veröffentlicht wurden. Darüber hinaus wurden Untersuchungen einschließlich experimenteller Implementationen zum Einsatz von Mitteln und Methoden der

- Objektorientierten Programmierung (Grundlage Smalltalk) sowie der
- Künstlichen Intelligenz (wissensbasierte Nutzerinterfaces)

durchgeführt.
Die erzielten Ergebnisse fanden 1989/90 ihre Wertschätzung in der intensiven Einbeziehung der Rostocker Forschungsgruppe in vom BMFT geförderte Projekte, wie

- Verbundprojekt STONE (A STructured and OpeN Software Environment)
- Forschungsprojekt Graphik-Nutzer-Interface
- Verbundprojekt EXPOSE (Expertensystem zur phasenorientierten Software-Ergonomie-Beratung) bei der Benutzerschnittstellen-Entwicklung.

3.2 Graphische Kernsysteme

Unter dem Thema Graphische Kernsysteme wurden die Arbeiten zur Standardisierung aus den Vorjahren kontinuierlich fortgesetzt [Ur90]. In den Jahren 1985 bis 1990 erfolgten verstärkt Untersuchungen zu GKS-3D und PHIGS (Programmers Hierarchical Interactive Graphic System).
GKS-3D wurde ebenfalls für das Kombinat Robotron als vertriebsfertiges System umgesetzt. Es war ein Subset des damaligen Standardentwurfs GKS-3D. Bereits ein Jahr nach Vorliegen des internationalen Standards GKS-3D wurde eine Implementierung unter der Vertriebsbezeichnung GKS1800 vorgelegt.
Alle Implementierungsarbeiten von GKS1600, GKS1600-Metafile und GKS1800 erfolgten arbeitsteilig mit dem Kombinat Robotron.
Außerdem wurde das Themenspektrum etwa ab Mitte der 80-er Jahre erheblich erweitert. Themen im Umfeld der Rastergraphik, der Computergraphikhardware, der effizienten Speicherung graphischer Daten, der realitätsnahen Bilddarstellung und vor allem der graphischen Benutzungsoberflächen gewannen zunehmend an Bedeutung.
Eine wesentliche Fragestellung bei der objektorientierten Programmierung war die Möglichkeit der Einbindung graphischer Standards in eine objektorientierte Umgebung. Die Ergebnisse dieser Untersuchungen führten u. a. auch zur Kritik an dem PHIGS-Standard aus objektorientierter Sicht.

Darüber hinaus wurden noch die folgenden Teilthemen bearbeitet:

- HLHSR-Konzepte (Hidden Line Hidden Surface Removal) in der Umgebung Graphischer Standards
- Geometrische Modellierung und Graphische Standards

- Integration von Fenstersystemen und Graphische Standards
- Ray Tracing in der Umgebung Graphischer Standards
- Konzepte zur Realisierung des Zentralen Strukturspeichers von PHIGS
- Freiformkurven und –flächen in Graphischen Standards
- CGM (Computer Graphics Metafile) und GKS.

Die Arbeiten zu Graphischen Kernsystemen und User Interface Management Systemen wurden 1987 mit dem Banner der Arbeit ausgezeichnet. Dabei wurde in erster Linie die fachliche Leistung anerkannt, aber auch das Einhalten von sehr engen Terminvorgaben aus der Industrie.

3.3 CAD/CAM und Sondervorhaben Softwareentwicklung

In der zweiten Hälfte der 80-er Jahre wurden vielfältige Arbeiten für konkrete CAD-Systeme und zahlreiche wissenschaftliche Beratung für Kooperationspartner durchgeführt. Dabei wurden die Arbeiten zu Datenbanken für CAD in den Wissenschaftsbereich „Datenbanken“ der Sektion Informatik ausgelagert. Im Rahmen von Forschungs- und Entwicklungsaufgaben des Wissenschaftsbereichs „Digitalgraphik“ der Sektion Informatik der Universität Rostock waren Mitarbeiter u. a. an folgenden Themen von Kooperationspartnern beteiligt:

- Arbeiten zur Softwarearchitektur des CAD/CAM-Systems „Schiff“
- Datenaustausch zwischen CAD-Systemen, Implementation notwendiger Prä- und Postprozessoren
- Datenbanken für CAD-System „Schiffsmaschinenanlagen“
- Gestaltung von Benutungsoberflächen für die Fertigungssteuerung von Brennschneidautomaten
- Realisierung eines Nutzerinterface für ein System zur rechnerunterstützten Rohrleitungsfertigungssteuerung.

Das Thema Sondervorhaben Softwareentwicklung fasste eine Reihe von Entwicklungsarbeiten zusammen, die unmittelbar mit der Softwareproduktion des Kombinats Robotron zusammenhingen. D. h., diese Arbeiten der Universität Rostock unterlagen den betriebsinternen Geheimhaltungsbestimmungen des Hauses Robotron. Der Geheimhaltung unterlagen alle technologischen Bedingungen der Softwareproduktion, die Softwareproduktion für künftige Systemprodukte des Kombinats Robotron, die entwickelten Algorithmen und alle Vorlaufuntersuchungen für künftige Produkte. Dazu gehörten vertriebsfähige Bestandteile des Graphischen Kernsystems GKS1800 insbesondere eine Reihe von Lösungen, wie

- Graphiktreiber zur Unterstützung mehrerer Workstationtypen für 2D- und 3D-Graphik mit 3D-Schraffurgenerierung, 3D-Mustergenerierung und 3D-Darstellungsverfahren
- Realisierung von Verdeckungsalgorithmen
- Realisierung einer Reihe spezieller Funktionen der Graphischen Standards

- Algorithmen für das Rückrechnen von Projektionsparametern

und ihre Implementierung. Hinzu kamen weiter Untersuchungen der Realisierbarkeit der Netzfähigkeit graphischer Kernsysteme und der Anforderungen an die Rechentechnik unter Berücksichtigung des Einsatzes von Graphischen Standards, von X-Window und objektorientierten Techniken. Ferner wurden auf dem Weltmarkt vorhandene CAD-Systeme analysiert, um eine Analogentwicklung in der DDR vorzubereiten bzw. zu realisieren.

4 Forschungskooperationen

Entsprechend dem Charakter der Forschungsarbeiten zur Computergraphik an der Universität Rostock existierten vielschichtige Kooperationsbeziehungen zu anderen Hochschulen, zu Forschungseinrichtungen und zu Industrieunternehmen der DDR sowie zu Hochschulen und Forschungseinrichtungen des Auslands, die in den mehr als 20 Jahren der Entwicklung in Form und Art der Zusammenarbeit variierten. Mit ausländischen Industrieunternehmen gab es keine Beziehungen. Zu den Hochschulen und Forschungseinrichtungen gehörten:

- Technische Universität Dresden
- Technische Hochschule Karl-Marx-Stadt
- Technische Hochschule „Otto von Guericke“ Magdeburg
- Technische Hochschule Leipzig
- Humboldt-Universität Berlin
- Bauakademie der DDR
- Akademie der Wissenschaften der DDR, Zentralinstitut für Kybernetik und Informationsprozesse sowie Zentrum für Rechentechnik.

Die Kooperation mit der TU Dresden, Sektion Mathematik, Arbeitsgruppe Digitalgeometrie, war zweifelsfrei am intensivsten und dauerhaftesten über die gesamte Zeitspanne von 1970 bis 1990.
Von besonderer Bedeutung für die Praxisrelevanz der Arbeiten zur Computergraphik an der Universität Rostock war die Kooperation mit der Industrie. Bereits bei der Initialisierung der Computergraphikforschung an der Universität Rostock gingen wesentliche Impulse vom Institut für Schiffbau Rostock aus, die zu konkreten Kooperationsverträgen führten. Kooperationsverträge bestanden für konkrete Aufgabenstellungen mit folgenden Unternehmen:

- Schiffbauindustrie (Institut für Schiffbau, MTW Wismar, Warnow-Werft Warnemünde)
- VEB Kombinat Robotron (ZFT, RED)
- VEB Carl Zeiß Jena, Forschungszentrum Automatisierung der technischen Vorbereitung (FZ AUTEVO)
- VEB Kombinat Rohrleitungen und Isolierungen Leipzig

- VEB Leitzentrum für Anwendungsforschung Berlin, Betriebsteil Schwerin (LFA)
- VEB Kombinat Umformtechnik Erfurt
- Forschungszentrum Werkzeugtechnik (FZW) Karl-Marx-Stadt.

Forschungsarbeiten zur Computergraphik sind ohne internationale Kontakte undenkbar. Leider gab es da erhebliche Einschränkungen und eine generelle Ausrichtung nur auf den Osten. Trotzdem gelang es ab 1985 mit Darmstadt, Wien und Zürich direkte Beziehungen aufzubauen. Die ausländischen Partner waren:

- Ungarische Akademie der Wissenschaften, SZTAKI Budapest
- Technische Hochschule Prag
- AdW der UdSSR Moskau
- Institut für Steuerungsprobleme IPU Moskau
- Kiever Polytechnisches Institut KPI Kiev
- Kazaner Universität
- TH und Zentrum für graphische Datenverarbeitung Darmstadt
- TU Wien
- Universität Zürich.

Mit den Kollegen der Ungarischen Akademie im SZTAKI Budapest verband uns eine besondere Beziehung. Durch sie bekamen wir Einblick in die Fachliteratur westlicher Länder, eine nicht hoch genug zu schätzende Hilfe. An der Universität Zürich war es möglich, einen weitgefächerten Einblick in Arbeiten zu User-Interface-Management-Systems (UIMS) zu erhalten. Insbesondere Studienaufenthalte von Mitarbeitern der Rostocker Gruppe am Zentrum der Computergraphik in Darmstadt (Leitung: Prof. Dr.-Ing. Dr. h.c. mult., Dr. e.h., Hon. Prof. mult. José L. Encarnação) waren von außerordentlichem Wert für die Arbeiten in Rostock. Zwei Sachverhalte wurden bei Studienaufenthalten von Rostockern in Darmstadt deutlich:

- Die Rostocker Arbeiten hielten einem fachlichen Vergleich Stand. Das betraf insbesondere die Qualifikationsarbeiten, den Ideenreichtum und die Leistungsfähigkeit der Rostocker Gruppe.
- Das Handicap der schlechten technischen Basis in Rostock war auf längere Sicht nicht auszugleichen. Bestimmte Arbeiten konnten einfach nicht durchgeführt werden.

5 Schlussbetrachtungen

Ein historischer Rückblick ist immer subjektiv. Die Autoren haben versucht, objektiv zu sein. Von Vorteil war dabei die Tatsache, dass eine Reihe von Artikeln (s. Literaturverzeichnis) zur Entwicklung der Informatik und speziell der Computergraphik vorlagen, die kritische Leser hatten. Die Autoren dieses Artikels waren an der Entwicklung der Computergraphik in Rostock in allen Jahren aktiv beteiligt. Sie sehen sich jedoch nur stellvertretend für eine Reihe von ehemaligen Kollegen, wie Prof. Dr. rer. nat. habil

Helmut Kiesewetter und Prof. Dr.-Ing. habil Hansgeorg Meißner und jüngere ehemalige Mitarbeiter, die als Studenten in der Computergraphik in Rostock begannen und heute weiterhin sehr erfolgreich Computergraphik in Rostock betreiben. Dazu gehören Frau Prof. Dr.-Ing. habil. Heidrun Schumann, Universität Rostock; Herr Prof. Dr. Bodo Urban, Universität Rostock und Leiter des Institutsteils Rostock des Fraunhofer-Instituts für Graphische Datenverarbeitung, und Herr Dr. Erhard Berndt, Abteilungsleiter im Institutsteil Rostock des Fraunhofer-Instituts für Graphische Datenverarbeitung.

Prof. Kiesewetter hatte 1969 den Mut, innerhalb der Mathematik gemeinsam mit der Industrie Computergraphik zu beginnen. Prof. Meißner, heute Geschäftsführer der Firma ANOVA Multimedia Studios Rostock, trug seit 1986 Mitverantwortung für die Entwicklung der Computergraphik in Rostock. Allen ehemaligen Kollegen sagen wir Dank für eine jahrelange, kreative, offene und leistungsorientierte Zusammenarbeit. Das Arbeitsklima in der Rostocker Computergraphik war stets sehr gut.

Mit dem Jahr 1990 begannen tiefgreifende gesellschaftliche Veränderungen, die sich auch auf die Computergraphik in Rostock auswirkten. Natürlich ließen sich nun die Pläne „Computergraphikzentrum der DDR“ nicht mehr realisieren. Arbeiten für das Kombinat Robotron wurden beendet. Durch die Umstrukturierungen an der Universität gab es zudem einen krassen Personaleinschnitt.
Trotzdem gelang es, die Computergraphik auszubauen. Das lag an erster Stelle an der Gründung des Zentrums für Graphische Datenverarbeitung (ZGDV) 1990 und des Fraunhofer Instituts für Graphische Datenverarbeitung (FhG-IGD) 1992, beides Außenstellen der gleichnamigen Einrichtungen in Darmstadt unter Leitung von Prof. Encarnação. Die vielfältigen Kontakte der 80-er Jahre wurden nun ausgebaut und damit vielen Computergraphikern eine neue Perspektive geboten. Zudem wurde an der Universität Rostock 1991 das Institut für Computergraphik gegründet, die erste Institutsgründung an der Rostocker Informatik. Zurzeit arbeiten in den drei Einrichtungen über 50 Computergraphiker, und Rostock hat nach wie vor einen guten Namen in der Computergraphik.

Literaturverzeichnis

[Ko72] Kotzauer, A.: Eine graphische Programmiersprache zur Bearbeitung geometrischer Objekte. Dissertation, Universität Rostock, 1972.

[KK85] Kotzauer, A; Kutschke, K.-H.: 15 Jahre Digitalgraphik an der Rostocker Universität. Rostocker Informatik-Berichte. Wilhelm-Pieck-Universität Rostock. Heft 1, 1985, S.33-53.

[Ku90] Kutschke, K.-H.: Die Sektion Informatik der Wilhelm-Pieck-Universität Rostock 1989. Rostocker Informatik-Berichte/Universität Rostock, Fachbereich Informatik, Heft 10, 1990, S. 24-40.

[KS99] Kutschke, K.-H.; Schumann, H.: Computergraphik in Rostock – ein historischer Rückblick. Rostocker Informatik-Berichte / Universität Rostock, Fachbereich Informatik, Heft 23. 1999. S. 7 -16.

[Ku06] Kutschke, K.-H.: Entwicklung der Informatik an der Universität Rostock von 1964 bis 1990. In: GI-Edition, Lecture Notes in Informatics. Naumann, Friedrich; Schade, Gabriele (Hrsg.): Informatik in der DDR – eine Bilanz. GI Bonn 2006. S. 331 – 340.

[KK07] Kutschke, Karl-Heinz; Kotzauer, Adolf: Informatik und Computergraphik an der Universität Rostock von 1964 bis 1990. Rostocker Informatik-Berichte / Universität Rostock, Institut für Informatik, Heft 31. 2007. S. 59 - 70.

[Lä84] Lämmel, U.: Spezifikation und Implementation von Kommandosprachen für Dialogsysteme mit Hilfe attributierter Grammatiken. Dissertation. Universität Rostock. 1984.

[TGL] TGL 44610/01 Ausgabe 11. 86 Graphisches Kernsystem (GKS). Fachstandards – Software – Funktionale Beschreibung 2 D – Graphik. Amt für Standardisierung, Messwesen und Warenprüfung Berlin (Hrsg.) Verlag für Standardisierung. Berlin 1988.

[TM72] Thielke, H.; Marin, H.: Digitalgraphische Algorithmen zur Darstellung von räumlichen Gebilden unter Berücksichtigung der Sichtbarkeitsverhältnisse. Dissertation, Universität Rostock, 1972.

[Ur 90] Urban, B.:Arbeiten zur Graphischen Standardisierung an der WPU Rostock. Informatik-Berichte/Universität Rostock, Fachbereich Informatik Heft 10, 1990, S.53-58.

[Va85] Vatterrott, H.-R.: Unterstützung der Generierung von Dialogsystemen. Dissertation. Universität Rostock. 1985.

Von ALGOL 60 zur symbolischen Simulation

Günter Riedewald

Fakultät für Informatik und Elektrotechnik, Institut für Informatik
Universität Rostock
Albert-Einstein-Str. 21
18051 Rostock
gri@informatik.uni-rostock.de

Abstract: Die Arbeitsgruppe „Programmiersprachen und Übersetzertechnik“ an der Universität Rostock existierte von 1966 bis Ende September 2007. Der Artikel zitiert die wichtigsten Forschungsvorhaben und Forschungsergebnisse aus dieser Zeit und geht auf das jeweilige Umfeld stärker ein.

1 Einleitung

Die Forschungsgruppe „Programmiersprachen“ wurde 1966 unter der Leitung von I. O. Kerner im Rahmen der Sektion Mathematik der Universität Rostock gegründet. Später wurde sie in Arbeitsgruppe „Programmiersprachen und Übersetzertechnik“ umbenannt und bestand unter diesem Namen bis zum Ende der Existenz des gleichnamigen Lehrstuhls am 31.9.2007. Zu den Pflichten der Arbeitsgruppe gehörten sowohl die Forschung als auch eine damit eng verknüpfte Lehre.

In der Forschungsarbeit waren die Hauptziele

- Beiträge zur Grundlagenforschung auf dem Gebiet der Programmiersprachen und der Sprachprozessoren zu liefern
- sowie durch studienbegleitende Forschung neueste Forschungsergebnisse anderen, insbesondere Studenten, so schnell wie möglich zugängig zu machen.

Die Realisierung des zweiten Ziels führte z.B. zur Einführung neuer Lehrveranstaltungen. So gehörte die Ausbildung von Mathematikern und später von Informatikern in Rostock zu den Studiengängen in Deutschland, die schon zeitig Vorlesungen wie Übersetzertechnik, Semantik von Programmiersprachen, Parallele Prozesse und Logische Programmierung einführten. Ebenso wurden Einführungslehrveranstaltungen zur Programmierungstechnik durch Gebiete wie Strukturierte Programmierung oder Abstrakte Datentypen frühzeitig modernisiert.

Da der Forschungsschwerpunkt in Rostock Computergraphik, früher Digitalgraphik genannt, war, ergaben sich für andere Arbeitsgruppen weniger Ressourcen. Das wurde unter anderem 1977 durch ein Schreiben von I. O. Kerner an die SED-Parteileitung der Universität zum Ausdruck gebracht. Leider hatte es kaum Wirkungen. Nicht desto trotz gingen aus der Arbeitsgruppe fünf Professoren hervor und hervorragend ausgebildete Assistenten und Studenten, die in der Regel später gute Arbeitsplätze fanden, leisteten wichtige Forschungsbeiträge.

Eine relativ gute Unterstützung gab es durch die Universität bei der Reisetätigkeit. Reisen ins westliche Ausland waren beschränkt auf die Reisen I. O. Kerners im Rahmen seiner Mitgliedschaft in der Working Group 2.1 (WG 2.1) der IFIP (**eine** Teilnahme bei zwei jährlichen Sitzungen) und auf wenige Reisen des Autors kurz vor der Wende zur Universität Linköping in Schweden und zum INRIA in Frankreich. Reisen ins östliche Ausland bzw. zu internationalen Konferenzen wie MFCS (Mathematical Foundations of Computer Science) und FCT (Foundations of Computational Theory), sofern sie im östlichen Ausland oder in der DDR stattfanden, konnten dagegen in größerer Anzahl realisiert werden. Das führte dazu, dass sehr gute Kontakte zur Universität Rīga, zur Tschechischen Technischen Hochschule (ČVUT) Prag und zu einer Informatik-arbeitsgruppe der Polnischen Akademie der Wissenschaften in Warschau, der z.B. die international renommierten Informatiker Blikle, Mazurkiewicz, Dembiński und Małuszyński angehörten, aufgebaut werden konnten. Diese Kontakte – abgesehen von der Mitgliedschaft I. O. Kerners in der WG 2.1 – stießen uns ein Fenster in westlicher Richtung auf, besonders, was den Zugang zu westlicher Literatur betraf, und ermöglichten damit überhaupt erst bestimmte Forschungen. Diese internationalen Kontakte waren aber nicht einseitig ausgerichtet. Z.B. gab es umfangreiche gemeinsame Diskussionen zum Thema attributierte Grammatiken. An der Universität Rīga wurden vom Autor Vorlesungen für höhere Semester über attributierte Grammatiken und die Semantik von Programmiersprachen gehalten. Unsere Erfahrungen gingen ebenfalls in die Mitarbeit in verschiedenen Arbeitsgruppen ein, wie z.B. WG 2.1 der IFIP oder RG 23 „Spezialsprachen und Methodik der Programmierung“ der mehrseitigen Kommission „Wissenschaftliche Probleme der Informatik“ der Akademien der sozialistischen Länder.

Mehrfach wurde uns von westlichen Kollegen Zusammenarbeit zu gemeinsam beforschten Gebieten angeboten, was leider nicht realisiert werden durfte. Z.B. arbeitete C. H. A. Koster aus Amsterdam an den Affixgrammatiken, wohingegen wir die Grammatiken syntaktischer Funktionen (GSF) aus der Taufe gehoben hatten. Kosters Angebot zur Zusammenarbeit musste abgelehnt werden, obwohl er sich zum damaligen Zeitpunkt mit Berlin, aber leider im falschen Teil der Stadt (Berlin West), recht nahe zu Rostock aufhielt Genauso durfte ein Angebot zur Zusammenarbeit auf dem Gebiet der syntaktischen Analyse und der Syntaxfehlererkennung nicht wahrgenommen werden. Nach der Wende wurden zwar noch Kontakte zur Universität Maribor und zum CWI Amsterdam aufgenommen, aber insgesamt ging die Reisetätigkeit wegen Mangels an Haushaltmitteln zurück.

Die Leitung der Arbeitsgruppe hatte, bis zu seiner Berufung am 1.9.1977 zum ordentlichen Professor an die Pädagogische Hochschule K.-F. Wander in Dresden, I. O. Kerner inne. Nach einer anschließenden kurzen Leitungsphase durch S. von Weber übernahm der Autor die Leitung. Die Ausrichtung der Forschung auf Programmiersprachen und deren Implementation blieb allerdings während der gesamten Zeit erhalten. Will man eine Klassifizierung der Forschungsgebiete der Arbeitsgruppe vornehmen, so könnte man folgende Hauptüberschriften wählen:

- Die Programmiersprache ALGOL60
- Die Programmiersprache ALGOL68
- Attributierte Grammatiken und Anwendungen
- Symbolische Simulation
- Programmtransformation.

Die Bearbeitung dieser Gebiete erfolgte ungefähr in dieser Reihenfolge, aber selbstverständlich mit Überlappungen. Da in der DDR die Forschung koordiniert wurde, waren die Rostocker Arbeiten in der Hauptforschungsrichtung „Mathematische Grundlagen der Informationsverarbeitung“ eingegliedert.

In den nachfolgenden Kapiteln wird zu den wichtigsten Gebieten ein Überblick gegeben. Eine Liste ausgewählter Veröffentlichungen kann vom Autor angefordert werden.

2 ALGOL 60, ALGOL 68

Der Beginn dieses Zeitraums war durch ALGOL 60 geprägt. Im Mittelpunkt standen das Studium und die Propagierung von ALGOL 60, seiner neuen Sprachkonstrukte und seiner neuen Methode der Syntaxdefinition (BNF), sowie seiner Compiler. Es entstanden Lehrbücher zur Einführung von ALGOL 60 (Kerner/Zielke 1969) und zur Einführung in die Informatik (Kerner 1970, 1973), welches vom Aufbau und den Prinzipien von Rechenanlagen, über die theoretische Basis von Programmiersprachen, Ideen der Implementation von Programmiersprachen bis hin zur Programmierung numerischer Verfahren in ALGOL 60 reichte. Erste praktische Erfahrungen mit der Implementation von Programmiersprachen konnten Mitarbeiter und Studenten an der Rechenanlage ZRA 1 sammeln.

Die Mitgliedschaft von I. O. Kerner in der Arbeitsgruppe WG 2.1 der IFIP ermöglichte unserer Arbeitsgruppe den schnellen Zugang zu neuestem Material über Definitionsmethoden von Programmiersprachen, insbesondere zur sich in Entwicklung befindlichen Programmiersprache ALGOL 68, die für viele heutige Programmiersprachen (prominentes Beispiel ist C) eine Vorreiterrolle spielte, sowie eine frühzeitige und intensive Beschäftigung mit dieser Sprache. In Seminaren - zunächst für Mitarbeiter und später ebenfalls für Studenten - wurde ein systematisches Studium der Sprache und ihrer neuen Konzepte betrieben. Eine enge Zusammenarbeit gab es dabei mit einer Arbeitsgruppe an der TU Dresden unter der Leitung von Prof. Stiller. Beide Gruppen repräsentierten in der DDR das geballte Wissen über ALGOL 68 und seine Implementation. Um dem deutschen Leser ALGOL 68 zugänglich zu machen, wurden durch

unsere Arbeitsgruppe eine deutsche Übersetzung des Sprachreports zu ALGOL 68 (Kerner 1972) und eine deutsche kommentierte Fassung des revidierten Sprachreports (Kerner 1978) erarbeitet.

Da es längere Zeit international keinen Compiler gab, wurde auch in der Forschungsgruppe über Möglichkeiten der Implementation von ALGOL 68 nachgedacht. Auf Grund der geringen Größe der Forschungsgruppe war eine vollständige Implementation ausgeschlossen. Stattdessen wurde ein Compiler für eine Untersprache von ALGOL 68 – ALGOL 60+8 – im Sprachumfang von ALGOL 60 für den Rechner R 300 im Rahmen von Qualifizierungsarbeiten (Dissertationen, Diplomarbeiten) realisiert (Kerner/Al-Sheik-Khalil 1974). Das Ergebnis war gleichzeitig ein Beitrag für die Untergruppe „Untersprachen von ALGOL 68“ der WG 2.1, die von I. O. Kerner geleitet wurde. Trotz des sehr positiven Echos kam der Compiler allerdings aus objektiven (Kompatibilitätsprobleme durch Hardware) und subjektiven (Bevorzugung von Pl/1 gegenüber ALGOL 68, Bevorzugung der IBM-Rechentechnik mit entsprechenden Softwarekonsequenzen) Gründen nur in beschränkten Einsatz.

Die Syntax und die statische Semantik von ALGOL 68 waren durch den neuen Formalismus der Zweistufengrammatiken (van Wijngaarden-Grammatik) beschrieben und stellten damit eine Herausforderung für die Forschung insbesondere zum so genannten Front-End von Compilern dar. In den Arbeitspapieren der WG 2.1 zu ALGOL 68 wurde ohne Beweis behauptet, dass ein entsprechender Parser für ALGOL 68 aus der formalen Definition ableitbar sei. Allerdings selbst in der WG 2.1 gab es keine konkreten Vorstellungen dazu. Unsere Arbeitsgruppe, die sich mit Methoden der Implementation von ALGOL 68-ähnlichen Sprachen beschäftigte, schrieb sich deshalb dieses Thema auf ihre Fahnen. Als Ergebnis wurde 1971 (Riedewald 1972, 1974, 1975) aus der Zweistufengrammatik eine neue Grammatikform – die Grammatik syntaktischer Funktionen (abgekürzt GSF) – als „ausführbare“ Grammatik (parametrisierte kontextfreie Grammatik) entwickelt. Zur Zweistufengrammatik von ALGOL 68 entstand auf der Basis intensiver Analysen ihrer Zweistufensyntax eine GSF für ALGOL 68, aus der ein Compiler-Front-End ableitbar war. Ende der 70-er Jahre stellte sich die enge Verwandtschaft der GSF sowohl zu attributierten Grammatiken als auch zu logischen Programmiersprachen – speziell Prolog – heraus. Als A. Colmerauer die Idee der GSF 1993 vorgetragen wurde, formulierte er das so: „Da haben Sie ja auch die logische Programmierung erfunden.“ Weiss bewies 1979, dass Grammatiken des Chomsky-Typs 0, Zweistufengrammatiken (van Wijngaarden), Verallgemeinerte attributierte Grammatiken (Małuszyński), Affixgrammatiken (Koster) und GSF äquivalent sind. Lässt man in der GSF die Terminale weg, bekommt man eine Grammatik, die eng mit den 1984 entwickelten relationalen attributierten Grammatiken (Deransart 1984, Courcelle/Deransart 1987) verwandt ist.

Für praktische Anwendungen musste für die GSF ein Algorithmus zur Berechnung der Parameterwerte (Attributwerte) entwickelt werden. Er entstand von 1973 bis 1974, wurde 1975 in einer Diplomarbeit getestet, aber erst 1977 in einem technischen Bericht (Riedewald 1977) veröffentlicht. Einige Jahre später stellte sich heraus, dass der Algorithmus eng verwandt mit der Argumentberechnung in Prolog, allerdings ohne Backtracking, ist.

1976 wurde ebenfalls in einem technischen Bericht (Riedewald 1976) ein Compilermodell auf der Basis der GSF vorgestellt, welches den Algorithmus verwendete, und durch eine Übersetzung aus der damals üblichen akademischen Beispielsprache ASPLE (Cleaveland/Uzgalis 1973) in FORTRAN 63 illustriert. Damit wurde nachgewiesen, dass die GSF als ein wirkungsvolles Beschreibungsmittel für Sprachprozessoren geeignet ist. In Zusammenhang mit Grammar Engineering taucht die GSF noch heute, z.B. bei R. Lämmel, in der Form der logischen Grammatik auf.

Mit der semantischen Synthese von ALGOL68-Compilern beschäftigte sich die Dissertation von Lorenzen (Lorenzen 1978). Vorgeschlagen wurde eine zweistufige Codeerzeugung, wobei zunächst einmal als Zwischenstufe Makrofolgen generiert werden. Es wird ebenfalls ein Modulkonzept vorgeschlagen, um Compiler (damals noch nicht modularisierter) umfangreicher Sprachen besser zu beherrschen.

3 Attributierte Grammatiken und Anwendungen

Aufbauend auf den praktischen Erfahrungen mit dem oben erwähnten Compilermodell wurde ein System zur automatischen Erzeugung von Compilern aus GSF-Spezifikationen – RÜGEN (Rostocker ÜbersetzerGENerator) – (Riedewald/U. Lämmel 1982) konzipiert und zu großen Teilen implementiert. Basis war der Algorithmus zur Berechnung der Parameterwerte einer GSF. Wegen des großen Umfangs wurde das Thema schließlich eingeschränkt auf die Entwicklung eines Systems zur Erzeugung von Compilern für Kommandosprachen – RUEGEN-KS (U. Lämmel 1984). Auch dieses System ging von GSF-Spezifikationen aus, aber es erleichterte dem Anwender den Entwurf der GSF. Der Anwender musste eine Art Signatur der Kommandosprache definieren, die vom System auf Vollständigkeit getestet und anschließend in Regeln einer GSF transformiert wurde. Voraussetzung hierfür war ein standardisiertes Format der Kommandos der Kommandosprachen. Zum damaligen Zeitpunkt war das der erste Vorschlag zur automatischen Erzeugung von Regeln einer attributierten Grammatik.

Grammatiken syntaktischer Funktionen sind - grob charakterisiert – parametrisierte kontextfreie Grammatiken und sehr eng verwandt mit den attributierten Grammatiken. Diese Verwandtschaft wurde uns während eines Forschungsaufenthalts an der ČVUT Prag bewusst, wo wir zum ersten Mal die Gelegenheit hatten, attributierte Grammatiken intensiv zu studieren. Attributierte Grammatiken unter besonderer Beachtung der GSF und ihre Anwendungen standen für längere Zeit im Mittelpunkt der Forschung der Arbeitsgruppe. Im Rahmen zweier durch uns organisierter Tagungen mit internationaler Beteiligung versuchten wir, die Forschungen zu attributierten Grammatiken und ihren Anwendungen besser zu propagieren, um sie einem breiteren Publikum zugängig zu machen. Zusammen mit einer von der Akademie der Wissenschaften der DDR organisierten Konferenz waren diese Tagungen Keimzellen der noch heute existierenden internationalen Tagung CC, wobei sich die Bedeutung dieser Abkürzung von Compiler Compiler zu Compiler Construction änderte. Diese Entwicklung wäre ohne die Unterstützung durch Kollegen aus dem westlichen Ausland, wie z.B. Deransart, Koskimies, Fritzson, Wilhelm, die Mitglieder in den Programmkomitees waren, nicht möglich gewesen.

Die äußere Ähnlichkeit attributierter Grammatiken und logischer Programme lässt den Gedanken einer inhaltlichen Verwandtschaft aufkommen. So wurde gezeigt (Deransart/ Małuszyński 1985), dass relationale attributierte Grammatiken und logische Programme unter bestimmten Voraussetzungen aufeinander überführbar sind. Der Vorteil dieser Verwandtschaft besteht darin, dass man Erkenntnisse des einen Gebiets auf das andere übertragen kann. Diese Idee bildete die Grundlage der von Deransart und Małuszyński aus der Taufe gehobenen Konferenz PLILP (Programming Languages Implementation and Logic Programming). Nachdem in Rostock ein damals noch auf der Embargoliste stehender Compiler für die Sprache micro-Prolog ohne Sprach- und Compilerbeschreibung aufgetaucht war, begannen wir uns sofort mit dieser Sprache intensiv zu beschäftigen, da wir natürlich auch von den Plänen der japanischen Informatiker bezüglich Prolog gehört hatten. Als Nebenprodukt entstand für die Lehre das System MICALG (Forbrig 1987), welches gestattete, algebraische Spezifikationen in micro-Prolog-Programme zu transformieren. Auf diese Weise wurde eine bessere Anschaulichkeit abstrakter Datentypen in der Lehre erreicht (später wurde eine Transformation in funktionale Programmiersprachen verwendet.).

Bei der Arbeit mit Prolog fiel die Ähnlichkeit – zunächst der Notation – zur GSF auf. Es wurde gezeigt, dass sich eine GSF nach gewissen Modifikationen auf ein Prolog-Programm überführen lässt (U. Lämmel 1987, Riedewald/U. Lämmel 1988). Dabei muss nicht einmal auf Terminale verzichtet werden, wie das bei Deransart und Małuszyński oder bei der Transformation von DCG (Pereira/Warren 1980) - eine Notation logischer Programme mit Terminalen, die als spezielle attributierte Grammatik betrachtet werden kann - in Prolog-Programme üblich ist. Unsere Transformation der GSF in Prolog-Programme behielt die grammatikalische Notation einschließlich Terminale weitgehend bei. Außerdem nutzten wir an Stelle von expliziten Parsern den Berechnungsprozess in Prolog für die (absteigende) syntaktische Analyse. Die Transformation von GSF in Prolog-Programme hat auch einen weiteren praktischen Vorteil, denn man erhält ausführbare Programme. Beschreibt man z.B. einen Compiler durch eine GSF, so stellt das entsprechende Prolog-Programm einen Prototyp-Compiler dar.

Die Beziehungen zwischen GSF und logischer Programmierung bestimmten für längere Zeit die Forschungsarbeit der Gruppe. Die schnelle Umsetzung von GSF-Spezifikationen in ausführbare Prolog-Programme legt den Gedanken nahe, dieses Vorgehen für die Entwicklung von Prototyp-Sprachen, Prototyp-Interpretern und allgemein Prototyp-Software zu nutzen (Riedewald 1991). Die Spezifikation von Software mit Hilfe attributierter Grammatiken wurde 1984 durch Rechenberg am so genannten Telegrammbeispiel demonstriert. Im gleichen Jahr wurde auch bei uns durch Forbrig (Forbrig 1984, Riedewald/Forbrig 1987) ein Vorschlag zur Softwarespezifikation mit attributierten Grammatiken, speziell GSF, gemacht. Dabei wird als Neuerung die Verknüpfung von datengesteuerter Softwareentwicklung mit abstrakten Datentypen unter Verwendung attributierter Grammatiken vorgeschlagen.

Natürlich wurde auch das ureigenste Forschungsthema der Arbeitsgruppe – die Entwicklung und Implementation von Programmiersprachen – nicht vergessen. An Stelle von RÜGEN, eines Übersetzer generierenden Systems im klassischen Sinn, tritt nun ein System – LDL (Language Development Laboratory) - zur Entwicklung von Programmiersprachen und zur Erzeugung von Sprachprozessoren, wobei die Spezifikationen durch GSF und die Implementationen durch Prolog erfolgen (Riedewald 1991, 1992, 1994, Harm/R. Lämmel/Riedewald 1997). Um die Entwicklung von Programmiersprachen zu unterstützen war eine Art Expertensystem vorgesehen, welches dem Anwender eine Datenbank mit Spezifikationsmustern für gängige Sprachkonstrukte einschließlich Anwendungsdemonstrationen bieten sollte. In einer Fallstudie im Rahmen einer Diplomarbeit wurden erste Spezifikationsmuster zusammen mit Operatoren zu ihrer Bearbeitung entwickelt. Insgesamt erwies sich das Problem aber als zu umfangreich für eine kleine Arbeitsgruppe.

In diesem Zusammenhang wurde auch die Korrektheit von Interpretern untersucht (Riedewald/R. Lämmel 1994). Ebenfalls zum Thema Korrektheit von Compilern/Interpretern – aber nicht unbedingt im Zusammenhang mit LDL - wurden Möglichkeiten zur automatischen Generierung von Testprogrammen, die sowohl syntaktisch richtig sind als auch die Kontextbedingungen einer Sprache erfüllen, untersucht (Harm 1997, Harm/R. Lämmel 2000). Ausgangspunkt waren durch den Autor durchgeführte Teste mit Prototypinterpretern in PROLOG, wobei die Prolog-Eigenschaft genutzt wurde, dass, je nach konkreter Eingabe, der gleiche Parameter einmal zur Eingabe und ein anderes Mal zur Ausgabe dienen kann. Beschreibt ein Prolog-Programm eine Sprache, kann das Programm unter bestimmten Bedingungen einmal zur Analyse und einmal zur Synthese verwendet werden.

Es gab auch eine Reihe signifikanter Anwendungen von LDL: Implementierung einer Spracherweiterung zur Beschreibung von Entwurfsmustern in Eiffel, Generierung von Nutzerschnittstellen, Implementierung einer domainspezifischen Sprache für symbolische Simulation, Entwicklung und Implementierung von Integritätsbedingungen einer Sprache für objektorientierte Datenbanken. Zusammenfassend lässt sich LDL als Kombination eines Expertensystems und einer Werkzeugbank zur Entwicklung von Programmiersprachen und ihrer Prozessoren betrachten. Damit war es einerseits ein schwergewichtiges System, aber andererseits fehlten einige wichtige Komponenten für einen breiteren Einsatz.

Die obigen Forschungsarbeiten erforderten eine intensive Auseinandersetzung mit der Theorie. So entstand Ende der 70-er Jahre ein algebraisches Modell von Programmiersprachen und deren Compiler (Riedewald 1981, 1985). Im Unterschied zu anderen algebraischen Modellen von Programmiersprachen wurden die Eigenschaften von Sprachkonstrukten nicht direkt sondern über syntaktische Operationen, die den kontextfreien Syntaxregeln zur Generierung der Konstrukte entsprechen, beschrieben. Nicht die Programmiersprache selbst, sondern ihre kontextfreie Syntax wird als Algebra dargestellt. Zum ersten Mal wurde hier auch die Definition von Kontextbedingungen von Programmiersprachen durch Termgleichungen dargestellt und an einem Beispiel demonstriert.

Ein weiteres Forschungsthema war das klassische Thema der Syntaxanalyse und der Behandlung syntaktischer Fehler (Forbrig 1979, 1980).

Zyklische attributierte Grammatiken wurden am Beispiel von einfachen Datenflussanalysen studiert, wobei Prolog als Implementierungssprache diente und Zyklen durch Iterationen aufgelöst wurden.

Das Problem von Syntax und Semantik spielt in der Informatik – nicht nur bei Programmiersprachen – eine wichtige Rolle. Es ging darum, dieses Problem in die Lehre zu integrieren, wobei sich Programmiersprachen aus praktischer und theoretischer Sicht als Demonstrationsobjekt anbieten. Neben einer Reihe von Forschungsartikeln gab es international erst 1981 zur Semantik von Programmiersprachen ein einziges Buch im englischen Sprachraum (Pagan 1981), welches dafür die Basis sein konnte. In dieser Situation boten Dembiński und Małuszyński die gemeinsame Herausgabe eines solchen Buches an, welches dann 1983 als erstes deutschsprachiges Buch zur Semantik (und Syntax) von Programmiersprachen erschien und – obwohl es kein Lehrbuch war – häufig in der Lehre genutzt wurde. Es erschien sowohl beim Akademie-Verlag Berlin als auch beim Oldenbourg Verlag München Wien.

Oben wurde bereits auf das System MICALG und seinen Einsatz in der Lehre hingewiesen. Ein weiteres Nebenprodukt der Forschung für die Lehre war das System FLR (Baum/Forbrig 1991), welches zur Textgenerierung aber auch zur Manipulation von Spezifikationen genutzt werden konnte. Die theoretische Basis war die durch eine reguläre Grammatik gesteuerte Ableitung von Wörtern mit Hilfe einer kontextfreien Grammatik.

4 Programmtransformationen, symbolische Simulation

Die in Kapitel 3 angeführten Forschungen reichen bereits über die Wendezeit hinweg und waren Fortsetzungen der zu Vorwendezeiten durchgeführten Forschungen. Die im Kapitel 4 beschriebenen Forschungen starteten zwar erst nach der Wende, sie bauten aber genauso auf den vorherigen Erfahrungen und Ergebnissen auf.

In den Jahren unmittelbar nach der Wende stagnierte die Forschung zunächst, da politische und fachliche Überprüfungen der Mitarbeiter, Umstrukturierungen an der Universität, Neuausschreibungen der verbliebenen Arbeitsplätze usw. das Leben aller Mitarbeiter stark belasteten. Nachdem dann 1992 eine Neueinstellung erfolgt war, begann allmählich der Universitätsalltag wieder Fuß zu fassen, auch wenn Strukturierungsprozess und ständige Diskussionen um knappe Haushaltmittel auch heute noch die Mitarbeiter verunsichern.

Wenden wir uns nun wieder der Forschung der Arbeitsgruppe zu. Sie behielt ihre hauptsächliche Orientierung bei, allerdings zogen neue Gebiete wie Programmtransformation und symbolische Simulation ein. Konzentrieren wir uns zuerst auf Programmtransformationen. Das System LDL zeigte bereits die Notwendigkeit der Beschäftigung mit Operationen der Metaprogrammierung. Es ging dabei um Operatoren zur Entwicklung und Verknüpfung von Fragmenten von Sprachspezifikationen. Dieses Thema wurde nun allgemeiner aufgegriffen, wobei zunächst Operationen auf Fragmenten beliebiger deklarativer Beschreibungen im Mittelpunkt standen (R. Lämmel/Riedewald 1995, 1997, R. Lämmel 1998, Lohmann 2000). Deklarative Beschreibungen schlossen hier unter anderem ebenfalls Grammatiken ein, wobei damit schon die Basis für Grammar Engineering und Language Engineering gelegt wurde. Diese Themen wurden später von R. Lämmel intensiv weiter geführt.

Die hier gewonnenen Erkenntnisse spielten eine wichtige Rolle bei Programmtransformationen allgemeiner Art (R. Lämmel/Riedewald 1998, 1999, R. Lämmel/Riedewald/Lohmann 1999, Lohmann/Riedewald 2003, Lohmann/Riedewald/ Stoy 2004). Z.B. wurde das Thema der Kommentarerhaltung bei Programmtransformationen sehr intensiv untersucht (Lohmann/Riedewald 2003). Da es nach wie vor nur als teilweise gelöst betrachtet werden kann, bleibt es ein aktuelles Forschungsthema.

Es wurde bereits erwähnt, dass LDL ein schwergewichtiges System zur Entwicklung von Sprachen und deren Prozessoren darstellt. In den letzten Jahren zeichnet sich ein Trend zur Bereitstellung leichtgewichtiger Systeme von Sprachprozessoren ab, die auch von Nichtspezialisten unter gewisser Anleitung verwendet werden können. Wir haben uns diesem Trend angeschlossen und das System Laptob (Language Processing Toolbox) entwickelt (R. Lämmel/Riedewald 2001). Dieses System ist ein Rahmenwerk mit Werkzeugunterstützung zur Manipulation von Sprachprozessoren in Form logischer Programme. Es gibt dabei typisierte logische Programme sowie logische Programme höherer Ordnung. Grammatiken können Bestandteile von Prozessoren sein. In der Werkzeugbank werden Operatoren (Metaprogramme) zur Adaption und Komposition von Prozessoren bereitgestellt. Durch Benutzung des Systems können seine Möglichkeiten auf der Basis der Wiederverwendung ausgebaut werden.

Durch die Zusammenarbeit mit der Universität Maribor (M. Mernik) profitierte nicht nur die Forschung an attributierten Grammatiken und deren Anwendung im Bereich der Sprachdefinitionen und Sprachimplementationen, sondern es begann eine intensive Beschäftigung mit aspektorientierter Programmierung. Als hervorragendes Ergebnis entstand ein Vorschlag zur aspektorientierten Programmierung in Prolog (Lohmann/Riedewald/Wachsmuth 2006, 2008). Es ist international der erste umfassende Vorschlag zur Aspektorientierung in Prolog überhaupt. Auf der Basis dieses Vorschlags gelang es, wohlbekannte Techniken der Prologprogrammierung zum ersten Mal formal und einheitlich zu beschreiben. Die Forschungen zur Aspektorientierung in Prolog sind erst am Anfang und daher fehlen zurzeit natürlich bequeme Vorgehensweisen für den Alltagsprogrammierer. So wäre es z. B. sinnvoll, dem Programmierer eine domainspezifische Sprache zur bequemen aspektorientierten Programmierung in Prolog zur Verfügung zu stellen.

Im Zusammenhang mit Untersuchungen zu attributierten Grammatiken ergab sich die Frage, inwieweit Sprachen wie SDL (Specification and Description Language) zur Beschreibung von Echtzeitsystemen mit Hilfe attributierter Grammatiken definierbar und implementierbar sind. Ausgangspunkt war, dass z.B. SDL-Programme Netze semantisch erweiterter endlicher Automaten definieren, wobei Instanzen von Automaten während der Arbeit mit den Netzen gelöscht oder generiert werden können. Die Kommunikation der Automaten geschieht durch Signale über Kanäle. Da Netze endlicher Automaten allgemein kontextfreie Sprachen akzeptieren, sollten attributierte Grammatiken zur Beschreibung ausreichen, wobei die semantische Erweiterung der Automaten über den Formalismus der Attribute und semantischen Regeln darstellbar sein müsste. Diese Idee erwies sich als tragfähiges Konzept, zumal durch den Zusammenhang zwischen attributierten Grammatiken und Prolog eine schnelle Prototypimplementierung möglich war.

Bei der Beschäftigung mit dieser Problematik stießen wir auf die zeitabhängigen Automaten (Alur/Dill 1994) und hybride Automaten/Systeme (Alur/Courcoubetis/Henzinger/Pei-Hsin-Ho). Diese beschrieben wir zunächst durch eine GSF, die dann in ein Prolog-Programm transformiert wurde. Wegen der in den Automaten allgegenwärtigen Bedingungen (bedingte Übergänge, Invarianten in Lokationen) und wegen der Verwendung der Automaten zur symbolischen Simulation erwies sich aber eine Beschreibung der Automaten durch eine logische Programmiersprache mit Nebenbedingungen – auch constraint-logische Programmiersprachen genannt - (CLP, Prolog III, Prolog IV) als bequemer (Urbina/Riedewald 1995, Riedewald 1995), wobei man natürlich nicht den Umweg über GSF oder andere attributierte Grammatiken gehen muss. Hierbei spielte unsere Gruppe international eine Vorreiterrolle.

Eine Beschreibung hybrider Systeme durch constraint-logische Programmiersprachen ist zwar leicht erlernbar, aber für den Nutzer nicht besonders bequem. So entstand die Idee, zwei Sprachen für eine einfachere Handhabung zu entwickeln: eine textuelle und eine graphische Sprache (analog zu SDL). Folgende Forderungen sollten von den Sprachen erfüllt werden:

- Modulare Sprachdefinition orientiert an den einzelnen Automaten eines Systems
- Wiederverwendbarkeit von Automatenbeschreibungen durch Einführung von Automatentypen und parametrisierten Automaten
- Möglichkeiten einer Automatenverfeinerung
- Nutzerfreundliche Darstellung von Anfragen
- Synchronisation über Signale

Als Ergebnis entstanden die Sprachen MODEL-HS (Tetzner/Riedewald 1998, 1999) und VYSMO (Tetzner/Brauch/Riedewald 2001), die obige Forderungen im Wesentlichen erfüllen. Beide Sprachen sind äquivalent und werden in constraint-logische Programme übersetzt (Tetzner/Riedewald 2003). Als umfangreiche Anwendung wurde die Modellierung von Studiengängen untersucht (Tetzner/Lantsman/Riedewald 2006). Unter der Voraussetzung, dass die zeitlichen Abhängigkeiten eines Studiengangs einschließlich der Bedingungen (z. B. Wiederholungsfristen von Prüfungen, absolvierte Lehrveranstaltungen vor einem Praktikum oder der Diplomarbeit,...) aus Prüfungs- und Studienordnung als hybride Automaten spezifiziert werden, kann man sich Vorschläge für komplette konkrete Studienabläufe oder Möglichkeiten der Fortsetzung eines Studiums in einer bestimmten Studiensituation (bestimmte Studienleistungen wurden zu einem gegebenen Zeitpunkt erbracht/nicht erbracht) durch symbolische Simulation generieren lassen. Da einige Teile des Gesamtsystems noch per Hand erzeugt werden, ist das System noch nicht fremdnutzbar.

5 Schlussbemerkungen

In den vorherigen Kapiteln wurde der Weg der Arbeitsgruppe „Programmiersprachen und Übersetzertechnik“ auf dem Gebiet der Forschung skizziert. Dieser Weg war zum einen mit Problemen durch das politische System der DDR und zum anderen mit lokalen Problemen gepflastert, die nur teilweise überwindbar waren. Sehen wir uns einmal einige davon an.

Literaturbereitstellung Hier war das größte Problem ein chronischer Mangel an Devisen zur Beschaffung westlicher Literatur. Teilweise ausgenommen davon waren Forschungsschwerpunkte.

Der Devisenmangel führte dazu, dass sowohl wichtige Zeitschriften als auch Monographien nicht gekauft werden konnten. International sehr erfolgreiche Monographien wurden manchmal allerdings von sowjetischen Verlagen in Russisch herausgegeben. Solche Bücher waren dann sogar zu relativ niedrigen Preisen erwerbbar, wenn man sich rechtzeitig darum bemühte. Durch den so entstandenen Mangel an westlicher Literatur waren die Forschung – und in gewissem Maße – auch die Lehre beeinträchtigt. Ein Ausweg aus dieser Situation war die Anforderung von Sonderdrucken bei den Autoren oder die Beschaffung von Kopien bei Auslandsreisen, wobei persönliche Kontakte eine hervorragende Rolle spielten. In den Jahren nach der Wende konnten die Bibliotheken ihre Bestände auf den neuesten Stand bringen, aber in den letzten Jahren begann erneut ein strenges Sparregime mit dem Streichen bei der Beschaffung von Zeitschriften.

Was die Lehrbuchbeschaffung betrifft, konnte man vor der Wende kaum westliche Literatur verwenden, falls sie nicht gerade in Russisch zur Verfügung stand, allerdings stand genug Literatur zu studentengemäßen Preisen ostdeutscher Autoren zur Verfügung.

Veröffentlichungen Das Hauptproblem vor der Wende ist, dass an Konferenzen im westlichen Ausland nur Reise- und Auslandskader (und auch das nur in bescheidenem Maße) teilnehmen konnten. Eine Veröffentlichung im westlichen Ausland über Proceedings war deshalb nur wenigen vorbehalten. Veröffentlichungen in westlichen Zeitschriften dagegen waren prinzipiell möglich, aber durch ein Genehmigungsverfahren, wo z.B. die Notwendigkeit der Veröffentlichung im westlichen Ausland begründet werden musste, erschwert. Auf diese Weise war eine Propagierung von Forschungsergebnissen in westlicher Richtung schwierig, so dass – mit Ausnahmen - weder die Forscher aus der DDR noch ihre Ergebnisse im Westen bekannt waren.

Ressourcen Das Problem der Bereitstellung von Ressourcen ist ein ständiges Problem der Hochschulen. Eine einfache Formel der Verteilung lautet: Ist dein Thema ein Schwerpunktthema, dann wirst du auch gefördert. Wegen der in der Summe knappen Mittel heißt das dann für andere Mangel an Ressourcen. Die Frage ist, wie die Schwerpunkte festgelegt werden. Vor der Wende wurden z.B. gemeinsame Projekte mit der Industrie gefördert. Heute ist es wichtig, besonders viele Drittmittel einzuwerben, wobei nicht immer nach dem Nutzen aus wissenschaftlicher Sicht, insbesondere der Grundlagenforschung, gefragt wird.

Persönliche Kontakte Auf die Bedeutung persönlicher Kontakte für die Forschung wurde bereits im Zusammenhang mit den Forschungen der Arbeitsgruppe eingegangen. Deshalb wollen wir an dieser Stelle darauf verzichten.

Trotz aller genannten Probleme kann die Forschungstätigkeit der Arbeitsgruppe insgesamt als erfolgreich sowohl vor als auch nach der Wende eingeschätzt werden. Aus der obigen Beschreibung der Forschung geht ebenfalls die Kontinuität von der Vor- zur Nachwendezeit hervor, was nicht heißt, dass immer wieder die gleichen Forschungsfragen bearbeitet wurden.

Die Ausbildung von Informatikern in Dresden – frühe Anfänge

Angela Buchwald

Universitätsarchiv
TU Dresden
01062 Dresden
angela.buchwald@tu-dresden.de

Abstract: Die Traditionslinie der heutigen Fakultät Informatik der TU Dresden wird bis zu ihren Anfängen zurückverfolgt und der Ursprung der Informatikausbildung in Dresden ergründet. Dabei wird davon ausgegangen, dass die Wurzeln der Wissenschaft Informatik zu großen Teilen auf den verschiedenen Gebieten der Elektrotechnik/Elektronik zu suchen sind. Gleichfalls ist es Anliegen, die Ausbildungsmöglichkeiten auf diesem Gebiet in den verschiedenen Zeitabschnitten darzustellen. Das Thema tangiert einerseits die Entwicklung der Rechentechnik und ansatzweise den Wandel im Sprachgebrauch für Sachverhalte der Informatik. Die Ausführungen sind eine erste Quellenpublikation, gestützt auf Akten des Universitätsarchivs, die vor allem Anregung für weitere Forschungen geben soll.

1 Einstimmung

Als Archivarin gehört es nicht nur zu meinen Aufgaben, das Schriftgut vergangener Zeiten sicher aufzubewahren, sondern es ist gleichfalls zu sichten, zu ordnen, zu verzeichnen und auszuwerten. Dabei habe ich u. a. Kenntnisse über die Entwicklung der Fakultät Informatik und ihrer Vorgängereinrichtungen erhalten, die mir wertvoll genug erscheinen, sie der Öffentlichkeit vorzustellen. Der Aufschrei, der Anfang der 90-er Jahre des 20. Jahrhunderts durch die Medien ging, dass es einen eklatanten Mangel an Informatik-Spezialisten gäbe, ließ auch mich aufhorchen. Vor allem die zur Begründung gegebene Meldung, dass erst ab Mitte der 70-er Jahre die entsprechende Ausbildung durchgeführt wurde, stand im Widerspruch zu dem, was ich aus Primärquellen während meiner Tätigkeit herausgelesen hatte. Nun ist es m. E. an der Zeit, die angesammelten Fakten zu bündeln und damit für Dresden nachzuweisen, dass der Beginn der neuen Wissenschaftsdisziplin „Informatik“ nicht verpasst wurde. Vielmehr gab es hier frühzeitig engagierte Bestrebungen, den Anforderungen moderner Technologien im weitesten Sinne durch Aus- und Weiterbildung gewachsen zu sein.

Beim Überdenken meines Themas fielen mir sofort zwei große Namen ein, die mit Computer in Zusammenhang zu bringen sind: N. J. Lehmann und Konrad Zuse. Über beide wurde bereits mehrfach publiziert, so dass ich sie hier nur nenne. Im Archiv fand ich Hefte zur **Vorlesungsreihe „Geschichte der Informatik“**, die Dozent Dr. Klett 1992 hielt. Da es mir als Nicht-Informatikerin kaum gelingen kann, inhaltsreicher zu sein, möchte ich diese und eventuell nachfolgende Hefte Interessenten weiterempfehlen und hier darauf verzichten allgemein anerkannte Informatik-Vorläufer darzustellen. So beschränkt sich mein Vortrag auf die Entwicklung der Informatik-Ausbildung hier in Dresden, deren Wurzeln ich in den verschiedenen Zweigen der Elektrotechnik / Elektronik sehe. Unter diesem Gesichtspunkt stelle ich Ihnen nun die Vorgängereinrichtungen der Fakultät Informatik vor, derer sich die Wissenschaftler heute nicht schämen müssen, schließlich kann die Fakultät 2001 de facto auf 150 Jahre Geschichte zurückblicken.

2 (Städtische) Gewerbeschule Dresden 1861 – 1926 (GS)

Die Gewerbeschule Dresden wurde als private Schule von Karl Wilhelm Clauß (*15.3.1829 – †4.11.1894) gegründet und lange Jahre geführt. Sie ist ab 1896 städtische Bildungseinrichtung und bekam 1901 sogar einen Schulneubau im Quartier Elisen-Dürer-Marschner-Gerok-Straße mit Adresse Dürerstraße 45 (vorherige Anschriften: u. a. Weiße Gasse 4, Maxstraße 9). Nachweisbar sind bereits an der GS Dresden 6 Schüler, die von Beruf Elektromechaniker / Elektrotechniker waren. Wegen der Quellenlage gilt diese Angabe nur für den Zeitraum 1886 – 1913. Außerdem gab es neben zahlreichen anderen Fachgebieten die FR Elektrotechnik ab 1895/96 (eingerichtet und geführt von Oberlehrer Hans Benisch[1]).

3 Technische Lehranstalten Dresden 1926 – 1952 (TLA, auch Städtische Ingenieurschule)

An den TLA wurde zusätzlich zur seit 1922 bestehenden Technischen Mittelschule ab Ostern 1928 eine **Höhere Maschinenbauschule** eingerichtet, welche per 5.12.1929 offiziell durch den Gutachterausschuss für technisches Schulwesen beim Reichsministerium des Innern als Ingenieurschule anerkannt wurde. Die Rekordschülerzahl an den TLA im Jahr 1926 mit ca. 2500 Studierenden ließ Pläne für einen Erweiterungsbau der Einrichtung entstehen. Dieser konnte im Januar 1930 auf dem Grundstück Elisenstraße 25 seiner Bestimmung übergeben werden. In Dresdens Schicksalsnacht 13./14.2.1945 wurden auch die Gebäude der TLA zerstört, jedoch hörte damit die Schule nicht auf zu existieren. Beinahe unverzüglich, im Oktober 1945, wurde der Lehrbetrieb wieder aufgenommen, wenngleich die Ingenieurausbildung erst wieder ab 1947 durch die Sowjetische Militäradministration in Deutschland (SMAD) erlaubt wurde.

[1] IHD 3142/1 und IHD 3142/2 „Statistiken zur Geschichte der Einrichtung“

Da sich das attraktive Gebäude an der Dürerstraße nach der Zerstörung nicht mehr mit vertretbarem Aufwand wiederherstellen ließ, war nun das Gebäude Elisenstraße 25 offizielle Adresse der Einrichtung. An den TLA gab es entsprechend der hier zu untersuchenden Fachgebiete die FR Elektrische Anlagen, Kabel und Geräte, FR Elektrotechnik und FR Elektrotechnische Anlagen und Geräte.

4 Intermezzo 1952 – 1962

Die Schule wurde zwischen 1952 und 1962 mehrfach umbenannt, was z. T. mit Strukturveränderungen einherging. Nachstehende Übersicht zeigt, dass bis auf 1 Jahr stets in der Fachrichtung Elektrotechnik mit ihren verschiedenen Abwandlungen ausgebildet wurde.[2]

FS für Maschinenbau, Elektrotechnik und Feinmechanik 1952 – 1953
- FR Elektrotechnik
- FR Elektro-Fernmeldewesen
- FR Elektroanlagen und Geräte
- FR Elektromaschinen

FS für Schwermaschinenbau, Elektrotechnik und Feinmechanik 1953 – 1955
- FR Elektrotechnik
- FR Elektrische Anlagen und Geräte
- FR Geräte und Büromaschinen

FS für Leichtbau 1955 – 1956
- FR Gerätebau
- FR Flugzeugbau (konnten u. a. als Flugzeugelektromechaniker arbeiten)

IS für Flugzeugbau 1956 – 1962[3]
- FR Elektrische Anlagen und Geräte
- FR Elektrische Anlagen und Geräte / Elektromaschinen
- FR Elektrische Netze
- FR Elektroanlagenbau
- FR Elektrofeinwerktechnik / Fertigung (ab 1961)
- FR Elektromaschinenbau
- FR Elektrotechnik
- FR Funkgerätebau
- FR Gerätebau / Elektriker oder Mechaniker

[2] Laufende Studiengänge FR Elektrotechnik und FR Elektrische Anlagen und Geräte konnten auch im Zeitraum 1955 – 1956 ihre Ausbildung fortsetzen. Durch die Gründung der IS für Flugzeugbau wurde das Konzept, auf die Ausbildung in Elektrotechnik zu verzichten, sehr schnell korrigiert.
[3] Die IS für Flugzeugbau war dem Vernehmen nach europaweit die einzige Einrichtung, an der ausschließlich spezielle Fachkräfte für die Flugzeugindustrie ausgebildet wurden.

5 IS für Maschinenbau und Elektrotechnik Dresden 1962 – 1969 (ISME)

Bestimmten zum Beginn der 60-er Jahre vor allem Maschinenbaufächer die Lehrinhalte, beendeten bereits 1966 die ersten Absolventen in der FR Elektronik ihr Studium.

Ohne die Maschinenbaufachrichtungen, v. a. im Abendstudium, aufzugeben, wurde die Seite der Elektronik um die verschiedenen Vertiefungsrichtungen der Elektronischen Datenverarbeitung erweitert und damit der Grundstein für die geplante Umwandlung in eine Ingenieurhochschule mit reinem EDV-Profil gelegt. So erhielten bereits in den Jahren 1963 und 1964 Studenten der FR Elektronik eine vertiefende Ausbildung in Hinblick auf die Datenverarbeitung.[4] Nach Errichtung eines Neubaus 1957/58 entlang der Dürerstraße und Aufstockung Mitte der 60-er Jahre des 20. Jahrhunderts erhielt der Gebäudekomplex der Ingenieurschule seine heutige Gestalt, in welchem noch bis Sommer 2006 Studenten und Mitarbeiter der Fakultät Informatik rege ein- und ausgingen.

Beim Aktenstudium wurde meine Aufmerksamkeit auf einige m. E. bedeutsame Ereignisse / Dokumente gelenkt, die ich Ihnen gern zur Kenntnis geben möchte.

5.1 Konzeption zur Gestaltung der Ausbildung auf dem Gebiet der Mathematik und elektronischen Datenverarbeitung bis 1970 von 1963

In o.g. Dokument wird die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden als Ausbildungsstätte für die neuen Komplexe der elektronischen Rechentechnik, Rechenanlagen und Datenverarbeitung für Wartungs- und Konstruktionsingenieure der elektronischen Datenverarbeitung vorgeschlagen. Zu diesem Zeitpunkt existieren in Dresden bereits zwei große Rechenzentren.[5] Außerdem wird eingeschätzt, dass der Lehrkörper der Ingenieurschule gute Voraussetzungen bietet für die Weiterbildung in elektronischer Datenverarbeitung.

Man plante, für den Zeitraum 1966 – 1969 ca. 300 Absolventen der Industrie zur Verfügung zu stellen. Die zunächst technisch orientierte Ausbildung sollte durch die Lehre in industriezweiggebundener Organisationstechnik ergänzt werden. Die Neuprofilierung der Fachrichtung Automatisierungstechnik unter Berücksichtigung der Analog- und Digitalrechentechnik sowie numerischer Probleme der Prozessautomatisierung wird eingefordert. Ein entsprechender Lehrplan war bis 1.9.1964 zu erarbeiten. Außerdem hatte die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden die Zusatzausbildung in elektronischer Datenverarbeitung und Programmierung für Fachschulabsolventen, die später organisatorisch in Betrieben und Lochkartenstationen eingesetzt sind, mit Beginn 1.9.1964 federführend zu realisieren.

---

[4] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“

[5] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“ Leider war aus den mir vorliegenden Unterlagen nicht ersichtlich, welche Rechenzentren gemeint waren. Vermutet wird, dass der VEB Robotron und der VEB Maschinelles Rechnen Dresden hier die entsprechenden Kapazitäten hatten.

5.2 Industriekonferenz am 14.2.1964

Am 14.2.1964 veranstaltete die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden als Beitrag zur engeren Gestaltung der Beziehungen zu den Industriezweigen, in denen die Absolventen eingesetzt werden, eine **Industriekonferenz**.

Angesprochen und eingeladen wurden dafür mindestens 24 Betriebe und Einrichtungen.[6] Ziel war, die Ausbildungsrichtungen der Ingenieurschule in einem breiten Kreis bekannt zu machen und, wo möglich, vertragliche Vereinbarungen mit den Betrieben und Einrichtungen zu treffen. Die entsprechenden Vertragsentwürfe (über Durchführung von Exkursionen, über Erfassung und Vorbereitung von Studienbewerbern, über Praktika und über Lösung von Industrieaufgaben) wurden den Industrievertretern am Ende der Tagung zur Bestätigung durch die Werkleiter übergeben. In Übereinstimmung mit der „Richtlinie des Staatssekretariats für Hoch- und Fachschulwesen über die Zusammensetzung, die Aufgaben und die Arbeitsweise der Beiräte an den Ingenieurschulen“ vom 15.1.1960 wurde die Konstituierung eines **Beirates der Ingenieurschule** angestrebt. Ihm sollten neben den Vertretern der Ingenieurschule Abgesandte von 11 weiteren Einrichtungen, darunter auch die TUD, angehören.[7]

Außerdem wurden u. a. „Entwicklungsstand und Perspektive des Fachgebietes Elektronik“, die Frage „Ist eine spezielle Ausbildung von Ingenieuren für die Datenverarbeitung notwendig?“ und „Stand der Automatisierung in der Fertigung (Behandlung der Programmierung im Unterricht)“ diskutiert. In der Arbeitsgruppe „Feinwerktechnik und Elektronik“ konstatiert ein Vertreter der Ingenieurschule für Maschinenbau und Elektrotechnik Dresden (Wilhelm Leupold), dass **Absolventen** der Einrichtung bereits **in Betriebe der Datenverarbeitung vermittelt** wurden und sich dort gut bewährt hätten. Bemängelt wird von einem Industrievertreter, dass es **keine einheitliche Erklärung für den Begriff „Elektronik“** gibt.[8]

5.3 Gutachten zur Frage der Ausbildungsstätten der elektronischen Datenverarbeitungsanlagen vom 6.11.1963

In den Unterlagen zur Industriekonferenz befindet sich ein (leider nicht autorisiertes) „**Gutachten zur Frage der Ausbildungsstätten der elektronischen Datenverarbeitungsanlagen**“ vom 6.11.1963. Darin geht man von 600 Wartungsingenieuren für Elektronische Datenverarbeitungsanlagen (EDVA) aus, die im Zeitraum 1966 – 1970 von der Industrie benötigt werden. Die Einrichtung spezieller Studiengänge für Wartungsingenieure sei aber dennoch nicht gerechtfertigt und wäre abzulehnen, waren sich die Vertreter der in Frage kommenden Ausbildungsstätten einig. Vielmehr sollten die Potentiale der Fachrichtungen „Hochfrequenztechnik und Elektronik“ und „Feinwerktechnik“ durch zusätzliche Vermittlung von Kenntnissen in maschineller Rechentechnik, Rechenanlagen und Datenverarbeitung, verbunden mit

[6] ISME 7 – 014 „Industriekonferenz am 14.2.1964“
[7] ISME 7 – 014 „Industriekonferenz am 14.2.1964“
[8] ISME 7 – 014 „Industriekonferenz am 14.2.1964“

Praktika im Rechenzentrum genutzt werden. Die Ingenieurschulen in Glashütte und Dresden waren als Ausbildungsstätten vorgesehen. Eine Variante sah die Zentralisierung der Ausbildung in Dresden vor, da hier 2 Rechenzentren arbeiteten und die Ingenieurschule selbst sehr gute Voraussetzungen bot.

Eine andere verwies darauf, dass im Lehrplan „Gerätetechnik" der Ingenieurschule Glashütte bereits Gesichtspunkte der Elektronischen Datenverarbeitung berücksichtigt sind. Man rechnete mit jährlich 90 Absolventen. Für den sofortigen Beginn der Ausbildung von Wartungsingenieuren waren 60 zusätzliche Absolventen avisiert.[9]

5.4 Vorlage des Staatssekretariats für Hoch- und Fachschulwesen für die Sitzung der staatlichen Kommission für Elektronik beim Ministerrat der DDR im Januar 1965

Während das Staatssekretariat für Hoch- und Fachschulwesen (SHF) in einer Vorlage für die Sitzung der staatlichen Kommission für Elektronik beim Ministerrat der DDR im Januar 1965 einschätzt, dass die Ausbildungskapazität der Fakultät für Elektrotechnik an der TUD bis 1970 kaum erweitert werden kann, ist in gleichem Dokument von einer **Umprofilierung der Ingenieurschule für Maschinenbau und Elektrotechnik Dresden** die Rede. Deren Gesamtkapazität von 420 Zulassungen pro Jahr soll ausschließlich der Ausbildung von Ingenieuren und Ingenieurökonomen auf dem Gebiet der **Datenverarbeitung** dienen. Diese Ausbildungsrichtung ist bei keiner anderen Universität, Hoch- oder Fachschule so explizit genannt.[10] **Somit wäre die Dresdner Ingenieurschule tatsächlich die erste Bildungseinrichtung gewesen, an der die Ausbildung von Spezialisten in der FR Elektronische Datenverarbeitung auch unter dieser Bezeichnung ab 1965 realisiert wurde!** Eine Vorlesungsreihe „Elektronik" und das „Elektronik-Praktikum" waren ab 1966 u. a. an der TUD vorgesehen. Für den Einsatz in der Elektronik-Industrie werden im Übrigen die Diplom-Physiker aller Universitäten und Hochschulen empfohlen.[11]

[9] ISME 7 – 014 „Industriekonferenz am 14.2.1964"
[10] ISME 7 – 012 „AG beim SHF 1964 – 1965"
[11] ISME 7 – 012 „AG beim SHF 1964 – 1965"

5.5 Die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden als Weiterbildungszentrum

Die Ingenieurschule verstand sich als **Weiterbildungszentrum** auf dem Gebiet der elektronischen Datenverarbeitung. Es wurden u. a. Kurse für leitende Kader der Volkswirtschaft und des Staatsapparates im Rahmen von Freundschaftsverträgen organisiert. Die Lehrgänge der VVB Datenverarbeitung in Waldbärenburg wurden mit Technik der Ingenieurschule unterstützt. Hinsichtlich der Industrieverbindungen wird konstatiert, dass man mit allen größeren Betrieben des Industriezweiges gute Zusammenarbeit pflegt.[12] Fortbildungswillige hatten die Möglichkeit, sich in 1 1/2 jährigen Abendschul-Kursen vom Techniker zum Ingenieur zu qualifizieren. Es gab jeweils Sonderklassen für ehemalige Offiziere (Armeeklassen), Parteikader und vorzeitige Abgänge aus Universitäten und Hochschulen in der FR Elektronik.

Ab November 1966 bietet die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden ein 18 Monate dauerndes **postgraduales Studium der Fachrichtung Elektronische Datenverarbeitung** für Ingenieure der Elektrotechnik sowie der Regelungstechnik an.[13] Seine Erweiterung findet dies im November 1967 und Sommer 1968, als man 2jährige postgraduale Studiengänge mit Abschluss „Fachingenieur für elektronische Datenverarbeitungsanlagen“, sowie eine zertifizierte Ausbildung auf dem Gebiet der „Ingenieurökonomie“ und „Grundkenntnisse der maschinellen und elektronischen Datenverarbeitung“ vorbereitet. Diese Kurse sollten einerseits an der Ingenieurschule direkt aber auch in größeren Betrieben, die dazu von der Ingenieurschule betreute Außenstellen einrichteten, als Abendstudium mit 12 Unterrichtsstunden pro Woche durchgeführt werden. Inhaltlich standen neben dem Schwerpunkt „Erkennen der zur Programmierung geeigneten Prozesse und Informationssysteme“ auch

„Politische Ökonomie“
„Kybernetik“
„Sozialistische Wirtschafts- und Menschenführung“
„Ökonometrie“
„Organisation und Planung der Produktion“
„Grundkenntnisse der Programmierung von Digitalrechnern“

auf dem Plan. Im Fach „Maschinelle Rechentechnik“ lernten die Studierenden die Lochkartentechnik, Lochkarten-Kleinmaschinen (Locher und Prüfer) sowie Lochkarten-Großmaschinen (Sortiermaschine, Tabelliermaschine, Bulldoppler, Elektronenrechner ASM 18, Motorblocksummenlocher Type 440) kennen. Laut Stoffplan „Programmierung“ gab es Unterweisungen in Blockschaltbild, Ein- und Mehradress-maschinen, Flussdiagramme, Programmierung SER 2c und R300, sowie Einführung in Programmiersprache ALGOL.[14]

[12] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“
[13] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[14] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“

Bewahrheitet hat sich eingedenk heutiger Erfahrungen die Richtigkeit der Erkenntnis, dass „der Einsatz der EDV mehr als nur Spezialisten erfordert“. Nicht zuletzt regte man an, dass sich alle Hoch- und Fachschulabsolventen Grundkenntnisse der Informationsverarbeitung aneignen und in regelmäßiger Weiterbildung die jeweils neuesten Erkenntnisse der EDV kennen lernen sollen.[15]

1965 und 1966 wurden die **Labors** für Elektronik, elektronische Regelungstechnik, elektronische Rechentechnik und eine Lochkartenstation neu aufgebaut, wobei die Beschaffung der entsprechenden Geräte beim zuständigen Versorgungskontor nicht unproblematisch war, was Verzögerungen zur Folge hatte.[16] Aus dem Automatisierungslabor wurde ein Labor der technischen Systeme.[17]

5.6 Die Fachrichtung Elektronische Datenverarbeitungsanlagen

An der Ingenieurschule für Maschinenbau und Elektrotechnik gab es ab 1.12.1966 eine **FR Elektronische Datenverarbeitungsanlagen** (EDVA, unter Leitung von Wilhelm Leupold, später Professor an der TUD)[18] mit Labors für Grundlagen der Elektrotechnik, Elektronische Regelungstechnik, Elektronische Rechentechnik und Elektronik-Labor.[19] Per 1.9.1966 wurde mit der Ausbildung von Programmierern für EDVA begonnen.

Eine undatierte Konzeption zur **Ausstattung der Ingenieurschulen mit Rechentechnik** nennt eine Mindestausstattung für die Rechenstationen. Demnach sollen entsprechend der „Faustregel“ 1500 Studierende bekommen 1 Rechner + 1 Programmierer + 1 Betreuungskraft + 1 Hilfskraft folgende Geräte vorhanden sein:

> Klein-Elektronenrechner D4a, SER 2b mit dazugehöriger 1. Peripherie (Lochbandmaschine)
> Geräte der 2. Peripherie (Schreibautomat Optima 528 oder Org.-Automat Ascota 7000)
> ein Analogrechner je nach Aufgabengebiet der Schulen[20]

1967/68 war vorgesehen, noch weitere Geräte (Motorblock-Summenlocher, Doppler, ASM 18, Zentraleinheit des R100, SER 2c, D4a, 3 Fakturierautomaten) angeschafft anzuschaffen. Für die Fachgruppe Mathematik wollte man im Rechenkabinett insgesamt 16 Tischrechner aufstellen.[21]

---

[15] SZ vom 19.6.1968 in ISME 7 – S – 08
[16] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“ Man beschwerte sich über die späte Bewilligung der Haushaltmittel durch das Staatssekretariat für Hoch- und Fachschulwesen und den zeitlich vorher liegenden Bestellannahmeschluss beim Versorgungskontor.
[17] SZ vom 13.11.1967 in ISME 7 – S – 08
[18] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“
[19] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[20] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[21] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“

Außerdem war ab September 1968 eine **Rechenstation** konzipiert. Neben der regulären Ausbildung für die Studenten kümmerte man sich in diesem Bereich um die Weiterbildung der Mitarbeiter. So ist die Teilnahme am „6. Kolloquium über elektronische Rechengeräte“ 1967 genauso verbürgt wie die am „Seminar für die Anwender der elektronischen Analogrechner vom Typ MEDA“ im Herbst des gleichen Jahres. Ein solcher Rechner „MEDA 20 TS“ nebst Zubehör „SENAS 2-10/B“ (Servomultiplizierer) und „BAK II“ (Koordinatenschreiber) gehörte 1968/69 neben dem elektronischen Kleincomputer „Cellatron C 8201 (C 8205)“ zur technischen Ausstattung des o. g. Labors für elektronische Rechentechnik. Ferner wurde die Weiterbildung für „Programmierung R 300“ wahrgenommen.[22]

Die FR EDVA arbeitete in zahlreichen Kommissionen und Arbeitsgruppen auch regional übergreifend mit. In der Zentralen Fachkommission „Informationsverarbeitung“ wurden Ingenieurschulen als Ausbildungszentren für Datenverarbeitung bestimmt. Die Dresdner Ingenieurschule steht dabei in einer Reihe mit den Ingenieurschulen in Görlitz, Zwickau, Wismar, Köthen, Lichtenberg, Schmalkalden, Rodewisch, Gotha, Glashütte und der Ingenieurschule für Automatisierungstechnik Leipzig.[23]

5.7 Andere Leistungen der Ingenieurschule für Maschinenbau und Elektrotechnik Dresden

Bereits ab 1966 regte eine **AG „Rationalisierung der Verwaltungsarbeit“** an, die Lochkartentechnik für

die Erfassung der Studienbewerber (praktiziert ab Immatrikulationsjahr 1967)
der Grundmittel inkl. Strukturfestlegung
die Lohn- und Gehaltsabrechnung einschließlich Stipendium
(Untersuchungen dazu erfolgten im Rahmen einer Belegarbeit der SG 65 Ö 02)
die Fachschulstatistik und Statistiken des Bereichs Studienorganisation

anzuwenden. Diese Arbeiten sollte das ingenieurschuleigene Lochkartenlabor (Lochkartenstation) ausführen, das im Dezember 1966 als völlig unausgelastet beschrieben wird.[24]

Die Ingenieurschule gab 1966/67 für das Fernstudium die **Lehrbriefreihe** „Elektronische Rechentechnik“ (7 Hefte) heraus.[25] In einer Konzeption zum Lehrstoffkomplex „Kybernetik und Informationsverarbeitung“ vom 15.12.1967 wird bereits der Grundsatz formuliert, dass „**Informationsverarbeitung**“ als Begriff die Rechentechnik und Datenverarbeitung, also Rechnerorganisation und Programmierung, meint, sowie die analoge und digitale Darstellung und Verarbeitung von Größen beinhaltet.[26]

---

[22] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[23] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[24] ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“
[25] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
[26] ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“

Zur „II. Zentralen Leistungsschau der Studenten und jungen Wissenschaftler“ 1967 stellten Studenten einen „Netzwerksimulator“ (u. a. Dieter Jungmann, später Prof. an der TUD) und ein „Binäres Rechenwerk“ vor.[27] Es sind 17 Fachrichtungen an der ISME nachweisbar, die ich zu den **Vorläufern** der neuen Wissenschaft **Informatik** zählen möchte. Darunter FR Elektronische Datenverarbeitungsanlagen (EDVA ab 1965 und als einzige Ausbildungsstätte in der DDR), FR Elektronik (ab 1963), FR Ingenieurökonomie der Datenverarbeitung (ab 1963). FR Programmierung von EDVA (ab 1966 und als einzige Ausbildungsstätte in der DDR).

5.8 Gründungsdokument der Ingenieurhochschule für Elektronik und Datenverarbeitung Dresden vom 10.3.1969

Am 10.3.1969 legte die Ingenieurschule für Maschinenbau und Elektrotechnik Dresden das „**Gründungsdokument der Ingenieurhochschule für Elektronik und Datenverarbeitung Dresden**“ zur Dienstbesprechung beim Minister für Hoch- und Fachschulwesen vor.[28]

Für die zwei Fachstudienrichtungen „Systemtechnik der Datenverarbeitung“ und „Informationselektronik“ waren die Vertiefungsrichtungen Einsatzvorbereitung, Fertigung, Programmierung, Mess- und Prüftechnologie, Prozesssteuerung, Prüfung und Wartung[29] vorgesehen. Über die Forschung hatte man nur vage Vorstellungen und sah Schwerpunkte bei Gestaltung technologischer Prozesse und Verfahren, Einsatzvorbereitung und Einsatz der EDV bei technischen, ökonomischen und gesellschaftlichen Systemen mit Informationsaustausch und –verarbeitung Weiterentwicklung der Lehre[30] Zur Sicherung praxisnaher Ausbildung verwies man auf vertragliche Verbindungen mit zehn Betrieben und Einrichtungen, darunter der TUD.[31] Nach Verweis auf das bereits bestehende postgraduale Studium werden künftige Vorhaben zur Weiterbildung, wie Frauensonderstudium, Teil- und Zusatzstudium oder Intensivlehrgänge für Industriekader angekündigt. Ferner wird eingeschätzt, dass auf die Fachschulausbildung im Raum Dresden nicht verzichtet werden kann. Deshalb wird die Weiterführung des Fachschulstudiums für die Ausbildungsrichtungen „Elektronische Datenverarbeitung“, „Technologie des Maschinenbaus“, „Apparate- und Gerätebau“ dringend angeraten.[32]

[27] ISME 7 – 026 „MMM 1964 – 1972“
[28] IHD 4300 „Grundsatzdokumente der Einrichtung 1969 – 1986“
[29] ISME 7 – 035 „Umbildung zur Ingenieurhochschule 1969“
[30] ISME 7 – 035 „Umbildung zur Ingenieurhochschule 1969“
[31] ISME 7 – 035 „Umbildung zur Ingenieurhochschule 1969“
[32] ISME 7 – 035 „Umbildung zur Ingenieurhochschule 1969“

5.9 Der Beginn der Hochschulausbildung

Zur Vorbereitung auf die Gründung einer Ingenieurhochschule am 29.4.1969 wurden schon zu Beginn des Studienjahres 1968/69 an der Ingenieurschule Studenten immatrikuliert, die als **Hochschulsystemingenieure FR Datenverarbeitung** (HS) bzw. **Hochschulingenieure für Rechenelektronik** (HE) nach 3 ½ jähriger Ausbildung, von denen das letzte Semester ein Betriebspraktikum war, ihr Studium absolvierten. Die betreffenden Studenten hatten die Zulassung für die

FR Programmierung	)
FR Ingenieurökonomie	)—FR HS
FR Technologie des Maschinenbaus	)
FR Technologie der Elektrofeinwerktechnik	)
FR Elektronische Datenverarbeitungsanlagen	)—FR HE
FR Technologie des Maschinenbaus.	)

Aus der Überlieferung geht hervor, dass einige Studenten dieser Seminargruppen die Chance wahrnahmen, ihre Hochschulabschlussarbeit zur Diplomarbeit auszubauen, und sich später an der TU Dresden um ein Diplomverfahren bemühten. Bei anderen wurde die Übernahme ins Forschungsstudium an der dann schon längst arbeitenden Ingenieurhochschule Dresden angestrebt.

6 Ingenieurhochschule Dresden 29.4.1969 – 3.10.1986 (IHD) und Informatikzentrum des Hochschulwesens der DDR an der TUD 4.10.1986 – 17.4.1991 (IZ)

Aus Platzgründen wird auf inhaltsreiche Darstellung beider Einrichtungen verzichtet. Trotzdem sollen hier die wichtigsten Daten genannt werden:

Die **Gründung der IHD** erfolgte im Zusammenhang mit der sog. III. Hochschulreform in der DDR **am 29.4.1969**. Unter der Bezeichnung **Ingenieurhochschule für Elektronik und Datenverarbeitung Dresden**[33] ging die Entwicklung der traditionsreichen Einrichtung in Dresden-Johannstadt weiter und fand dort erst mit Umzug der Fakultät Informatik in den Neubau an der Nöthnitzer Straße ein Ende.[34] Mit Gründungsdatum der Hochschule wurden an der IHD die Sektionen „**Systemtechnik der Datenverarbeitung**“ (Sektion 11) und „**Informationselektronik**“ (Sektion 12) gebildet[35]. An der IHD arbeitete ab 1970 ein eigenes **Rechenzentrum**, welches vor allem der Ausbildung und Forschung der Hochschule diente, aber auch Dienstleistungen im Verwaltungsbereich (z.B. Studentenverwaltung, Gehaltsabrechnung) erbrachte.

---

[33] Rektor wurde Prof. Eberhard Buzmann.
[34] Der Straßenname änderte sich Ende 1971 von "Elisenstraße 25" in "Hans-Grundig-Straße25"
[35] Sektionsdirektoren wurden Dr. Hans-Joachim Grüneberger (11) und Dr. Manfred Tierock (12).

Mit Urkunde des Ministers für Hoch- und Fachschulwesen wurde dem Wissenschaftlichen Rat der IHD das **Recht zur Verleihung des akademischen Grades Diplom-Ingenieur** per 2.10.1972 erteilt.[36] Vom 26. – 28.11.1975 fand in Dresden eine Tagung „Entwicklung und Anwendung der elektronischen Rechentechnik in der DDR", die sog. „**EDV-Anwender-Konferenz**" statt. Die IHD beantragte im Oktober 1979 das **Recht zur Verleihung des akademischen Grades Doktor-Ingenieur**, welches der Wissenschaftliche Rat der IHD per 1.2.1980 zugesprochen bekam. Im Zusammenhang damit wird als neues Vorhaben der **Aufbau des Technikums Medizintechnik, Kleinrechentechnik und Mikroakustik** ausgewiesen. Die Realisierung einer Schülerakademie, die heute als Schülerrechenzentrum immer noch weiterlebt, wird ebenfalls erwähnt.[37]

Nach Beschlüssen des Ministerrates der DDR und der Parteiführung stand es 1985 fest, dass die Zukunft der Ingenieurhochschule in einem gemeinsam mit der Sektion 08[38] der TUD neu zu gründenden Informatikzentrum liegt. Die **Gründung** des IZ wurde **am 4.10.1986 in einem Festakt vollzogen**. Der tatsächliche Aufbau des IZ, also die Zusammenführung der Potentiale der TUD und der IHD erfolgte schrittweise bis 31.8.1987. Erst danach, also ab 1.9.1987 tritt die TUD in die Rechtsträgerschaft ein.[39]

Quellennachweis

Autorenkollektiv der Ingenieurhochschule Dresden: Von der Gewerbeschule Dresden zur sozialistischen Hochschule 1861 – 1986

TU Dresden, Fakultät Informatik, Institut Datenbanken / Künstliche Intelligenz, Doz. Dr.-Ing. habil. Eberhard Klett, Lehrheft 1 und 2 für die Vorlesung „Geschichte der Informatik" im WS 1992/1993

UA „Urkundensammlung Sektion 08"

Akten aus Universitätsarchiv, Archivteil IHD:

alle Studentenakten

GS 1 – 019 „Hans-Benisch-Stiftung 1910 – 1914"
GS 1 – 025 „Abschlusszensuren Ergänzungskurse der Technischen Mittelschule 9/1922"
GS 1 – N – 1 Nachlass Karl Wilhelm Clauß
GS 1 – S – 1 Jahresberichte der Städtischen Gewerbeschule Dresden

TLA 2 – S – 2 Jahresberichte der Technischen Lehranstalten Dresden

[36] IHD 4300 „Grundsatzdokumente der Einrichtung 1969 – 1986"
[37] IHD 4302 „Promotionsrecht 1979 – 1980"
[38] Die Sektion 08 Informationsverarbeitung entstand am 1.1.1969 an der TUD. siehe Gründungsdokument in S. 08 5082 „Urkunden 1969 – 1982"
[39] IHD 5071 „Bildung des Informatikzentrums 1985 – 1986" – In dieser Akte befindet sich der „Führungsplan des MHF zur Verwirklichung des Beschlusses des Ministerrates vom 21.8.1986 über die Vereinigung der TUD und der IHD und die Bildung des Informatikzentrums des Hochschulwesens an der TUD" vom 14./15.7.1986, welcher als Gründungsdokument anzusehen ist. Direktor wurde Prof. Horst Tzschoppe.

ISME 7 – 012 „AG Elektronik, Schwachstromtechnik, Datenverarbeitung und Automatisierungstechnik beim SHF 1964 – 1965“
ISME 7 – 013 „Arbeit der Rationalisierungskommission 1966 – 1967“
ISME 7 – 014 „Industriekonferenz am 14.2.1964“
ISME 7 – 025 „Studentenwettstreit 1963 – 1964“
ISME 7 – 026 „MMM 1964 – 1972“
ISME 7 – 035 „Umbildung zur Ingenieurhochschule 1969“
ISME 7 – 171 „Ablage der FR EDVA 1962 – 1969“
ISME 7 – N – 1/VIII.2 „Zeitungsausschnittsammlung 1964 – 1975“
ISME 7 – S – 08 „Sammlung Zeitungsausschnitte 1967 – 1968“

IHD 3142/1 und IHD 3142/2 „Statistiken zur Geschichte der Einrichtung“
IHD 3152 „Zentrale Veranstaltungen der IHD 1969 – 1974“
IHD 3284 „EDV-Anwender-Konferenz 1975“
IHD 3970 „Gründungsurkunde der IHD 1969“
IHD 4097 „Verleihung des Promotionsrechts Dr.-Ing. an die IHD 1980“
IHD 4098 „Ordnungen und Weisungen 1969 – 1981“
IHD 4300 „Grundsatzdokumente der Einrichtung 1969 – 1986“
IHD 4302 „Promotionsrecht 1979 – 1980“
IHD 5064 „Rechentechnik am IZ 1986 – 1990“
IHD 5071 „Bildung des Informatikzentrums 1985 – 1986“
IHD 5075 „Urkundensammlung zur Struktur der Einrichtung“

S. 08 5082 „Urkunden 1969 – 1982“

IZ 4567 „Forschung am IZ 1986 – 1987“

Zur Geschichte und Lehre der Informatik an der Technischen Universität Dresden bis zur Gründung des Informatik-Zentrums 1986

Erwin Schmidt

Wettinstr. 3
01445 Radebeul
erwin.schmidt@tu-dresden.de

Abstract: Nach kurzer Einführung zur Entstehung der Sektion 08 Informationsverarbeitung der TU Dresden im Zeitraum 1950 bis 1968 werden insbesondere Fakten zum Direktstudium bis zur Gründung des Informatikzentrums im Oktober 1986 zusammengestellt und damit zusammenhängende Themen wie Personalentwicklung, andere Studienformen und Gerätetechnik für die Ausbildung behandelt.

1 Vorbemerkung

Der Autor studierte Elektrotechnik an der Technischen Hochschule/Universität Dresden und wechselte nach Ende seiner Assistentenzeit im 1. Halbjahr 1969 als wissenschaftlicher Assistent und Oberassistent an die neu gegründete Sektion 08 „Informationsverarbeitung“. Einige Fakten zur Geschichte der Sektion und zur universitären Ausbildung sollen im folgenden den Gedächtnisbericht von Prof. Dr.-Ing. habil. Ottomar Herrlich unterstreichen, den er auf dem 2. Symposium „Informatik in der DDR“ 2004 in Chemnitz gehalten hat [He04]. Trotz intensiven Quellenstudiums kann dies allerdings keine abschließende und vollständige Aufbereitung sein, weitere Hinweise und Ergänzungen von Zeitzeugen sind gern gesehen.

2 Der Zeitraum bis 1968

Mit der weltweiten und stürmischen Entwicklung der EDV entstand auch in der DDR in den 1960-er Jahren ein Bedarf an Informatik-Spezialisten, die den Einsatz und die Anwendung der Rechentechnik ingenieurmäßig realisieren konnten, wobei die Ziele der Ausbildung von Fachleuten für die Produktion, Anwendung und Bedienung von EDV-Anlagen nach heutiger Sicht zu hoch gesteckten waren [Pi04]. An mehreren Hochschulen und Universitäten wurden die ersten Rechner angeschafft und Weiterbildungs- und Spezialisierungskurse angeboten, meist in den Mathematik-Instituten konzentriert [NS06]. An der TU Dresden hatte sich Prof. Dr. rer. nat. habil. Nikolaus Joachim Lehmann an seinem Institut für Maschinelle Rechentechnik seit den 1950-er Jahren große Verdienste erworben, Computer entwickelt und 15 Diplomandenplätze geschaffen [Pe04; Le78]. In Vorbereitung der 3. Hochschulreform arbeitete von Oktober 1966 bis

1968 eine „Arbeitsgemeinschaft Datenverarbeitung“ des Rektors unter Leitung von Prof. Dr.-Ing. habil. Dr. E.h. Hans Frühauf, ab März 1967 aus gesundheitlichen Gründen von Prof. Dr.-Ing. habil. Fritz Wiegmann abgelöst [Pr66]. Mitglieder waren neben Prof. Dr. Lehmann mehrere Professoren der Fakultät für Elektrotechnik und Industrievertreter. Die drei Hauptaufgaben bestanden in der breiten Einführung der Rechentechnik an der TU, der Vorbereitung eines selbständigen Universitäts-Rechenzentrums und der Erarbeitung von Grundlagen der Informatikausbildung [Pu03; Pr66]. Beispielsweise wurde im Dezember 1967 ein erster Entwurf zu einem „Stufenplan für die Ausbildung im Fachgebiet Informationsverarbeitung und –übertragung“ (in 3 Stufen) für alle TU-Studenten erarbeitet [Pr66], damit sie Probleme des Fachgebietes in einer Programmiersprache formulieren und mit Informationsverarbeitungsanlagen lösen können.
In dem angeforderten Bericht an das Staatssekretariat für das Hoch- und Fachschulwesen vom Juni 1968 über EDV und Kybernetik in den Grundstudienplänen ist aufgeteilt nach Fakultäten ausgeführt, daß 2 oder mehr Wochenstunden EDV und 2 oder mehr Wochenstunden Kybernetik neu ab Jahrgang 1968 in allen Ingenieursdisziplinen gehalten werden sollen. Zu diesem Zweck wird ein 3-Stufen-Plan aufgestellt. Seine Umsetzung nach der im Oktober stattgefundenen 3. Hochschulreform ist unklar.
Mehrere Beschlüsse [Pr66] bereiteten die Gründung des Universitätsrechenzentrums als selbständige Struktureinheit am 1.12.1968 vor [Po03, S.307].
Bezüglich der Spezialausbildung auf dem Gebiet der Informatik konnte sich Prof. Dr. Lehmann nicht durchsetzen, diese in der Sektion Mathematik anzusiedeln. Es wurde im Zuge der 3. Hochschulreform eine eigene Sektion 08 gebildet, die gemeinsam mit den Sektionen 09 „Informationstechnik“, 10 „Elektroniktechnologie und Feingerätetechnik“ (stark erweitert gegenüber der früheren Studienrichtung „Elektrischer Feingerätebau“ [Kr00]) und 11 „Elektrotechnik“ einer „Fakultät für Datenverarbeitung“ als Nachfolger der Fakultät für Elektrotechnik zugeordnet war. Die Fakultätsbezeichnungen sind im Gegensatz zu falschen Darstellungen in manchen anderen Quellen auch im Beitrag von Prof. Dr. Lunze 1993 [Lu93] richtig angegeben. Das Universitätsverzeichnis 1968/1969 nennt: „8 (Sektion Kybernetik und Datenverarbeitung – Arbeitstitel: Systemingenieurwesen) Gründung in Vorbereitung“ [TU69]. Die Fakultäten blieben nur für gewisse Aufgaben (Promotionsgeschehen, Vergabe der Facultas docendi, Forschungsthemen, Wahl des Rektors u.ä.) bestehen [Re68] und waren darin für die zugeordneten Sektionen zuständig. Durch diese Entmachtung fiel den Elektrotechnikern die Umbenennung und Erweiterung der Fakultät als Zugeständnis an die politischen Forderungen wohl nicht ganz so schwer [Pu03].
Dem unter Leitung von Prof. Dr. Stuchlik (TH Magdeburg) erarbeiteten 4-Stufen-Programm [Mi68] folgte schon 1969 eine überarbeitete Fassung [St69], in der die Stufe 4 die „Mindestanforderungen für die Grundlagenausbildung für Spezialkader zur Anwendung der EDV“ festschrieb. Die Absolventen dieser mit 360 Stunden veranschlagten Ausbildung sollten Algorithmen aufstellen, sie in problemorientierter und maschinenorientierter Programmiersprache auf EDVA nutzen können und Kenntnisse über Struktur und Realisierung von Programmsystemen besitzen.

3 Das Gründungsjahr 1969

Zum 1. 1. 1969 erfolgte die Gründung der Sektion 08 unter der Bezeichnung „Informationsverarbeitung“ an der TU Dresden. Die Hochkonjunktur der Kybernetik in der DDR war vorüber und die Bezeichnung „Informatik“ besonders im befreundeten Ausland (UdSSR) übergreifend im Bibliothekswesen verankert [He04; Fu04; Pe04]. So plädierte auch Prof. Dr. Lehmann gegen diese Bezeichnungen [Pe04]. Der endgültige Sektionsname ging evtl. auf die Bezeichnung der IFIP (International Federation of Information Processing) zurück – ohne dass man dies zugeben konnte.
Ebenfalls im Januar 1969 konstituierte sich die Fakultät für Datenverarbeitung, als Dekan wurde Prof. Dr. Wiegmann gewählt [Ei69].
Noch im September 1969 ist die Notwendigkeit der neuen Studienrichtung entgegen dem Bestreben, die Anzahl der Grundstudienrichtungen zu verringern, betont worden [Re69]. Da das Studium in den Ingenieurwissenschaften 1969 auf 4 Jahre verkürzt worden war [Kr00], 8.1) begann auch der neue Studiengang Informationsverarbeitung mit einer Regelstudienzeit von 8 Semestern und sollte mit dem akademischen Grad „Dipl.-Ing. für Systeme der Informationsverarbeitung“ [Re69] abschließen, später vereinfacht zu „Dipl.-Ing. für Informationsverarbeitung“.
Zum Gründungsdirektor der Sektion 08 wurde Prof. Dr.-Ing. Heinz Stahn [He06] berufen, der auf dem Gebiet des Rechnereinsatzes zur Produktionssteuerung in der chemischen Industrie gearbeitet hatte [BBH04]. Da die TU nicht allein die benötigte Anzahl von Informatikern ausbilden konnte, erfolgte auch in Dresden fast zeitgleich am 29.04.1969 die Gründung der Ingenieurhochschule Dresden [Ge78] mit einer starken Sektion Informationsverarbeitung und dem Ziel, im Herbstsemester 1969 an beiden die Ausbildung zu beginnen. Im April erhielt die Sektion 08 ihren Sitz in der Außenstelle Dürerstraße der TU in Nachbarschaft zur Ingenieurhochschule. Da war wohl an eine engere Zusammenarbeit gedacht, aber erst gute 5 Jahre später (September 1974) ist mir der Entwurf einer konkreten Vereinbarung dazu bekannt [Se70]. Viele Probleme sind wohl auch dadurch entstanden, dass an der TU den Fragen der Etikette nicht solche Bedeutung beigemessen worden ist wie an der Ingenieurhochschule.
An der Sektion 08 mussten zuerst Mitarbeiter aus der Industrie und vorrangig aus der TU selbst gewonnen werden. Neben Prof. Dr. Stahn waren dies anfangs Dr. rer. nat. habil. Klaus Voß als Hochschuldozent und Stellvertreter für Erziehung und Ausbildung, Dr.-Ing. habil. Ottomar Herrlich und Dr.-Ing. Hans Loeper als Oberassistenten, ein R300-Spezialist und eine Sekretärin. Die Absicht, Dr. Kerner zur Stärkung der mathematischen Richtung an die Sektion 08 zu berufen, scheiterte an der Verweigerung der Freigabe durch den Rektor der Universität Rostock [He04]. Das Universitätsverzeichnis 1970 nennt 20 Mitarbeiter der Sektion [TU70] für das Studienjahr 1969/70. Erster Honorarprofessor ist Dr. rer. nat. Karl-Heinz Müller.
Zum 1. 9. 1969 wurden 157 Studenten immatrikuliert [Ho69] (die Zahlen schwanken in anderen Quellen infolge von Nachimmatrikulationen zwischen 152 [Ho70] und 159 [Se71] (25.9.1070)). Diese waren von anderen Studienwünschen „umgelenkt“ worden, denn die Bewerbungen für eine Fachrichtung hätten schon ein Jahr vorher erfolgen müssen. [He04]. Mit dem geringen Personalbestand und den nicht unbedingt den Interessen aller Studenten entsprechenden Umlenkungen stellte dies doch ein gewaltiges Wagnis dar und war nur zu bewältigen, weil das Grundstudium überwiegend von ande-

ren Sektionen bestritten wurde und viel Wert auf eine sofortige Weiterbildung aller neu eingestellten Mitarbeiter gelegt wurde.
Im Grundstudium waren in den ersten drei Semestern je 32, im 4. Semester 28 Wochenstunden geplant, die sich wie folgt verteilten [Re69] (Wochenstunden Vorlesung/Übung/über Semester): Mathematik: 22/16/4, Physik: 8/8/4, Elektrotechnik und Gerätesysteme durch Sektion 09: 6/2/2, Informationsverarbeitung, Kybernetik, Systemtheorie und Programmierungstechnik durch Sektion 08: 13/25/4 und der Rest für M/L (Marxismus-Leninismus, Politische Ökonomie), Sprachen und Sport. Zusätzlich standen Praktika und Intensivlehrveranstaltungen im Plan.
Die Studenten waren in Seminargruppen organisiert, die von einem Assistenten als Seminargruppenberater angeleitet und „sozialistisch erzogen" werden sollten. So diente das Gruppenseminar im 1. Semester zum gegenseitigen Kennenlernen, der Einführung in das Studium an der Universität, der Organisation gegenseitiger Hilfe im Studium und der Durchsetzung administrativer Maßnahmen wie z.B. Teilnahme an Einsätzen in Wohnheim, Landwirtschaft und Industrie. Alle Benotungen waren im Gegensatz zur heutigen Zeit öffentlich und gestatteten so die Durchführung von „Leistungsgesprächen" und die oft rechtzeitige Organisation von Patenschaften über leistungsschwache Studenten und von „Lernzirkeln". Die Seminargruppenberater sind ihrerseits durch einen Hochschullehrer als Jahrgangsleiter angeleitet worden. Auf dem Zeitzeugenkolloquium der Elektrotechnik stellte Prof. Dr. Krause fest [Kr00], dass auch heutige Studenten in den ersten Semestern ein ähnliches Gruppenstudium mit Tutoren zur Erleichterung des Einstiegs als wünschenswert erachten. Mit einer Hauptprüfung am Ende des Studiums im 4. Studienjahr vor der Diplomarbeit wurde der Abschluss „Hochschulingenieur" erworben. Ein Abgang von der Universität war damit möglich, wurde aber praktisch kaum genutzt. Der Studienabschluss als Hochschulingenieur bei Weiterführung eines Forschungsstudiums ohne Diplomabschluss war zwar von der TU-Leitung vehement gefordert worden, um die Gesamtstudiendauer für promovierte Kader zu verkürzen, aber von der Sektionsleitung wurde dieser Weg nicht favorisiert.
Während des gesamten Wirkungszeitraums der Sektion musste in Hinsicht auf personelle, räumliche und technische Ausstattung der TU-Leitung und dem TU-Rechenzentrum unter Leitung von Prof. Dr. Stahl jedes Zugeständnis abgerungen werden, z.B. hatte Prof. Stahl bei der „dezentralen Aufstellung" der Kleinrechner-Technik klare Einsatzstudien gefordert, die vom TU-Rechenzentrum zu genehmigen wären [Re75].

4 Die Jahre 1970 – 1974

4.1 Direktstudium

In der Realisierung des Studienplanes wurden einige Änderungen erforderlich. So hielten die Lehrveranstaltung Kybernetik I + II die Hochschuldozenten Dr. Klaus Voß und dann Dr. Hardwin Jungclaussen von der Sektion 08, Graphentheorie entfiel und Physik wurde auf 2 Semester gekürzt, da diese Sektion keine spezielle Ausrichtung auf Modellbildungsprozesse realisieren konnte [He04; Fr71].
Im Jahrgang 1971 [Fr71] wurden statt der Systemtheorie wegen der Beurlaubung von Dr. Voß (siehe 4.3) im (2. Semester) die Lehrveranstaltung Kybernetik II (Jungclaussen) und zwei Lehrveranstaltungen zu Automaten und Informationstheorie durch Prof.

Dr. Stahn (08) und Dr. Riedel (09) gehalten. In den Semestern 3 und 4 bestritt die Sektion 09 (Dr. Entreß) die Lehrveranstaltung „Gerätetechnik der digitalen Informationsverarbeitung". Nach Ende des 1. Studienjahres erfolgte ein Einsatz im Rahmen von Studentenbrigaden in der Industrie, z.B. im Kernkraftwerk Greifswald, während des 4. Semesters im April 1973 eine militärische bzw. Zivilverteidigungs-Ausbildung.
Das Fachstudium vom Herbstsemester 1973 bis 1975 beinhaltete 2 Semester Mathematik/Optimierung (Prof. Dr. Terno), „Automaten- und Algorithmentheorie" (Dr. Seidel, Mathematik) und 3 Semester Sozialistische Betriebswirtschaftslehre als Fremdleistungen anderer Sektionen. Im 5. Semester gab es dazu die Lehrveranstaltungen „Betriebssysteme" (Werner), „Organisation von Datenbanken" (K.-H. Müller) und „Informationsgewinnung und –nutzung" (Stahn), im 6. Semester „Formale Sprachen und Compiler" (Loeper), „Steuerungs- und Regelungstechnik" (Giese) und am Ende das Ingenieurpraktikum (Berufspraktikum). Dieses wurde in Betrieben und im Ausnahmefall auch an der TU durchgeführt und umfasste neben dem eigentlichen fachlichen Thema auch ein M/L-Thema. Im 4. Studienjahr gab es die Lehrveranstaltungen „Analyse und Projektierung von automatisierten Systemen der Informationsverarbeitung" (Schubert), „Algorithmische Sprachen / ALGOL 68" (Stiller), „Informationsübertragung" (Löffler), „Prozeß- und Fertigungssteuerung" (Stahn) und „Automatisierte Systeme der Informationsverarbeitung in Leitung und Planung" (Pflug). Vor der Hauptprüfung musste ein Großer Beleg angefertigt werden. Weiterhin fand im 8. Semester die Hauptprüfung statt und danach die Diplomarbeit von knapp 4 Monaten Dauer [Fa69].
Gemäß der im März 1972 überarbeiteten „Ausbildungskonzeption der Grund- und Fachstudienrichtung Informationsverarbeitung (Kurzfassung)" (in [Re72] am 19.6.1972 als Anlage) wurde lt. konkreter Stundentafel Physik wieder in drei Semestern gelehrt. Insgesamt umfasste die Ausbildung in mathematisch-naturwissenschaftlichen Grundlagen (Mathematik und Physik) 23% und in der fachspezifischen Ausbildung mit Programmierungstechnik, Grundlagen und Anwendung der EDV, Systemtheorie/Kybernetik, Forschungsseminar und einer Nebenfachausbildung 52% des Zeitfonds. Das Nebenfachstudium vom 6. bis zum 8. Semester ermöglichte in ausgewählten Wissensgebieten – vorzugsweise in der Ökonomie, dem Maschinenbau, der Energietechnik und der Informationstechnik – den Erwerb von Spezialkenntnissen.
Als fruchtbar erwiesen sich Forschungsseminare im 4. Studienjahr. Unter Leitung von Hochschullehrern stellten die Studenten ihre Konzeptionen und Ausarbeitungen des Großen Belegs vor und Forschungsstudenten informierten über den Inhalt ihrer Dissertationsarbeiten, so dass stets ein guter Kontakt zwischen Studenten und Hochschullehrern gewährleistet war.
Am 1.2.1972 studierten in den Immatrikulationsjahrgängen 1969 – 1971 ca. 530 Studenten an der Sektion 08 [Re72], darunter auch einige Ausländer. Die Immatrikulationszahlen beliefen sich laut [Wi78], Statistik S.31, in den Jahren 1970 bis 1974 auf 175, 204, 168, 101 und 85 DDR-Studenten. Das Sinken der Immatrikulationszahlen ab 1972 ist eine Folge der weiter unten dargelegten Beschlüsse des Wissenschaftlichen Rates. Das Diagramm Abbildung 1 wurde aus Angaben mehrerer Quellen ermittelt. Die Studentenzahl an der Sektion 08 sollte bis Anfang der 1980-er Jahre noch bis auf unter 300 sinken ([Re81] und Abbildung 2), bevor dann wieder höhere Zulassungszahlen gestattet waren.

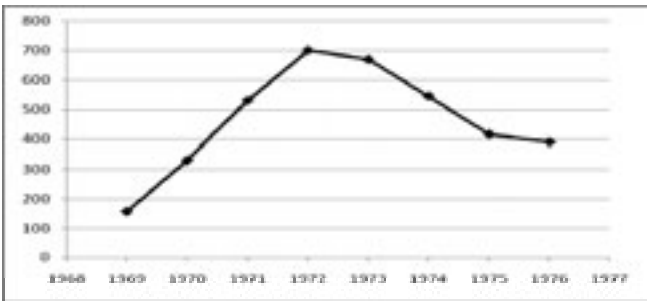


Abbildung 1: Anzahl der Direktstudenten an der Sektion 08

Im Jahre 1973 schlossen die ersten 130 Studenten des Jahrgangs 1969 ihr Studium als Diplomingenieur für Informationsverarbeitung ab [Fa69] (21. 2. 1974). Sie erreichten in den Hauptprüfungen 1973 nur zu 39,6% die Noten „gut“ und „sehr gut“ – wohl auch eine Folge der „Umlenkungen“. Demgegenüber verbesserte sich der Jahrgang 1970 darin auf 55,6% [Re75]. Im Diplomverfahren und damit auch im Gesamtprädikat wurden aber in allen Jahrgängen bessere Noten erreicht.

4.2 Fernstudium

Im Fernstudium sind 1969, und 1970 je etwa 100 Studenten immatrikuliert worden, per Stand vom 2.12.1971 studierten sie an der Sektion 08 in 17 Seminargruppen. 1972 wurden noch einmal 123 Fernstudenten immatrikuliert [Wi78] (Statistik S. 32). Die Fernstudenten erhielten zum Selbststudium Lehrbriefe, die oft von Mitarbeitern der Sektion ausgearbeitet und/oder überarbeitet worden sind. Zu mehreren einwöchigen Kursen pro Semester an der Universität wurden sie von ihren Betrieben freigestellt, ebenso 4 Wochen für die Abschlussarbeit.

4.3 Personelle Entwicklung und Auswirkungen auf die Lehre

In diesem Zeitraum ist der Personalbestand ständig gewachsen. 1974 lehrten an der Sektion 59 Personen wissenschaftliches Personal, davon 13 Hochschullehrer [TU74]. Einige Honorarprofessoren (Prof. Dr. Frank Baldeweg, Prof. Dr. Gerhard Merkel, Prof. Dr. Karl-Heinz Müller, Prof. Dr. Dietrich Schubert), Dr. Dieter Müller als Honorardozent, und Dr. Bonitz und andere, auch noch nicht Promovierte, als Lehrbeauftragte unterstützten in der Lehre.
Mit der steigenden Mitarbeiterzahl wurde die Sektion zuerst in die Wissenschaftsbereiche Programmiersysteme (Dr. Herrlich) und Automatisierte Systeme der Informationsverarbeitung (Dr. Pflug) [TU72], dann in die drei Wissenschaftsbereiche Programmierungstechnik (Prof. Dr. Herrlich), Informationsverarbeitungstechnik (Dr. Werner) und Automatisierte Systeme der Informationsverarbeitung (Dr. Pflug) [TU74] gegliedert.
Eine erste Krise ereilte die Sektion im Jahre 1971 (siehe auch [He04]). Der Versuch, die Formalisierung von materiellen, energetischen und informationellen Systemen auch auf ideologische, gesellschaftliche Systeme zu übertragen, ohne Klassenkampf und Erfahrungen der Parteikader zu berücksichtigen [Se69] (8.4.1971) ; [Vo69], führte zu harschen Reaktionen des Chefideologen Kurt Hager. Der Kybernetiker Dr. Voß wurde

gegen Ende 1971 von seiner Funktion als Stellvertreter für Erziehung und Ausbildung entbunden und als Hochschullehrer für drei Jahre in die Praxis delegiert [Se69] (2. 12. 1971). Die von ihm konzipierte 4-semestrige Vorlesung zu Kybernetik [He04] konnte daher so nicht realisiert werden. Gewisse persönliche familiäre Probleme haben ihm dabei diesen Schritt etwas erleichtert. Nach Ende der Delegierungszeit musste er nicht wieder an die Sektion zurückkehren [Fa75] (22. 9. 1975), er nahm einen Ruf als Professor an die Friedrich-Schiller-Universität Jena an. Prof. Dr. Stahn war aber aus der Praxis berufen worden. So wurde er vom Politbüromitglied Kurt Hager – wohl auf dem Parteiwege durchgestellt – ernst gerügt und musste wohl zusagen, dieses Thema nicht wieder anzusprechen und sich von Dr. Voß loszusagen. Diese harte Linie war evtl. eine Folge der Veränderungen im Übergang von Ulbricht zu Honnecker, wie es Seising [Se06] beschrieben hat.
Eine zweite Krise folgte gleich 1972. Auf der 24. Sitzung des Senats des Wissenschaftlichen Rates am 19.6.1972 wurde eine „Studie zur Ausbildung auf dem Gebiet der Informationsverarbeitung“ behandelt [Re72], die unter Leitung des Prorektors für Wissenschaftsentwicklung aus der Sektion Mathematik entstanden war. Es sollte geprüft werden, ob eine eigene Sektion richtig sei oder die Aufgaben in die Mathematik, evtl. auch teilweise in die Informationstechnik, das Rechenzentrum und andere Fachsektionen verteilt werden müssten. Die mathematischen Grundlagen würden fehlen – obwohl in der Anlage 6 Semester Mathematikausbildung mit insgesamt 29 Wochenstunden Vorlesung und 16 Wochenstunden Übungen plus Praktika und Beleg ausgewiesen waren!? Der Direktor des TU-Rechenzentrums informierte demgegenüber über die Aussage einer Arbeitsgruppenberatung des RGW, daß die DDR auf diesem Gebiet einen ansehnlichen Vorlauf hätte.
Mit den ab Herbstsemester 1972 geplanten 200 Neuimmatrikulationen kämen dann an der Sektion 08 etwa 70 Studenten auf einen Hochschullehrer gegenüber einem TU-Durchschnitt von 28,55 [Wi78] (S. 35). Daher wurde eine deutliche Reduzierung der Studentenzahlen im Direktstudium und statt neuer Fernstudien-Matrikel die Aufnahme eines postgradualen Studiums festgelegt. Der Zeitpunkt für eine endgültige Entscheidung sei aber noch nicht reif, zumal erst 1973 die ersten Absolventen die Sektion verlassen werden. Schließlich sicherte der Rektor dem Direktor der Sektion 08 seine weitere Unterstützung zu.

4.4 Weiterbildung und Promotionsgeschehen

Zu Forschungsergebnissen in Theorie- und Applikationsfragen wurden verschiedene Problemseminare mit Vertretern anderer Hochschulen/Universitäten und Praxiskadern durchgeführt.
Mit Stand vom 1.5.1972 arbeiteten an der Sektion 23 Forschungsstudenten, 8 wissenschaftliche Assistenten und 24 Aspiranten an Dissertationsthemen [Se71]. Im Forschungsstudium sollten befähigte Absolventen in kürzerer Zeit zur Promotion geführt werden. Zu diesem Zwecke waren sie gegenüber Wissenschaftlichen Assistenten weitreichend von Lehrverpflichtungen entlastet.
Im Jahre 1973 promovierten dann die ersten 7 Forschungsstudenten der Sektion an der Fakultät für Datenverarbeitung zum Dr.-Ing. [Fa69].

5 Die Jahre 1975 – 1986

5.1 Direktstudium

Mit dem Jahrgang 1975 wurde das 4 1/2-jährige Studium eingeführt. Das Ingenieurpraktikum (Berufspraktikum) sollte einheitlich im 7. Semester liegen, die Verlängerung des Studiums insbesondere für

- eine Verlängerung der Diplomphase (im 9. Semester),
- eine Verlängerung des Berufspraktikums auf 20 Wochen und
- zusätzliche praxisbezogene Phasen intensiver wissenschaftlicher Arbeit

genutzt werden [Re75]. Im Studienplan waren jetzt 22 Praktika mit insgesamt 29 Wochenstunden vorgesehen [Se72].

Nach guten Erfolgsquoten der Jahrgänge 1970 und 1971 mit über 90% prognostizierte man aber [Pr76], dass für die Jahrgänge 1972 und 1973 sehr viel geringere Quoten zu erwarten seien „unter dem Aspekt, dass der Anteil der Mädchen sprunghaft gestiegen" sei (50-60%) – und dies bei der Frauenförderung in der DDR! Sehr gute Studienleistungen wurden mit besonderen Stipendien belohnt. Am Jahresende 1975 gab es an der Sektion bei insgesamt etwas über 400 Studenten 144 Sonder- und Leistungsstipendien.

Die Statistik über die Immatrikulationszahlen von DDR-Studenten [Wi78], die nach dem Senkungsbeschluss von 1972 im Jahrgang 1975 immer noch bei 100 lagen, weist dann einen drastischen Rückgang auf ein Minimum von nur 40 im Jahrgang 1979 auf, was aber auf die Einführung eines einjährigen Vorpraktikums für den Jahrgang 1980 zurückzuführen war. Deshalb wurden 85 Immatrikulationen im Jahrgang 1980 geplant [Se69], dann folgten wieder Steigerungen auf 100 im Jahr 1982, 120 im Jahr 1983 und je 140 in den Jahren 1984 und 1985 [Se72]. Auch erfolgte schon vor der Gründung des Informatikzentrums für die Jahrgänge 1984 und 1985 eine Erhöhung der jährlichen Immatrikulationszahlen beider Sektionen Informationsverarbeitung der TU und der Ingenieurhochschule per Beschluss des Ministerrates [Be84] auf 280 DDR-Studenten (ohne Ausländer) [Re82], eine Steigerung auf mehr als das Doppelte gegenüber dem Jahrgang 1981.

Zum Ausländerstudium sind Zahlen für 1976-77 bekannt, von geplanten 25 sind 23 immatrikuliert [Re77]. Im Jahrgang 1977 wurden 12 Ausländer immatrikuliert [Se72], das ist immerhin ein Anteil von 18%, im allgemeinen lag er bei knapp 10%.



Abbildung 2: Anzahl der DDR-Direktstudenten an der Sektion 08

Nachgewiesen sind in Abbildung 2 die Werte von 1976, 1984 und 1985, die anderen extrapoliert aus diesen und Immatrikulations- und angenäherten Absolventenzahlen.
Neben Grundlagen- und Vertragsforschung gab es noch die „Forschung für die Lehre". Oft geschah das im Rahmen von „Jugendobjekten" mit wesentlichem Arbeitsanteil der Studenten. Themen waren beispielsweise „Testmonitor", „Dialogsystemgrundlagen der Programmierungstechnik", „GDL-Programmiersystem K1620" und „Mikrorechnersoftware". Für die Sektion wurde ein Bürocomputerkabinett installiert, im Bereich Rechnersysteme ein Mikrorechnerlabor und in unserem Bereich ein halbautomatisches Datenerfassungssystem DARO 1600 in Kopplung mit einem ROBOTRON-Kleinrechner KRS 4200, die dann im Rahmen von Praktika für die Lehre genutzt worden sind. Ende des Jahres 1985 ist in Kooperation mit der Sektion 14 dort im Bereich von Prof. Dr. Zachau ein Industrieroboter des Kombinates Fortschritt mit einer von unserer Sektion mitentwickelten Steuerung IRS 650 vom VEB Numerik Karl-Marx-Stadt in der Absicht aufgestellt worden, diese Geräte ebenfalls in der Lehre zu nutzen. Nach der Gründung des Informatikzentrums unterblieb dies jedoch infolge anderer Prioritätensetzung.

5.2 Fernstudium

Der Immatrikulationsjahrgang 1972 beendete sein Studium als Diplomingenieur im Jahre 1977. Trotz teilweiser Freistellungen von etwa 20% durch die Betriebe stellte das Fernstudium doch eine große Belastung dar, so dass nur 61 den Abschluss erreichen konnten [Re77]. Ab 1976 wurde dann wieder in geringerem Umfang immatrikuliert.

5.3 Weitere personelle Entwicklung der Sektion 08

1977 löste Prof. Dr. Schubert Prof. Dr. Stahn als Sektionsdirektor ab und bekleidete diese Funktion bis zur Amtsübergabe an Prof. Dr. Loeper am 1.9.1984 [Ge78]. Die Bezeichnungen der drei Wissenschaftsbereiche der Sektion änderten sich in PT-Programmierungstechnik (Prof. Dr. Stiller), RS-Rechnersysteme (Prof. Dr. Löffler) und AIS-Automatisierte Informationssysteme (Prof. Dr. Stahn) [TU77]. Am Stichtag 19.2.1986 bedeuteten die 560 Studenten eine Quote von 40 pro Hochschullehrer, 8,2 pro wissenschaftlichem Personal und 6,6 pro Gesamtpersonal [1)].
In [Re85] sind in Auswertung der Dienstbesprechung des Ministers für das Hoch- und Fachschulwesen am 6.12.1985 mehrere Beratungen dokumentiert zum Zusammenschluss der Sektion 08 und der Ingenieurhochschule noch unter der Arbeitsbezeichnung „Informatik-Zentrum Dresdner Hochschulen IZD". In dieses sollen die Sektion Informationsverarbeitung und das Rechenzentrum der Ingenieurhochschule mit 44 Hochschullehrern und die Sektion 08 der TU mit 23 (oder 17?)[1)] Hochschullehrern zusammengeführt werden [Re85]. Diese Konzentration war sinnvoll, weil in den 1980-er Jahren keine gravierenden Unterschiede in der Ausbildung bestanden. Bei der Bildung des Informatikzentrums des Hochschulwesens an der TU Dresden im Oktober 1986 wurden ca. 85 Mitarbeiter der Sektion 08 (davon etwa 70 wissenschaftliches Personal) und etwa 240 Mitarbeiter der Ingenieurhochschule zusammengeführt.

[1] Nach eigenen Aufzeichnungen aus einem Leitungsseminar der Sektion 08 vom 19.2.1986 lehrten zu diesem Zeitpunkt allerdings nur 14 Hochschullehrer an der Sektion, und es war bis zum 1.9.1986 nur eine Erhöhung auf 17 bis 18 geplant.

5.4 Weiterbildung und Promotionsgeschehen

Im Jahre 1976 wurden lt. Beschluss von 1972 [Re72] die ersten 30 Postgradual-Studenten immatrikuliert, die innerhalb von zwei Jahren eine Zusatzausbildung auf dem Gebiet der Informationsverarbeitung erhalten sollten. An der TU Dresden immatrikulierten wir nun alle zwei Jahre etwa 30 Studenten – alternierend mit der TH Magdeburg.
Die Problemseminare der Sektion fanden ab 1978 mit denen der Mathematik im Rahmen des Weiterbildungszentrums Mathematische Kybernetik und Rechentechnik / Informationsverarbeitung (WBZ MKR/IV) statt, welches aus dem 1971 vom Minister für das Hoch- und Fachschulwesen 1971 gegründeten WBZ MKR hervorgegangen war, geleitet von Prof. Dr. Lehmann. Von 1980 bis Mitte 1985 haben in diesem Kontext 26 Problemseminare und 16 Lehrgänge der Sektion 08 stattgefunden [Re82]. Themen waren u.a. Datenbanken, Steuerungssoftware für Industrieroboter, Automatisierte Systeme für technologische Prozesse, Höhere Programmiersprachen in der Prozeßsteuerung, Softwaretechnologie, Programmierung von Mikrorechnern, Programmiersprachen und Sprachimplementation, System- und Echtzeitprogrammiersprachen, Rechnerverbundsysteme, Lokale Rechnernetze und Rechnerunterstützte Lehrsysteme.
Im Zeitraum von 1972 bis Juni 1979 wurden 64 Promotionen abgeschlossen [Se71]. Am 19.2.1986 betrug die Anzahl der Forschungsstudenten nach eigenen Aufzeichnungen 8 und die der planmäßigen Aspiranten 14.

6 Schlussbemerkungen

Mein Dank gilt der TU-Leitung und im Archiv Frau Buchwald für die Möglichkeit bzw. die Unterstützung bei der Recherche. Weiterhin danke ich den Professoren Hebenstreit, Herrlich, Loeper, Schubert, Voß und ehemaligen Studenten und Mitarbeitern für Hinweise und Überlassung von Unterlagen.

Literaturverzeichnis

[BBH04] Born, W.; Burkhardt, P.;, Hollnagel, G.: Die Anfänge des Prozeßrechnereinsatzes in der DDR in den sechziger Jahren. [NS06], S. 204
[Ei69] Einladungen und Protokolle der Fakultät für Datenverarbeitung 1969 – 1974, TU-Archiv Nr. 453
[Fa69] Fakultät für Datenverarbeitung 1969 – 1974. TU-Archiv Nr. 453
[Fa75] Fakultät für Elektrotechnik-Elektronik 1975 – 1976. TU-Archiv Nr. 454
[Fr71] Friedrich, Ch.: Lehrveranstaltungen des Studienjahrgangs 1971, unveröffentlicht
[Fu04] Fuchs-Kittowski, K.: Grundlinien des Einsatzes der modernen Informations- und Kommunikationstechnologien in der DDR. Wechsel der Sichtweisen zu einer am Menschen orientierten Informationssystemgestaltung. [NS06], S. 55
[Ge78] Geschichte der Technischen Universität Dresden 1828 – 1978. Verlag der Wissenschaften, Berlin 1978
[He04] Herrlich, O.: Gründung und Wirken der Sektion Informationsverarbeitung der Technischen Universität Dresden – ein Gedächtnisbericht. [NS06], S. 319
[He06] Herrlich, O.: Ein Blick zurück im Informatikjahr. Dresdner UniversitätsJournal 17 (2006) Nr. 7, S. 10

[Ho69] Hochschulstatistik 1969, TU-Archiv Nr. Dir. 4579/13, Teil „EDV-Berichterstattung“, Anlage Sektion 8

[Ho70] Hochschulstatistik 1969, TU-Archiv Nr. Dir. 4579/14, Teil „EDV-Berichterstattung“, Anlage Sektion 8

[HP00] Hänseroth, Th.: Pulla, R. (Hrsg.): Elektrotechnik und Informationstechnik an der TU Dresden in der DDR. Beiträge des Zeitzeugenkolloquiums am 14. März 2000 in Dresden. Zentrale Vervielfältigung der TU Dresden 2000

[Kr00] Krause, W.: Zur Entwicklung der Feinwerktechnik – mit Diskussion. [HP00], S. 66

[Le78] Lehmann, N.: Die Entwicklung und die Aufgaben im Wissenschaftsbereich Mathematische Kybernetik und Rechentechnik in der Sektion Mathematik der Technischen Universität Dresden. Wiss. Zeitschrift der TU Dresden 27 (1978) H. 1 „150 Jahre TU Dresden“, S. 101

[Lu93] Lunze, K.: Entwicklung der Fakultät Elektrotechnik von 1952 bis 1990. Wissenschaftliche Zeitschrift der TU Dresden 42 (1993) H. 2, S. 3-16

[Mi68] Ministerrat der Deutschen Demokratischen Republik. Ministerium für Hoch- und Fachschulwesen H. Lehrprogramm für das Lehrgebiet Informationsverarbeitung zur Ausbildung in den Grundstudienrichtungen der technischen Wissenschaften (mit Ausnahme der Grundstudienrichtungen Informationsverarbeitung, Städtebau und Architektur) und weiterer ausgewählter Grundstudien- und Fachrichtungen an Universitäten und Hochschulen der DDR (Kat.-Nr. 01 8000 31 2), Berlin 1968

[NS06] Naumann, F.; Schade, G. (Hrsg.): Informatik in der DDR – eine Bilanz. Symposien 1. bis 9. Oktober 2004 in Chemnitz und 11. bis 12. Mai 2006 in Erfurt. GI-Edition Lecture Notes in Informatics (LNI) – Thematics. Gesellschaft für Informatik, Bonn 2006. Köllen Druck+Verlag GmbH, Bonn. ISBN 978-3-88579-420-2

[Pe04] Petzold, H.: Zum Beitrag von Nicolaus Joachim Lehmann für die Herausbildung der Informatik in der DDR. [NS06], S. 104

[Pi04] Pieper, Ch.: Wechselbeziehungen zwischen Wissenschaft, Politik und Wirtschaft in der Hochschulinformatik der DDR (1960-er Jahre). [NS06], S. 351

[Po03] Pommerin, R.: 175 Jahre TU Dresden. Bd. 1 Geschichte der TU-Dresden 1828 – 2003. Böhlau-Verlag GmbH, Köln 2003

[Pr66] Protokolle der Arbeitsgemeinschaft Datenverarbeitung 1966 - 1968, Archiv der TU Dresden Nr. I/527 Rektorat

[Pr76] Protokolle der Kollegiumssitzungen des Rektors 1976. TU-Archiv-Nr. B1) Rekt. 400 [68 - 90]

[Pu03] Pulla, R.: Elektrotechnik und Informationstechnik – Strukturen und Konjunkturen in Thomas Hänseroth (Hrsg.): 175 Jahre TU Dresden. Bd. 2 Wissenschaft und Technik – Studien zur Geschichte der TU Dresden -. Böhlau-Verlag GmbH, Köln 2003

[Re68] Rektorat Bestandsakte 1968 – 1990 im Vorwort zu Aufgaben der neuen Fakultäten, TU-Archiv

[Re69] Rektorat 1968 – 1990, Wissenschaftlicher Rat. Sitzungsprotokolle des Wissenschaftsrates der TU Dresden mit Anlagen 1969. TU-Archiv Nr. 461. Protokoll der 6. Sitzung des Präsidium des Wissenschaftsrates am 29. 9. 1969, TOP 1 „Beratung über die Grundstudienrichtung „Informationswissenschaften“ und Anlage v. 15. 7. 1969

[Re72] Rektorat 1968 – 1990. Wissenschaftlicher Rat. Sitzungsprotokolle des Senats des Wissenschaftlichen Rates 1972-73, TU-Archiv Nr. 465. Protokoll der 24. Sitzung des Wissenschaftlichen Rates am 19. 6. 1972, TOP 1 „Studie zur Ausbildung auf dem Gebiet der Informationsverarbeitung an der TU Dresden“ v. 12. 5. 1972 und Anlage 1 „Ausbildungsprofil der Grund- und Fachstudienrichtung Informationsverarbeitung (Kurzfassung) v. März 1972

[Re75] Rektorat 1968 – 1990. Protokolle der Großen Dienstberatungen des Rektors mit den Sektionsdirektoren (8/75 – 12/76). TU-Archiv Nr. 409, (Beratung 8.-10.10.1975)

[Re76] Rektorat Protokolle der Kollegiumssitzungen des Rektors 1976. TU-Archiv Nr. B1) Rekt. 400 [68-90]
[Re77] Rektorat Protokolle der Großen Dienstberatungen des Rektors mit den Sektionsdirektoren Januar 1977 – Februar 1978. TU-Archiv Nr. 410
[Re81] Rektorat Protokolle der Großen Dienstberatungen des Rektors mit den Sektionsdirektoren Februar 1981 – Oktober 1981. TU-Archiv Nr. 416
[Re82] Rektorat – Wissenschaftlicher Rat. Plenartagungen 1981 – 1986. TU-Archiv Nr. 479
[Re85] Rektorat – Wissenschaftlicher Rat- Protokolle und Vorlagen zu Sitzungen des Senats des Wissenschaftlichen Rates 1985. TU-Archiv Nr. 489
[Se06] Seising, R.: Manfred Peschel (1932-2002): Systemverhalten – Systemversagen. [NS06], S. 483
[Se69] Sektionat Sektion Informationsverarbeitung. Protokolle der „Arbeitsgruppe Sektionsrat“ 1969 – 1980. TU-Archiv Nr. 15
[Se70] Sektion „08 Informationsverarbeitung“. Wissenschaftsentwicklung 1969 – 1978, TU-Archiv 3897/08 Bd. 2 Nr. 1
[Se71] Sektionat „08 Informationsverarbeitung“ Bd. 3+4 Leitungstätigkeit der Sektion 08 1970 – 1977. TU-Archiv 3897/08 Nr. 2
[St69] Stufenprogramm für die Hochschul- und Universitätsausbildung auf dem Gebiet der Elektronischen Datenverarbeitung/Informationsverarbeitung. Rahmenlehrprogramm für das Hochschulstudium. Ministerrat der Deutschen Demokratischen Republik, November 1969
[TU03] 175 Jahre TU Dresden. Bd. 3. Böhlau-Verlag GmbH, Köln 2003
[TU69] TU Dresden Universitätsverzeichnis 1968/1969 (Stand vom 31. 10.1968)
[TU70] TU Dresden Universitätsverzeichnis 1970
[TU72] TU Dresden Universitätsverzeichnis 1972
[TU74] TU Dresden Universitätsverzeichnis 1974
[TU77] TU Dresden Universitätsverzeichnis 1977
[Vo69] Voß, K.: Brief an meinen Bruder 19. 4. 1969 und Tagebuchnotizen Ende 1971/1972, unveröffentlicht
[Wi78] Wissenschaftliche Zeitschrift der TU Dresden 27 (1978) H. 1 „150 Jahre TU Dresden“

Die Studienrichtung Informationsverarbeitung an der TU Dresden – aus studentischer Perspektive

Thomas Kudraß

Fachbereich Informatik, Mathematik, Naturwissenschaften
Hochschule für Technik, Wirtschaft und Kultur Leipzig
Postfach 301166
04251 Leipzig
kudrass@imn.htwk-leipzig.de

Abstract: Der Beitrag beschreibt den Ablauf des Informatik-Studiums an der TU Dresden aus studentischer Perspektive im Zeitraum 1985-1990. Beginnend mit den Studienvoraussetzungen, werden die wichtigsten Inhalte der universitären Ausbildung in der Studienrichtung Informationsverarbeitung an der TUD erörtert. Dabei wird auch der Zusammenhang von Lehre und Forschung skizziert Beispielhaft wird das Fachgebiet Datenbanken etwas detaillierter betrachtet. Neben der fachlichen Sicht werden auch die Studienbedingungen der damaligen Zeit charakterisiert. Abschließend erfolgt eine zusammenfassende Bewertung des Studiums aus heutiger Sicht.

1 Voraussetzungen

Die Technische Universität Dresden war bis Mitte der 1980er Jahre die einzige Ausbildungsstätte der DDR, an der ein universitäres Informatik-Studium, noch unter der Bezeichnung „Informationsverarbeitung“, möglich war [He06]. Auf einen Studienplatz kamen ungefähr 0.5 Bewerber, so dass viele Studienanwärter im Ergebnis einer Umlenkung sich für dieses Studium entschieden. Einer der Gründe für dieses relativ geringe Interesse lag im geringen Bekanntheitsgrad dieses Studienganges sowie in der noch vorhandenen Dominanz der Mainframe-Rechner. Somit waren die Perspektiven für Absolventen in der Wirtschaft auf Großbetriebe und Rechenzentren konzentriert. Alle Studieninteressenten wurden nach einer Bewerbung zu einer persönlichen Vorstellung an die Sektion Informationsverarbeitung eingeladen, wobei Einzelgespräche mit Professoren geführt wurden.

Für die Aufnahme des Studiums wurde ein Nachweis über den Erwerb beruflicher Kenntnisse und praktischer Erfahrungen gefordert. Dies erfolgte in einem einjährigen Vorpraktikum. Für Wehrdienstleistende verkürzte sich die Dauer dieses Praktikums auf 5 bzw. 2 Monate. Der Anteil der weiblichen Studienbewerber lag bei ca. 50%, somit waren deren berufliche Erfahrungen vor Studienbeginn im Allgemeinen wesentlich besser als bei männlichen Kommilitonen. Vorgeschrieben war die Tätigkeit in einem Betrieb, der über ein Rechenzentrum verfügte, oder in einem Datenverarbeitungszentrum

(DVZ). Das Vorpraktikum sollte den Studienbewerbern die „Funktionsprinzipien der technischen Einrichtungen und Verfahren zur Darstellung, Speicherung und Verarbeitung von Informationen“ vermitteln. Neben der Gerätetechnik sollten auch Kenntnisse über den technologischen Prozess im Rechenzentrum und seine Operationen erworben werden.

Viele Studienbewerber, die nach dem Abitur ein Vorpraktikum absolvierten, wurden von ihrem Betrieb zum Studium „nachdelegiert“ (obgleich der Studienplatz schon vorhanden war). Die Beziehung zum Betrieb wurde somit auch während des Studiums aufrechterhalten, mit der Perspektive einer Rückkehr als Absolvent.

2 Randbedingungen des Studiums

Im Jahre 1985 wurden an der Sektion Informationsverarbeitung ca. 125 Studenten immatrikuliert, die in 6 Seminargruppen aufgeteilt wurden. An der Sektion gab es drei Wissenschaftsbereiche (WB): „Programmierungstechnik“, „Rechnersysteme“ und „Automatisierte Informationssysteme“ (AIS). Die Zuordnung der Seminargruppen zu den Wissenschaftsbereichen war festgelegt, woraus sich für die meisten Studierenden von Studienbeginn an automatisch die spätere wissenschaftliche Vertiefung ergab. Nur in Ausnahmefällen konnte diese noch geändert werden und war dann mit einem Seminargruppenwechsel verbunden.

Die Seminargruppenstruktur wurde sogar bei der Zuteilung der Plätze im Wohnheim berücksichtigt, das sich in unmittelbarer Nachbarschaft der Sektion befand. Aufgrund dieser räumlichen Nähe entstand eine recht intensive Gemeinschaft der Studierenden, die sich in vielen Situationen bewährte (z.B. Prüfungsvorbereitung), auch wenn die Wohnbedingungen aus heutiger Sicht katastrophal erscheinen.

Die Zusammensetzung der Studentenschaft blieb für die Dauer des gesamten Studiums (9 Semester) im Wesentlichen gleich. Allerdings erfolgte 1986 die Anwerbung von norddeutschen Studenten für das in diesem Jahr beginnende Informatik-Studium an der Universität Rostock. Es gab auch (geheim gehaltene) Versuche, Studenten für einen Wechsel zu einem Kybernetik-Studium in Ilmenau zu gewinnen, um später in der militärischen Forschung bei Carl Zeiss eingesetzt zu werden. Unter den ausländischen Studierenden bildeten Bulgaren im Jahrgang die größte Gruppe, auch aus Madagaskar, Nordkorea (bis 1989) oder der Mongolei kamen Studenten. Darüber hinaus gab es eine Anzahl von Berufsoffiziersbewerbern, die im Auftrag der NVA ihr Studium absolvierten und eine Perspektive als Führungskader für die „Automatisierung der Truppenführung“ besaßen. Die straffe Organisation des Studiums und das damit verbundene Prüfungspensum machte es möglich, dass alle Studenten nahezu zeitgleich ihr Studium mit der Verteidigung der Diplomarbeit abschlossen und im Februar 1990 als letzter „DDR-Jahrgang“ feierlich exmatrikuliert wurden.

Für die Informatik-Fächer gab es nur in sehr geringem Umfang Fachliteratur. In einigen Lehrveranstaltungen wurden Lehrhefte verteilt, insbesondere bei der Ausbildung in Programmiersprachen, oder es konnten Robotron-Handbücher ausgeliehen werden.

Einige Professoren hatten zu ihren Vorlesungen eigene Lehrbücher verfasst, die jedoch mit einem Preis von bis zu 50 Mark für Studenten sehr teuer waren: „Lokale Netze“ [Lö87], „Compiler und Interpreter“ [LJO87], „Betriebssysteme“ [Ka88]. Bundesdeutsche oder englischsprachige Fachbücher und Fachzeitschriften gab es in begrenztem Maße in der Sektionsbibliothek, allerdings nicht zur Ausleihe. Die Möglichkeiten zum Selbststudium über den Lehrstoff hinaus waren dadurch beschränkt.

Die rechentechnische Basis war zu Beginn des Studiums noch sehr gering, es überwog zunächst die Ausbildung an ESER-Rechentechnik, z.B. in der Programmierungstechnik. Seit 1986 erfolgte dann der Aufbau von PC-Kabinetten, die jedoch extrem überlastet waren. Somit lautete ein verbreitetes Motto: „Studenten rechnen in der dritten Schicht“. Neuerwerbungen von Hardware an der Sektion führten oft zu spontan anberaumten Praktika, so z.B. ein C-Programmierpraktikum auf einem VAX-Rechner des Rechenzentrums oder ein Programmierwettbewerb auf KC85-Homecomputern am WB AIS. Die rechentechnische Ausstattung an den Lehrstühlen war insgesamt etwas besser. Manche Studenten nutzten bereits persönliche Computer - mangels Westgeld: Marke Eigenbau.

3 Das Curriculum des Studienganges Informationsverarbeitung

3.1 Struktur des Studiums

Die Grundzüge des Curriculums entstammten noch den Anfangsjahren der Sektion Informationsverarbeitung, so dass der Immatrikulationsjahrgang 1985 der letzte mit einem Abschluss als „Diplomingenieur für Informationsverarbeitung“ war [He06]. Die Gründung des Informatik-Zentrums 1986 ging einher mit einer Neustrukturierung des Informatik-Studiums, von dem allerdings nur die nachfolgenden Jahrgänge betroffen waren.

Der zeitliche Ablauf des Studiums 1985-90 gestaltete sich wie folgt:

- 1. Semester: Mathematik: Analysis, Lineare Algebra, Gerätetechnik (GT), Grundlagen der Programmierungstechnik (GPT), Physik, Russisch, Englisch, Marxistisch-leninistische Philosophie (ML-DHM)
- 2. Semester: Mathematik: Analysis, Differentialgleichungen, Gerätetechnik (GT), Maschinenorientierte Programmierungstechnik (MOPT), Determinierte Automaten und Systeme (DAS), Physik, Russisch, Englisch, ML-DHM
- 3. Semester: Wahrscheinlichkeitsrechnung/Mathematische Statistik, Physik, MOPT II, Problemorientierte Programmierungstechnik (POPT), Russisch, Englisch, Politische Ökonomie (POLÖK)
- 4. Semester: Numerische Mathematik, Mathematische Optimierung, MOPT III, POPT II, DAS II, Automatisierte Systeme der Informationsverarbeitung (AIS), Mikrorechentechnik (MRT), POLÖK

- 5. Semester: Mathematische Optimierung, MOPT IV, Datenbanken (DB), Sozialistische Arbeitswissenschaft (SAW), Sozialistische Betriebswirtschaft (SBW)[1], wahlobligatorisch: Künstliche Intelligenz (KI)
- 6. Semester: Betriebssysteme, Übersetzerprogrammtechnik (ÜPT), POPT III, Informationsgewinnung und -übertragung (IGÜ), Informationstheorie (IT), Steuern und Regeln (SR), SBW, Wissenschaftlicher Sozialismus (WISO), wahlobligatorisch: Künstliche Intelligenz (KI)
- 7. Semester: Ingenieurpraktikum (zumeist im Delegierungsbetrieb)
- 8. Semester: IGÜ, Systemmodellierung (SYSMOD), Technologie in Organisations- und Rechenzentren (TORZ), Prozess- und Fertigungssteuerung (PROFEST), Sozialistisches Recht, wahlobligatorisch: Informationsrecherche-Systeme (IRS), Großer Beleg, Forschungsseminar, Hauptprüfung
- 9. Semester: Diplomarbeit

Insgesamt lassen sich die Lehrveranstaltungen wie folgt klassifizieren:

Bereich	**Fächer**
Mathematik	Analysis, Lineare Algebra, Differentialgleichungen, Wahrscheinlichkeitsrechnung/Statistik, Numerische Mathematik, Optimierung (2 Sem.)
Physik	Physik / Physikalisches Praktikum (3 Sem.)
Informatik: Grundlagen	DAS (2 Sem.), Informationstheorie, GPT
Informatik: Programmierungstechnik	GPT, MOPT (4 Sem.), POPT (2 Sem.), ÜPT Betriebssysteme, AIS
Informatik: Datenbanken und Informationssysteme	Datenbanken, SYSMOD, KI (wo, 2 Sem.), IRS (wo), SR, PROFEST
Informatik: Rechentechnik	GT (2 Sem.), MRT, IGÜ (2 Sem.)
Sonstiges	SAW, TORZ
Fremdsprachen	Russisch (3 Sem.), Englisch (3 Sem.)
Gesellschaftswissenschaften	ML-DHM (2 Sem.), POLÖK (2 Sem.), WISO (2 Sem.), SBW (2 Sem.), Soz. Recht

Tabelle 1: Inhaltliche Struktur des Studiums

[1] Der Zusatz „Sozialistisch“ wurde im Falle von SBW und SAW bei der Erstellung der Abschlusszeugnisse im Februar 1990 bereits gestrichen.

Die Hauptprüfung, die am Ende des 8. Semesters vor Ausgabe des Diplomthemas abgelegt werden musste, umfasste alle Fachgebiete des Studiums. Neben der dabei geforderten fachlichen Breite wurde den Absolventen auch das Verständnis von Zusammenhängen unterschiedlicher Fachgebiete abverlangt. Eine Abgrenzung zwischen einem Grundstudium (abgeschlossen durch ein Vordiplom) und einem darauf aufbauenden Studium gab es nicht. Die Darstellung des Curriculums zeigt aber, dass der Anteil der Nicht-Informatik-Fächer in den unteren Semestern deutlich über 50% lag.

3.2 Inhaltliche Schwerpunkte der Informatik

Die nachfolgend angegebenen Schwerpunkte der Hauptprüfung widerspiegeln die inhaltliche Struktur der Informatik-Ausbildung im Studium.

1. Grundlagen der Informatik

- Relationen, Eigenschaften binärer Relationen auf einer Menge
- Operationen auf Mengen und Relationen
- Darstellungsformen von Algorithmen
- Graphen als Beschreibungsmittel für Strukturen
- Shannonscher Informationsbegriff, Entropie, semantischer und pragmatischer Informationsaspekt, Codierungstheorie
- Digitale Simulationstechnik, Simulationssprache SIMDIS

2. Programmierungstechnik

- Problemorientierte Programmierungstechnik: Datentypen und Datenstrukturen und deren Deklaration, File- und E/A-Konzepte in einer ausgewählten höheren Programmiersprache
- Betriebssysteme: Arten von Betriebssystemen, Funktionselemente und Struktur eines Betriebssystem-Kerns, Funktionen von Systemprogrammen (Assembler, Compiler, Interpreter, Generator, Lader, Programmverbinder), Speicherverwaltung, Jobsteuerung, Unterbrechungssteuerung
- Softwaretechnologie: Software-Lebenszyklus, Entwurfsmethoden, Testmethoden, Aufbau von Programmierumgebungen, Kriterien der Softwarequalität und Lösungswege zur Verwirklichung einer hohen Softwarequalität
- Grundzüge der Theorie der formalen Sprachen und der Übersetzerprogrammtechnik

3. Datenbanken und Informationssysteme:

- Basisprozess / Basissystem – Informationsprozess / Informationssystem
- Analyse und Modellierung von IS: Prozess- und Systemanalyse und -beschreibung, Klassen und Eigenschaften von IS
- Konzeptuelle Informationsmodelle
- Theorie der Coddschen Relationen
- Datenbanktechnologie: Datenmodelle, Datenbeschreibungs- und Datenmanipulationssprachen mit datenmodelltypischen Operatoren, Charakteristik von Datenbankbetriebssystemen (DBBS), Datenbankanwendung zur Stücklistenverwaltung

- Dateiorganisation und Zugriffsmethoden: Indexsequentielle Speicherung, Hash-Dateien, B-Baum-Indexe, inverse Dateien, Multilistenorganisation
- Relationale DBBS und DB-Sprachen (SQL)
- Steuerungssysteme: Maschinenbelegungsproblem und Lösungsmethoden
- Grundzüge der Wissensverarbeitung / Aufbau und Funktionsweise von Expertensystemen

4. Rechentechnische Basis von Automatisierten Informationssystemen:

- Arbeitsweise eines informationsverarbeitenden Automaten, von-Neumann-Automat, Grundzüge von Post-von-Neumann-Rechnern (5. Rechnergeneration)
- Rechnerklassen und ihre typischen Merkmale: Universalrechner, Kleinrechner, Personal- und Arbeitsplatzcomputer, Mikrorechner, Hochleistungsrechner
- Rechnersysteme und Rechnernetze: Konzepte der Rechnerkopplung, OSI-Schichtenmodell, Protokollbegriff, Paketvermittlung/Leitungsvermittlung, LAN/WAN

5. Weltanschauliche, ökonomische und soziale Aspekte moderner Informationstechnologie und der flexiblen Automatisierung:

- Rolle der Informatik und der Rechentechnik bei der Verwirklichung der ökonomischen Strategie und der Schlüsseltechnologien
- soziale Auswirkungen des Einsatzes der Rechentechnik in der kapitalistischen und in der sozialistischen Gesellschaft

3.3 Verbindung von Theorie und Praxis, Lehre und Forschung

Von Beginn des Studiums an wurde der Verbindung von Theorie und Praxis große Bedeutung zugemessen, das Vorpraktikum spielte für die Berufsorientierung eine wichtige Rolle (siehe Kapitel 1). Während des Studiums waren zwei weitere Praktika in der Wirtschaft obligatorisch: ein 4-wöchiges Sommerpraktikum sowie das Ingenieurpraktikum für die Dauer des 7. Semesters.

Zahlreiche Informatik-Lehrveranstaltungen wurden durch kleine praktische Projekte begleitet, die mit einem Beleg abgeschlossen wurden. Das Spektrum der praktischen Ausbildung reichte hierbei von Maschinencode- und Makro-Assemblerprogrammierung (ESER), über PL/1, ALGOL-68, Mikrorechner-Assembler, SIMDIS (zur Simulation von Prozessen), Programmierung von Steuerungen (z.B. Ampel), Analogrechnerpraktikum bis hin zu einer Löt-Übung. In den Programmierbelegen musste neben dem korrekten Resultat auch die jeweilige Vorgehensweise (Analyse, Lösungskonzeption) dokumentiert werden. Der Umfang einer solchen Arbeit führte in einigen Fächern dazu, bei einer sehr guten Bewertung des Beleges diesen als Prüfungsleistung anzuerkennen.

Die Verbindung von Lehre und Forschung wurde durch die Zuordnung der Seminargruppen zu Wissenschaftsbereichen ermöglicht. Hierdurch entstand über die Seminargruppenberater der Kontakt zu den jeweiligen Hochschullehrern und Wissenschaftlern des Bereichs. Somit erhielt man einen Einblick in dessen aktuelle Forschungsprojekte; interessierten Studenten wurden frühzeitig Möglichkeiten zur Mitwirkung angeboten.

Sehr viele Projekte trugen den Charakter von Auftragsforschung, weil sie in Zusammenarbeit mit Betrieben durchgeführt wurden (z.B. in Dresden: MIKROMAT, ZMD, Robotron-Projekt). Für die wissenschaftliche Vertiefung gab es in jedem WB ein Forschungsseminar, das thematische Orientierung und Motivation für die Diplomarbeit bieten sollte. Es gab Fächer, die stark durch die Persönlichkeit des Lehrenden und seine wissenschaftlichen Ideen geprägt waren. Beispielhaft genannt sei hier Prof. Heinz Stahn mit der CAMARS-Technologie [ES89] für die Entwicklung flexibler Produktionssteuerungssysteme, deren Ansätze maßgeblich den Inhalt seiner PROFEST-Vorlesungen (Prozess- und Fertigungssteuerung) bestimmten.

3.4 Bewertung

Das Studium der Informationsverarbeitung an der TU Dresden zeichnete sich generell durch eine hohe Breite im Fächerspektrum aus. Diese entstand dadurch, dass nahezu alle Fächer des Curriculums als Pflichtfächer von den Studenten absolviert werden mussten. Der technologische Stand der Entwicklung der Rechentechnik und die damals aktuellen Forschungsschwerpunkte widerspiegelten sich auch in den Schwerpunkten des Studiums. Die Verbindung theoretischer Grundlagen wurde eng mit dem Erwerb praktischen Wissens verbunden.

Insbesondere die Ausbildung in der Programmierungstechnik erfolgte über mehrere Semester sehr systematisch „bottom-up“, d.h. beginnend mit Maschinencode-Programmierung bis hin zu Compilerbau. Vorrang hatten zunächst immer Konzepte. Erste „Programmiersprache“ war demzufolge Pseudo-Code, ehe dann ALGOL-68 oder PL/I folgten. Ähnlich verhielt es sich auch in anderen Fächern, z.B. Datenbanken oder Prozess- und Fertigungssteuerung, bei denen sehr großer Wert auf die Modellbildung bei der Systementwicklung gelegt wurde.

Die Mathematik-Ausbildung war den Anforderungen des Informatik-Studiums angemessen. Hierbei wurden Informatiker primär als Ingenieure angesehen, die zur Anwendung bestimmter mathematische Verfahren befähigt werden sollten. Die Ausbildung in Physik über drei Semester war nicht auf Informatiker zugeschnitten und hatte somit keine nennenswerte praktische Bedeutung.

Die gesellschaftswissenschaftlichen Anteile des Studiums waren bedingt durch die in der DDR üblichen ideologischen Vorgaben (ML) und wurden von den Studierenden durchweg als unnützer Ballast empfunden. Positive Ausnahme war das Fach „Arbeitswissenschaften“, das von einem Arbeitspsychologen gelesen wurde und viele Anregungen enthielt, z.B. für die Gestaltung von Benutzeroberflächen. Auch Soft Skills wurden hier schon thematisiert (allerdings noch nicht unter diesem Namen).

Die Wahlmöglichkeiten (und damit auch die Abwahlmöglichkeiten) bei Lehrveranstaltungen waren für Studenten recht eingeschränkt und kamen eigentlich erst bei der Themenwahl für die wissenschaftliche Vertiefung (Praktikum, Großer Beleg, Diplomarbeit) zur Geltung. Die Mitarbeit an sog. Jugendobjekten u. a. studentische Projektarbeiten wurden generell an der Sektion gefördert und oftmals als adäquate Prüfungsleistung anerkannt, so dass hierdurch Freiraum für studentische Eigeninitiative entstand.

4 Das Fachgebiet Datenbanken

Im WB „Automatisierte Informationssysteme“ war das Fachgebiet Datenbanken angesiedelt, das von Prof. Dietrich Schubert sowie später auch von Prof. Karl-Heinz Müller in der Lehre vertreten wurde. Die Zusammenführung der Sektion 08 (Informationsverarbeitung) mit der Ingenieurhochschule Dresden (IHD) 1986 hatte für die Datenbanken-Ausbildung fast keine Konsequenzen, zumal die Trennung der Studenten der TU und der früheren IHD in den einzelnen Lehrveranstaltungen fortbestand.

Das Curriculum des Studienganges Informationsverarbeitung wies zwei Pflichtveranstaltungen aus, die Bezug zum Fach Datenbanken hatten:

- Datenbanken (DB) / Prof. Müller im 5. Semester
- Systemmodellierung (SYSMOD) / Prof. Schubert im 8. Semester

Die Lehrveranstaltungen wurden durch Übungen und Praktika ergänzt. Eine wissenschaftliche Vertiefung war für die Studenten möglich, die ihren Großen Beleg bzw. die sich daran anschließende Diplomarbeit im Fachgebiet Datenbanken anfertigten. Die Teilnahme an einem begleitenden Datenbanken-Forschungsseminar war in diesem Fall obligatorisch.

Grundlagen-Lehrveranstaltung „Datenbanken“

Die Grundlagen-Lehrveranstaltung „Datenbanken“ umfasste folgende Schwerpunkte:

- Motivation
- Datenmodelle
- Datenbanksprachen (SQL)
- Physische Datenorganisation
- Netzwerkdatenbanken (DBS/R)
- Datenbank-Architektur

Der Einsatz von Datenbanken wurde motiviert durch betriebswirtschaftliche und technologische Anwendungen, die den Produktionsalltag der DDR widerspiegelten. So bildete eine durch einen Gozinto-Graphen beschriebene Stückliste die Beispiel-Datenbank, die durchgängig in der Vorlesung verwendet wurde. Die Behandlung der Datenbankeigenschaften erfolgte jedoch noch auf Basis einer netzwerkartigen Organisation der Datensätze. Es wurde sehr großen Wert auf die formale Beschreibung der Datenbanken gelegt – weniger auf deren praktische Umsetzung.

Als semantische Datenmodelle wurden das Entity-Relationship-Modell (ERM) behandelt, aber auch das ENALIM-Modell auf Grundlage von Sätzen der natürlichen Sprache. Das Relationenmodell und die Sprache SQL bildeten einen weiteren Schwerpunkt und wurden überblicksartig vermittelt. Das seinerzeit in der DDR vorherrschende DBS/R auf der Basis des Netzwerkmodells wurde ebenfalls vertieft. Die Behandlung der physischen Datenorganisation konzentrierte sich auf Indexstrukturen. Architektur und Implementierung von Datenbanksystemen waren nicht Gegenstand der Vorlesung, obwohl es zu dieser Zeit im Kombinat Robotron und in anderen Betrieben sehr viel Entwicklungserfahrung gab.

Aufbau-Lehrveranstaltung „Systemmodellierung“

Gegenstand der Lehrveranstaltung Systemmodellierung war die Modellierung von sog. Automatisierten Informationssystemen (AIS). Das Modell eines AIS umfasste dabei eine Informationsbasis (IB) und eine Informationsprozessbasis (IPB), deren Entwurf jeweils mit Hilfe verschiedener Modelle vertieft wurde. Der IB-Entwurf konzentrierte sich dabei vollständig auf die Entwurfstheorie für relationale Datenbanken. Die Prozess-Sicht auf ein Informationssystem wurde wesentlich durch das Entity-Operation-Event-Modell (EOEM) als konzeptuelles Modell ausgedrückt. Verglichen mit heutigen Modellierungsansätzen für Informationssysteme (z.B. ARIS, UML) lassen sich viele Parallelen finden, zumal die Koexistenz von Daten- und Prozess-Sicht beim Entwurf von Softwaresystemen weiterhin Gegenstand wissenschaftlicher Diskussion ist.

Wissenschaftliche Vertiefung

Als aktuelles Forschungsthema wurden bereits objektorientierte Informationsmodelle für sog. Non-Standard-Datenbankanwendungen behandelt unter Nutzung von Ansätzen der Wissensrepräsentation (Frame-Modell, Molekül-Atom-Datenmodell). Hier machte sich der Einfluss von entsprechenden Forschungsarbeiten am WB bemerkbar [Go91]. Als neuartige Anwendungen erschienen vor allem CAD-Systeme, die spezielle Anforderungen hinsichtlich des Managements komplexer Objekte stellten. In diesem Zusammenhang stand auch der geplante Einsatz des neuen DBMS INTERBAS [Bi91], das mit einer Koexistenzarchitektur für unterschiedliche Datenbankmodelle einen zu jener Zeit richtungweisenden Ansatz verfolgte. Die politische Wende 1990 verhinderte hierbei jedoch eine Weiterentwicklung und produktive Einführung von INTERBAS.

5 Das Studium der Informationsverarbeitung – eine Bilanz

Aus heutiger Sicht erscheint das Studium mit seiner starren Struktur vom 1. bis zum 8. Semester sehr verschult, auch wenn es damals von den Beteiligten keineswegs so empfunden wurde. Positiv hervorzuheben ist die breit angelegte Vermittlung von Fachwissen auf einem hohen theoretischen Niveau unter Einbeziehung zahlreicher Praktika. Die Verbindung zur Berufswelt war ebenfalls integraler Bestandteil des Studiums.

Die Darstellung der Studieninhalte macht deutlich, dass der Anteil von Nicht-Informatik-Fächern vergleichsweise hoch war, so dass zusätzliche Vertiefungsrichtungen oder gar ein Nebenfach keinen Platz im Studium fanden. Somit wurden die Studenten erst relativ spät im 8. Semester zu einer wissenschaftlichen Spezialisierung gezwungen, wenn sie sich nicht vorher selbst darum gekümmert hatten.

Eine Internationalisierung des Studiums war noch weitgehend unbekannt. Trotz eines relativ hohen Anteils von Fremdsprachen-Ausbildung am Studium gab es kaum Angebote für Auslandssemester. Hierfür bot sich nur das 7. Semester an, möglich war ein Aufenthalt am sowjetischen Forschungszentrum Dubna.

Im Zeitraum des Studiums (1985-90) beherrschte die Vision von der durchgängig automatisierten Produktion, „Computer Integrated Manufacturing“ (CIM), als Fortführung von CAD/CAM die Fachwelt. Die Entwicklung der Mikroelektronik und der Datenkommunikation lieferten hierfür wichtige Impulse [Fu06]. Die Lehrinhalte des Studiums nahmen zahlreiche Aspekte davon auf, die die Rolle der Informatik als „Schlüsseltechnologie“ für eine stark industriell geprägte Volkswirtschaft der DDR bestimmten.

Viele der damals behandelten Themen und Ansätze erscheinen auch heute noch aktuell, obwohl sich das Informatik-Studium in den letzten 20 Jahren durch den technischen Fortschritt stark verändert hat. Einige Fachgebiete wurden in der Informatik-Ausbildung der damaligen Zeit noch etwas unterschätzt, auch wenn viele theoretische und praktische Erkenntnisse schon vorhanden waren (z.B. Datenbanken). Die Weiterentwicklung der Hardware und die damit einhergehende Digitalisierung und Vernetzung brachten seit Ende des Studiums viele neue Disziplinen hervor wie z.B. Multimedia- und Internet-Technologien. Das Studium der Informationsverarbeitung schuf jedoch ein solides und vor allem breites Fundament für ein lebenslanges Lernen. Dessen Bedeutung wurde schon recht früh an der TU Dresden vermittelt.

Somit hatten alle Absolventen 1990 eine gute Grundlage und ideale Startposition für den Eintritt ins nunmehr gesamtdeutsche Berufsleben.

Literaturverzeichnis

[Bi91] Bittner, J.: Die Architekturkonzeption eines DBMS aus pragmatischer Sicht, in: Appelrath, H.-J. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Proceedings, GI-Fachtagung Kaiserslautern, 1991, Informatik-Fachberichte Bd. 270, Springer-Verlag.

[ES89] Engelien, M.; Stahn, H.: Software Engineering – CAMARS-Technologie, Akademie-Verlag, Berlin, 1989.

[Fu96] Fuchs-Kittowski, K.: Orientierungen der Informatik in der DDR – Zur Herausbildung von Sichtweisen für die Gestaltung automatengestützter Informationssysteme und zum Ringen um eine sozial orientierte Informatik, in: Naumann, F.; Schade, G. (Hrsg.): Informatik in der DDR – eine Bilanz, Lecture Notes in Informatics (LNI) – Thematics, Bd. T-1.

[Go91] Gorchs, T.: Adäquatheit im FORMAD-Modell als eine Voraussetzung zur Datenbankunterstützung für wissensbasierte Systeme, in: Appelrath, H.-J. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Proceedings, GI-Fachtagung Kaiserslautern, 1991, Informatik-Fachberichte Bd. 270, Springer-Verlag.

[He06] Herrlich, O.: Gründung und Wirkung der Sektion Informationsverarbeitung der Technischen Universität Dresden – ein Gedächtnisbericht, in: Naumann, F.; Schade, G. (Hrsg.): Informatik in der DDR – eine Bilanz, Lecture Notes in Informatics (LNI) – Thematics, Bd. T-1.

[Ka88] Kalfa, W.: Betriebssysteme, Akademie-Verlag, 1988.

[LJO87] Loeper, H.; Jäkel, H.-J.; Otter, W.: Compiler und Interpreter für höhere Programmiersprachen, Akademie-Verlag, Berlin, 1987.

[Lö87] Löffler, H.: Lokale Netze., Akademie-Verlag, Berlin, 1987.

Betriebssysteme
erfahren – erleben – begreifen - verstehen

Winfried Kalfa

Fachhochschule Dresden
Gasanstaltstrasse 3-5
01237 Dresden
winfried.kalfa@fh-dresden.eu

Abstract: 1982 wurde der Lehrstuhl „Betriebssysteme“ an der Sektion „Informationsverarbeitung“ der TU Dresden eingerichtet und zur Besetzung ausgeschrieben, die schließlich 1984 erfolgte. Betriebssysteme sind Systeme, die durch menschliche Sinne nicht wahrnehmbar sind und sich damit einer direkten Beobachtung und Beeinflussung von außen entziehen. In der universitären Lehre haben sich international einige Herangehensweisen herausgebildet, die mit mehr oder weniger Erfolg praktiziert wurden und werden. Im Vortrag wird darauf näher eingegangen und insbesondere die Betriebssystemlehre an der TU Dresden vor und nach 1990 dargestellt. Parallel dazu wird an geeigneten Stellen auf einige Vorurteile und Falschaussagen über den universitären „Mikrokosmos“ eingegangen. Der Beitrag soll dessen Dokumentation unterstützen, natürlich subjektiv aus der Sicht eines unmittelbar Beteiligten.

1 Einleitung

Der Titel des Symposiums „Informatik in der DDR“ erlaubt einige Deutungen und Folgerungen. Gab es „überhaupt“ Informatik? Wenn ja, war es eine „andere“ Informatik?

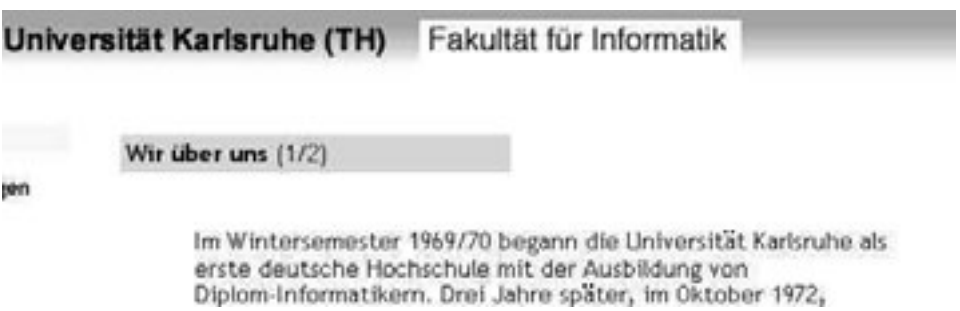


Abbildung 1: Auszug aus aktuellen (2008) Webseiten

> Lagen und liegen die damaligen Universitäten der DDR in Deutschland? Man könnte meinen, nein! Obwohl ich bereits 1992 den damaligen Dekan der Fakultät Informatik der Universität Karlsruhe darauf aufmerksam gemacht habe, dass die Ausbildung von „Diplomingenieuren für Informationsverarbeitung" bereits im Sommersemester 1969 an der TU Dresden begann, hält sich bis heute diese ***Falschaussage*** (s. Abb. 1), aber vielleicht war Dresden damals nicht deutsch, sondern DDR! Oder liegt es daran, dass Informationsverarbeitung als was anderes betrachtet wird als Informatik.

Semantisch sind beide gleich. Das hatte aber bereits Herrlich [HER04] klargestellt, dass der Begriff INFORMATIK seit langem im RGW[1]- Raum für ein Gebiet im Bibliothekswesen vergeben war, und erst als sich die Akademie der Wissenschaften der UdSSR[2] Anfang der 80-er Jahre der europäischen Begriffswelt anschloss, war es in den anderen RGW-Ländern opportun, Informatik zu verwenden, aus den Sektionen „Rechentechnik und Datenverarbeitung" bzw. Informationsverarbeitung wurden Sektionen INFORMATIK.

Es ließe sich noch weiter polemisieren. Gibt es eine historische Betrachtung „Informatik in der BRD"? Nein, man redet dann über „Informatik in Deutschland"! Dieses Symposium sollte dazu beitragen, dass es bis 1990 zwei Wurzeln für die Informatik nach 1990 IN DEUTSCHLAND gegeben hat. Ein kleiner Ausschnitt aus der östlichen Wurzel wird hier dargestellt.

2 Betriebssysteme an der TU Dresden

Die Bedeutung von Wissenschaftsgebieten unterlag in der DDR durchaus staatlich und parteilich gesteuerten Höhen und Tiefen, wobei meistens nach dem großen Vorbild UdSSR geschaut wurde. Auf einer der Höhen, die zusammenfiel mit der III. Hochschulreform der DDR, wurde die Sektion Informationsverarbeitung an der TU Dresden zum 01.01.1969 gegründet mit sofortiger Aufnahme des Lehrbetriebes. Trotz dünner Personaldecke waren Betriebssysteme von Anfang an Bestandteil der Ausbildung, die von Kollegen (Helmut Löffler, Dieter Werner) zusätzlich zu ihrem eigenen Lehrgebiet wahrgenommen wurden. Nach einem Tief bis Ende der 70-er Jahre gab es Anfang der 80-er Jahre einen erneuten Aufschwung von „Partei- und Staatsführung" für die EDV (Informatik). Die Sektion Informationsverarbeitung (08) der Technischen Universität Dresden nutzte die Chance für die Einrichtung weiterer Hochschullehrerstellen, darunter auch den Lehrstuhl "Betriebssysteme", der durch einen „Ordentlichen Professor" zu besetzen war.

[1] RGW = Rat für Gegenseitige Wirtschaftshilfe
[2] Union der Sozialistischen SowjetRepubliken

Das oft genannte ***Vorurteil*** „Professoren werden durch die Partei eingesetzt" lässt sich am Beispiel Betriebssysteme widerlegen. Priorität hatte die fachliche Qualifikation, die formal durch Promotion B (Dr. sc. techn. oder sc. nat.), Facultas Docendi, Auslandsaufenthalt und Lehrerfahrung (s. Abb.2) zu belegen waren. Man wurde von der Universität zum Vortrag eingeladen, wurde über seine Vorstellungen zu Lehre und Forschung befragt, Gutachten (national und international) wurden eingeholt, eine Vorschlagsliste über Fakultät, Senat an das MHF[3] gesandt. Von der zuständigen Fachabteilung des MHF wurde man zu einem „Berufungs"gespräch eingeladen, wo u. a. die Bereitschaft zu einem Auslandseinsatz und zu einem zeitweiligen Einsatz in der Ministerialbürokratie abgefragt wurde. Das Prozedere unterscheidet sich im Wesentlichen soweit nicht vom heutigen, zumindest was die Besetzung des Lehrstuhles „Betriebssysteme" an der TU Dresden betraf. Sicher gab es auch andere Beispiele, die gab es aber auch in den Jahren 1991 – 1993!


Technische Universität Dresden

Lehrstuhl für **Betriebssysteme**
zum 1.9.1983
Gefordert werden umfassende Kenntnisse in der Theorie und Technologie moderner Betriebssysteme, Erfahrungen in der Lehrtätigkeit an Hochschulen, Befähigung zur Leitung von Forschungsarbeiten, insbesondere zu Betriebssystemen für Steuerrechner und für Mehrprozessorsysteme sowie Erfahrungen aus der Tätigkeit in der Praxis einschließlich Rechenzentren.
Bewerbungen für den Lehrstuhl und für die Dozentur sind bis zum 30.7.1982 zu richten an:
Technische Universität Dresden
Rektorat
8027 Dresden
Mommsenstraße 13

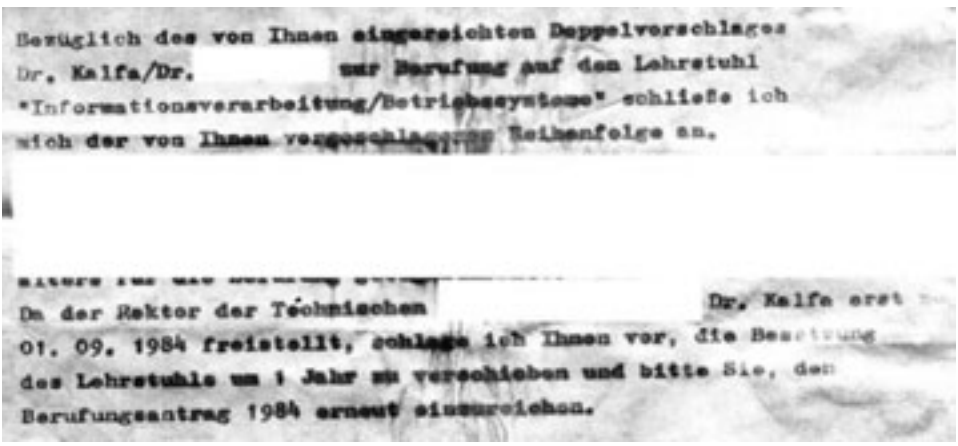
Das Hochschulwesen

3/82

Abbildung 2: Ausschreibung des Lehrstuhls „Betriebssysteme" an der TU Dresden

[3] Ministerium für Hoch- und Fachschulwesen

An verschiedenen Zeitpunkten im genannten Prozedere musste noch die Partei ihren „Segen“ geben oder verweigern. Quotenregeln (SED, Blockpartei, parteilos, Geschlecht, ...) spielten dabei die dominante Rolle, Loyalität zu Partei und Staat vorausgesetzt. Insofern stimmt das Vorurteil teilweise in dem Maße, wie es überall in Wirtschaft und Gesellschaft üblich war, dass die Partei das letzte Wort hatte. Das wurde jedoch auch intrigant genutzt (keine Gesellschaft ist vor solchen menschlichen Schwächen gefeit). Wenn man bereits an einer Hochschule arbeitete, benötigte man für einen Wechsel zu einer anderen die Freigabe durch den Rektor der Vorgängerhochschule. Wenn das jemand nicht wollte, erzählt man dem Parteisekretär Hochschule irgendeine Geschichte (z.B. „noch nicht reif für einen Lehrstuhl“), der dann dem Rektor rät, die Freigabe nicht oder nur verzögert zu geben (s. Abb.3).


Bezüglich des von Ihnen eingereichten Doppelvorschlages
Dr. Kalfa/Dr. zur Berufung auf den Lehrstuhl
"Informationsverarbeitung/Betriebssysteme" schließe ich
mich der von Ihnen vorgeschlagenen Reihenfolge an.

Da der Rektor der Technischen Dr. Kalfa erst
01. 09. 1984 freistellt, schlage ich Ihnen vor, die Besetzung
des Lehrstuhls um 1 Jahr zu verschieben und bitte Sie, den
Berufungsantrag 1984 erneut einzureichen.

Abbildung 3: Schreiben des Ministers für Hoch- und Fachschulwesen an den Rektor der TU Dresden vom x.x.1983 („Bezüglich des von Ihnen eingereichten Doppelvorschlages Dr. Kalfa/ Dr. X.X. zur Berufung auf den Lehrstuhl „Informationsverarbeitung/Betriebssysteme“ schließe ich mich der von Ihnen vorgeschlagenen Reihenfolge an. … Da der Rektor der Technischen xxx xxx Dr. Kalfa erst zum 1.9.1984 freistellt, schlage ich Ihnen vor, die Besetzung des Lehrstuhles um 1 Jahr zu verschieben und bitte Sie, den Berufungsantrag 1984 erneut einzureichen“)

Letztendlich war das Gebiet „Betriebssysteme“ an der TU Dresden Ende der 80-er Jahre gut vertreten, zum Lehrstuhl gehörten neben einem „Ordentlichen Professor“ (Winfried Kalfa) noch ein „Außerordentlicher Professor“ (Dieter Werner, Echtzeitsysteme, Theorie von Betriebssystemen) und ein „Dozent“ (Klaus Irmscher, Leistungsbewertung verteilter Systeme) sowie zwei unbefristete und vier befristete „Wissenschaftliche Mitarbeiter“.

2.1 Lehre

Betriebssysteme waren seit Beginn 1969 im Studienplan für Dipl.-Ing. Informationsverarbeitung, ab 1986 im Studienplan für Informatiker [MHF86] und ab 1990 im an die BRD-Regelungen angepassten Studienplan fest verankert.

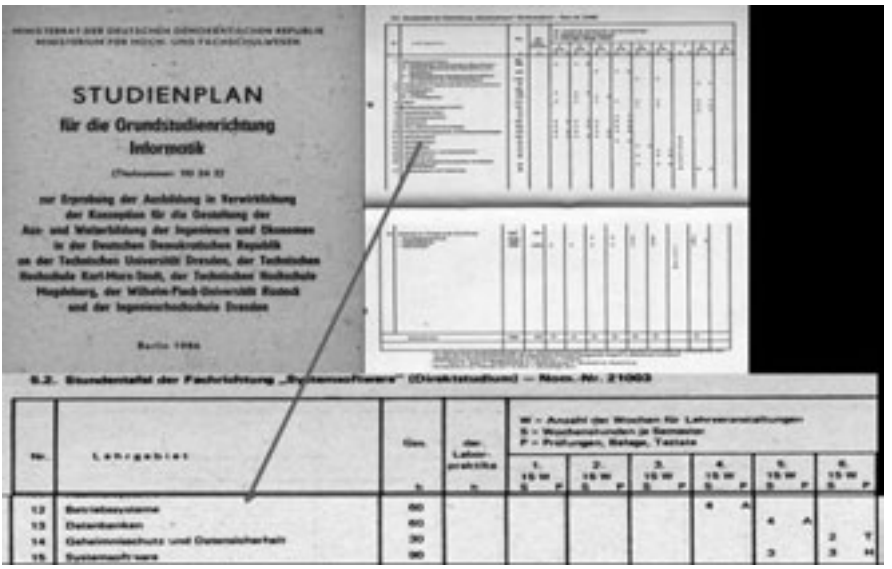


Abbildung 4: Der „vorletzte“ DDR-Studienplan Informatik

Der vorletzte DDR-Studienplan wurde wie alle Studienpläne vom MHF verbindlich für alle Universitäten herausgegeben, ausgearbeitet jedoch von einer Expertengruppe, die sich an aktuellen Studienplänen in der damaligen BRD und im Ausland orientierte, wie an den Curricula der ACM/IEEE [EAB83], [EAB86].

Figure 1-2. The 14 knowledge focus groups

Discrete Structures (DS)	Human-Computer Interaction (HC)
Programming Fundamentals (PF)	Graphics and Visual Computing (GV)
Algorithms and Complexity (AL)	Intelligent Systems (IS)
Architecture and Organization (AR)	Information Management (IM)
Operating Systems (OS)	Social and Professional Issues (SP)
Net-Centric Computing (NC)	Software Engineering (SE)
Programming Languages (PL)	Computational Science (CN)

Abbildung 5: Betriebssysteme im neuesten US-Curriculum als eine Fokusgruppe vertreten

> Er war mit seiner Aktualisierung 1989 einer der modernsten in Deutschland, womit das gelegentliche ***Vorurteil***, dass die fachliche Lehre an den Universitäten der DDR antiquiert gewesen sei, nicht ganz den Tatsachen entspricht. Die Lehre insgesamt und besonders im Gebiet „Betriebssysteme“ hat viele Jahrgänge sehr guter Absolventen hervorgebracht.

Mitarbeiter, Habilitanden, Promovenden des Lehrstuhls „Betriebssysteme“ sind heute Inhaber von Professuren an den Fachhochschulen Leipzig, Dresden, Mittweida, Köln, der Berufsakademie Sachsen, der Universitäten Ilmenau und Braunschweig. Auch im neusten Curriculum der Joint Task Force on Computing Curricula (IEEE Computer Society and Association for Computing Machinery) von 2001 behauptet sich das Gebiet der “Betriebssysteme” (s. Abb. 5) wie auch in aktuellen deutschen Curricula. Der letzte DDR-Studienplan Informatik (Abb. 6) brachte noch eine Stärkung der Betriebssysteme. Leider ist danach Anteil der „Betriebssysteme“ in deutschen Studienplänen Informatik gesunken.


TECHNISCHE UNIVERSITÄT DRESDEN
Fakultät für Informatik

Dresden, den 21. 6. 1990

STUDIENPLAN
zum
INFORMATIKSTUDIUM
an der
Technischen Universität Dresden

5.	Praktische Informatik							
	- Algorithmierung							
	Programmierung		4		4			
	- Softwaretechnik					4		
	- Betriebssysteme						4	?
	- Rechnernetze							4

Abbildung 6: Der „letzte“ DDR-Studienplan Informatik, spezifisch an der TU Dresden

An der TU Dresden wurde inhaltlich darauf Wert gelegt, dass Prinzipien, Wirkmechanismen und Methoden gelehrt wurden, konkrete Betriebssysteme dienten nur als Exempel! Methodik und Didaktik der Betriebsystemlehre wird in 3. näher betrachtet. Ergänzt wurde die Grundlehrveranstaltung durch spezielle Lehrveranstaltungen, wie „Theorie der Betriebssysteme“ (Dieter Werner) und „Echtzeitsysteme“(Uwe Schneider).

2.2 Forschung

Die Forschung am Lehrstuhl Betriebssysteme war gekennzeichnet durch folgende Linien:

- Spezielle Betriebssystemkerne und –dienste [SON91], [SCH91], [KAL92]
- Komponenten lokaler Netze (Multithreading in threadlosen Betriebssystemen, Fileserver) in Zusammenarbeit mit dem Lehrstuhl „Rechnernetze“ (Helmut Löffler) [KAL92]
- Terminalemulation inklusive Protokollimplementationen [OEL90]
- Methodik und Didaktik der Lehre [GOL90]



Abbildung 7: Auch der „Genosse Minister“ interessierte sich für LotuNet und Betriebssysteme

Mit den meisten der Themen haben wir nach 1990 einen guten An- und Einschluss in die für uns neue Forschungslandschaft gefunden [KAL94]. Die **Literaturbeschaffung** war nicht immer einfach, trotzdem erreichten uns die Standards der ECMA ebenso wie viele Sonderdrucke. Die Deutscher Bücherei war ein wahrer Literaturquell, denn dort gab es ohne große Verzögerung alle deutschsprachigen Veröffentlichungen. So konnten für die Studenten und Mitarbeiter notwendigen Informationen, z.B. über die Betriebssysteme CP/M-86 und MS-DOS, durch Abfotografieren und Reassemblieren des Hexcodes zusammengestellt werden. Nur mussten wir es bis 1989 SCP satt CP/M-86 bzw. DCP statt MS-DOS nennen.

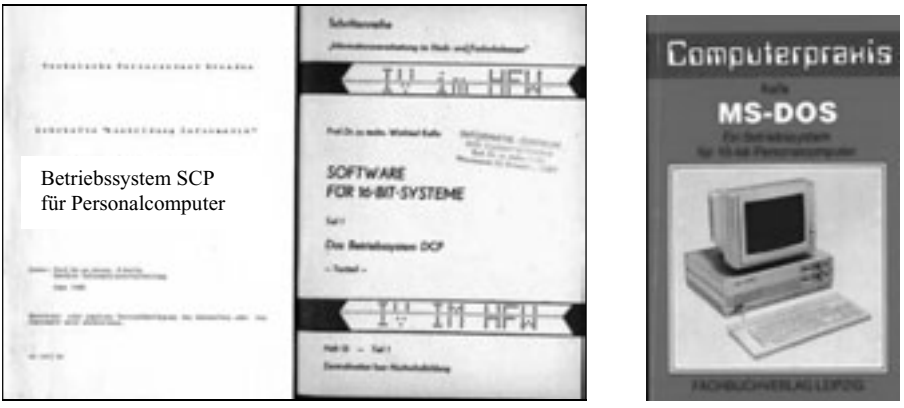


Abbildung 8: Schwierige Literaturbeschaffung

Es gab einige Kollegen mit KA[4]-Reiseerlaubnis, unter ihnen gab „solche“ und „sone“. Erstere gaben alles Papier, dessen sie habhaft werden konnten, an ihre Kollegen weiter, sogar ihre personellen Kontakte, aus denen Besuche dieser „West“kollegen (Helge Scheidig, Lutz Richter) in Dresden wurden und sogar Einladungen ins KA erfolgten oder nicht. Auf eine KA-Reiseerlaubnis musste man lange warten!

3 Wie lehrt man Betriebsysteme

Traditioneller Frontalunterricht an Schulen und Universitäten hat seine Grenzen. In technischen Disziplinen und Naturwissenschaften sind die Systeme, über die Kenntnisse, Fertigkeiten im Umgang bzw. Fähigkeiten zum Verständnis erworben werden sollen, weitgehend durch den Menschen wahrnehmbar und damit *begreif*bar und *erfass*bar. Betriebssysteme und ihre Teilsysteme sind es nicht! Gegenstände bei der Vermittlung von Wissen über Betriebssysteme sind:

- Architektur von Betriebssystemen bzw. deren Teilsysteme,
- Wirkprinzipien,
- Phänomene und ihre Ausprägungen in konkreten Systemen.

In der Lehre kann man diese Gegenstände

- sichtbar machen in
  - Bildern oder
  - durch Visualisierung von Logs realer Systeme,
- dynamisch erklären mittels Animationen,
- begreifbar machen durch Interaktion
  - mit Teilsystem realer Systeme bzw.
  - mit Simulationen von Phänomenen und Wirkprinzipien.

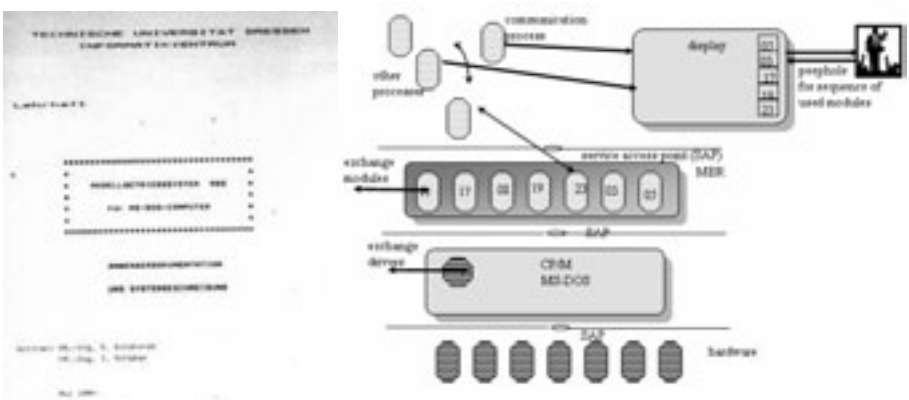


Abbildung 8: MBS[5] des Lehrstuhls Betriebssysteme auf Basis CP/M-86 und MS-DOS

[4] KA: Kapitalistisches Ausland
[5] ModellBetriebsSystem

Ein ***Vorurteil*** lautete, Initiative und Engagement zahlt sich nicht aus, „die da oben" bestimmen alles. Wenn man konnte und wollte und Gelegenheit dazu hatte, ging manches. In der Berufungsverhandlung im MHF stellte ich fest, dass an der TU Dresden keine technische Basis für eine praktische Betriebssystemausbildung vorhanden sei. Daraufhin wurde ich zum „Bilanzierer" für Rechentechnik im MHF geschickt. Da ich diesen aus 13 Jahren Rechenzentrumszeit kannte, schrieben wir gemeinsam den Bilanzbescheid 1985 für die TU Dresden: „25 PC 1715 an die Sektion 08 (Informatik) für die praktische Ausbildung der Studenten".

Das war die Basis für praktische Ausbildung mit dem selbst entwickelten MBS. Ein Semester Praktikum (300 Studenten) von 06:00 – 22:00 Uhr mit jeweils drei Aufgaben: parallele Prozesse, eine Routine des MBS ergänzen, einen Driver schreiben und testen.

Der Lerneffekt verbessert sich, beginnend mit dem *„Sichtbarmachen"* bis zur interaktiven Simulation. Letztere hat noch den Vorteil gegenüber realen Systemen, dass in ihnen die Abläufe determiniert reproduzierbar sind und man eine Reduktion auf das Wesentliche des Phänomens (Wirkprinzips) vornehmen kann.

Betriebssysteme mit all ihren Teilen, Wirkprinzipien und Phänomenen sind für den Menschen weitgehend nicht wahrnehmbar. und damit nicht ergreifbar und nur schwer begreifbar. Selbst grafische oder analytische Modelle sind noch zu fern von der Realität. Deshalb bieten sie sich geradezu an für interaktive Simulationen in der Lehre. Aus dem interaktivem Erleben und „Begreifen" wird **Verständnis**!

Unsere Simulationsmodule (Abb. 8) wurden nach einem einheitlichen Muster ausgelegt für

- Demonstration im Frontalunterricht,
- Animation in der individuellen Nachbereitung des Lehrstoffes,
- Experimente im Praktikum Betriebssysteme.

Die beiden ersten Nutzungen werden erreicht durch wohlüberlegte Vorgabewerte ausgewählter Szenarien.



Abbildung 8: Muster für Simulationsmodule



Abbildung 9: Hauptspeicherzuteilung (Java)

Während die ersten Arbeiten individuell mechanisch (Horst Wettstein, Karlsruhe, Mitte der 80-er Jahre) bzw. „Hand programmiert“ waren, erlaubten die neuen Plattformen, wie die kostenfrei auf den gängigen Rechensystemen verfügbar Java-Umgebung, eine komfortable Entwicklungsumgebung für die Simulation und die grafische Gestaltung der Oberfläche (Abb. 9).

4 Übergang

Am 03.10.1990 traten die Länder … der Bundesrepublik Deutschland nach Artikel 23 des [GG88] bei, wo es heißt: *“In anderen Teilen Deutschlands ist es nach deren Beitritt in Kraft zu setzen.”* Die Konsequenz daraus war, dass alle Regelungen der alten BRD zu übernehmen sind. Das war zu lernen, mit den Inhalten von Lehre und Forschung in der Informatik, insbesondere auch auf dem Gebiet der Betriebssysteme, hatten wir keine Probleme. Bezüglich der Regelungen unterstützen uns unsere Kollegen der alten BRD mit viel Papier, Beispielen und guten Ratschlägen. In Lehre und Forschung fanden wir gute Aufnahme unter gegenseitigem Respekt in Gedankenaustausch und informaler und formaler Zusammenarbeit in Projekten. Zur Konkretisierung seien – vor allem für die Zeit von 1990 bis 1992 – genannt:

- Wolfgang Glatthaar und Kollegen des ENC[6] Heidelberg von IBM Deutschland für Projekte mit der Industrie *(„Betriebssystemvoraussetzungen für Multimedia“)*
- Herbert Weber und Hans Decker von der Universität Dortmund für *Administration in der Universität*
- Hermann Härtig von der GMD in Birlinghoven für DFG-Projekte *(„Objektorientierte Betriebssysteme“)*
- Helge Scheidig von der Universität des Saarlandes für *Betriebssystemlehre*
- Fridolin Hofmann von der Universität Erlangen für die ad hoc Integration in den Arbeitskreis (heute Fachgruppe*) Betriebssysteme der GI*
- Horst Wettstein von der Universität Karlsruhe für die Forschung zur *Lehre von Betriebssystemen*

Bezüglich der Regelungen in der universitären Lehre und in der Forschungslandschaft haben wir schnell unsere „Hausaufgaben“ gemacht. Bezüglich des Inhaltes von Forschung und Lehre gab es keinen Bruch. Wir fanden Partner für unsere beibehaltenen Schwerpunkte. ABER mit den besseren Möglichkeiten in

- einer qualitativ hohen technischen Basis
- der vollen Integration in die weltweite community
- einer unbeschränkten Publikation, passiv wie aktiv.

[6] European Network Center

5 Zusammenfassung

Einige kursierende Vorurteile und Falschaussagen über die Informatik an den Universitäten der DDR stimmen einfach nicht, sondern es gilt:

- Es gab ein akademisches Leben, getragen vom gewillten Lehrkörper und den Studenten!
- Keiner der Absolventen musste eine „Nachqualifikation“ über sich ergehen lassen!
- Die Studienpläne in Informatik waren aktueller als an mancher Universität der alten BRD!
- Man konnte mit Initiative und Engagement auch in der DDR etwas „bewegen“!
- Viele Professoren fanden sofort Anschluss an die nunmehr „deutsche“ Wissenschaftslandschaft mit eigenen Drittmittelprojekten von Bund und Wirtschaft.

Die Studieninhalte zur Informatik aufgrund des Studienplanes [MHF87] entsprachen 1990 dem aktuellen internationalen Stand. Die schnell und unbürokratisch angelaufene Ausstattungshilfe brachte die Infrastruktur auf ein angemessenes Niveau. Kritisch war für einige Jahre nur die personelle Besetzung der Professuren.

Methodik und Didaktik der Betriebssystemlehre war konform mit der einiger westdeutschen Kollegen, so dass eine jahrelange fruchtbringende Zusammenarbeit entstand. Die nunmehr aufkommenden grafischen Möglichkeiten auf allen Rechnern erlaubte es, für den Menschen unmittelbar nicht wahrnehmbare Systeme wie Betriebssysteme in ihrer Struktur und Dynamik auf unterschiedlichste Art (Simulation und Animation von Prinzipien, Monitoring realer Systeme usw.) erfahrbar, erlebbar, begreifbar zu machen. Das Verständnis und damit die Bereitschaft zum bewussten und aktiven Wirken im Gebiet der Betriebssysteme werden damit erhöht. Zwei Wurzeln dieses wissenschaftlichen Interesses in Ost- und Westdeutschland führte ab 1990 zu einem gemeinsamen Bestreben. Ergebnisse wurden national und international veröffentlicht und mündeten in das BMBF-Programm „Neue Medien in den Hochschulen“.

Es konnte am Beispiel Betriebssysteme deutlich gemacht werden, dass der Übergang „Informatik in der DDR“ in „Informatik in Deutschland“ annähernd problemlos vonstatten gegangen ist, sofern es die fachbezogene universitäre Ausbildung und Forschung betrifft.

Literaturverzeichnis

[ACM01] Computing Curricula 2001 Computer Science (Final Report). Joint Task Force on Computing Curricula IEEE Computer Society/ACM, December 15, 2001.

[EAB83] Educational Activities Board. The 1983 model program in computer science and engineering. Technical Report 932, Computer Society of the IEEE, December 1983.

[EAB83] Educational Activities Board. Design education in computer science and engineering. Technical Report 971, Computer Society of the IEEE, October 1986.

[GG88] Grundgesetz der BRD, Stand 1988.

[GOL90] Goldhardt, S., J. Schäfer: MBS – Ein Modellbetriebssystem für 16-Bit Rechner. Wiss. Beitr. Inform. 5 (1991), H. 5.

[HER04] Herrlich, O.: Gründung und Wirken der Sektion Informationsverarbeitung der TU Dresden – ein Gedächtnisbericht. In: Informatik in der DDR, LNI GI-Edition 1, Bonn 2006.

[KAL92] Kalfa, W., C. Vogt: Betriebssystemumgebung für verteilte Multimediaumgebungen auf gekoppelten AIX Rechnern. IBM Hochschulkongress, 92. DR6.

[KAL92] Kalfa, W.: Komponenten verteilter Betriebssysteme lose gekoppelter Rechner. Wiss. Beitr. Inform. 5 (1991), H. 1.

[KAL94] Graupner, S., W. Kalfa, F. Schubert: Multi-level Architecture of object-oriented Operating Systems. TR-94-056 ICSI Berkley.

[KAL02] Kalfa, W.: How to comprehend not perceivable things? ED-Media 2002, Denver.

[MHF86] Studienplan für die Fachrichtungen Theoretische Informatik, Systemsoftware, Angewandte Informatik, Rechnersystemgestaltung und –betrieb innerhalb der Grundstudienrichtung Informatik. MHF 1986.

[OEL90] Oelsner, T.: Entwurfs- und Implemenationskonzept für die Steuerung der zeichenorientierten Datenübertragung. Wiss. Beitr. Inform. 4 (1990), H. 1.

[SCH91] Schalm, M.: Replikaten in verteilten Filesystemen. Wiss. Beitr. Inform. 5 (1991), H. 6.

[SON91] Sonntag, S.: Objekte in Birlix. Wiss. Beitr. Inform. 5 (1991), H. 6.

Das LEDA-Experiment

Hardwin Jungclaussen

Dresdner Str. 32
01326 Dresden
hj2@inf.tu-dresden.de

Abstract: Es wird von einem Experiment berichtet, das im Jahre 1979 mit dem Lehr- und Dateisystem LEDA durchgeführt wurde. Es sollte die Frage beantworten, ob der Computer befähigt werden kann, Studenten hinsichtlich Wissen und Fähigkeiten mit der Verlässlichkeit eines Lehrbeauftragten zu beurteilen. 75 Studenten lösten im Computerdialog Aufgaben in boolescher Algebra. Auf der Grundlage der während des Bildschirmdialogs gesammelten Daten beurteilte der Computer unter Verwendung statistischer und adaptiver Verfahren jeden Studenten. Das Experiment beantwortete die gestellte Frage positiv. Es wird ein Widerspruch der LEDA-Konzeption zur damaligen „Parteilinie“ aufgezeigt. Die Bezeichnungen „Informatik“ und „EDV“ werden einander gegenübergestellt.

1 Zwei Forschungsvorhaben aus dem Jahre 1970

Im Oktober 1969 wurde ich als Dozent an die neugegründete Sektion Informationsverarbeitung der TU Dresden berufen. Als solcher hatte ich einen Beitrag zur Erarbeitung eines Lehrkonzeptes Informatik zu leisten. Außerdem musste ich mir, wie alle neuberufenen Hochschullehrer, ein geeignetes Forschungsthema suchen. Mir kam der Gedanke, Lehre und Forschung unter der Überschrift „Rechnerunterstützte Lehre“ zu vereinigen. Bald stellte ich fest, dass viele Pädagogen und Informatiker auf diese Idee gekommen waren, sodass das Thema fast zu einer Mode geworden war. Dennoch wurde mein Vorschlag genehmigt und ein Forschungsthema eröffnet, das später den Namen „Lehr- und Dateisystem“, kurz LEDA erhielt. LEDA war eins von zwei Vorhaben, deren Bearbeitung ich 1970 in Angriff nahm. Das andere hieß „Asynchrone Operatorennetze“ und entwickelte sich zu einer Lehrkonzeption Informatik. Ich hatte erkannt, dass jedes informationsverarbeitende System eine Hierarchie von Hard- und Softwareoperatoren darstellt. Ausgehend von dieser Vorstellung wollte ich ein für alle Teilgebiete der Informatik einheitliches Begriffs- und Lehrgebäude entwickeln. Beide Themen wurden parallel bearbeitet.

2 LEDA

Noch vor meiner Berufung hatte unter den Mitarbeitern der neuen Sektion die Idee Fuß gefasst, alle Forschungsvorhaben unter den Aspekt „Adaptive Systeme“ zu stellen. Das war die Thematik, die ich mir für meine zukünftige Arbeit wünschte, als ich 1968 von Dubna zurückkehrte, wo ich an dem dortigen internationalen Institut für Kernforschung gearbeitet hatte. Ich hatte an der Suche nach transuranen Elementen teilgenommen und nebenbei Programme im Maschinencode der sowjetischen Maschine N20 geschrieben, um den Ablauf bestimmter Kernreaktionen zu simulieren. Gegen Ende meiner Dubna-Delegierung hat mich ein Vortrag über die adaptiven Eigenschaften künstlicher neuronaler Netzte so sehr fasziniert, dass ich mir vornahm, über adaptive Systeme zu arbeiten. Die Berufung zum „Dozenten für kybernetische Systeme der Informationsverarbeitung“ entsprach also meinen Wünschen.

Aus dieser Vorgeschichte ergab sich, als ich nach einem Forschungsthema suchte, fast mit zwingender Folgerichtigkeit der Gedanke, ein adaptives Lehrsystem aufzubauen, das die Fähigkeiten und das Wissen der Studenten mit Hilfe neuronaler Netze erkennt und auf dieser Grundlage den Lehrprozess für jeden Studenten individuell gestaltet. In der Pädagogik werden die Voraussetzungen, die ein Lernender für die Bewältigung des Lehrstoffs mitbringt, *Dispositionsmerkmale* genannt. In Abb.1 sind fünf allgemeine Dispositionsmerkmale aufgeführt, von denen die ersten beiden der Leistungszensierung, die restlichen einer allgemeinen Beurteilung der Studenten dienen könnten. Die große Frage war, ob der Computer befähigt werden kann, Studenten nach solchen Merkmalen mit der Verlässlichkeit eines Lehrbeauftragten zu beurteilen. Diese Frage wurde 1979 experimentell mit dem System LEDA-BA beantwortet, das für die Durchführung rechnerunterstützter Übungen in boolescher Algebra entwickelt worden war. LEDA wurde zunächst auf einem Honeywell DDP-516 implementiert. Der Rechner befand sich in einem Raum, dessen Fenster vergittert und mit Metallplatten verblendet waren, weil der Honeywell ein Embargo-Rechner war, der als Vorlage für die DDR-Prozessrechnerfamilie PR-4000 diente.

Die erforderliche Software war von Studenten, Diplomanden und Forschungsstudenten erarbeitet worden. Drei von ihnen möchte ich namentlich nennen: Horst Piehler, Heinz Pietzarka und Bernd Bergner. Alle drei sind heute führende IT-Fachleute und in Positionen mit hoher Verantwortung tätig. Zwei von ihnen sind Chefs ihrer eigenen Firmen. Aber nicht nur die drei Genannten, sondern alle Studenten waren mit Eifer, teilweise mit Begeisterung bei der Sache. Sogar im Bierkeller wurde über LEDA, neuronale Netze und die Geheimnisse des Gehirns diskutiert.

LEDA-BA konnte mit booleschen Werten und Variablen rechnen und Transformationen zwischen booleschen Ausdrücken und Schaltungen durchführen. Dadurch war es in der Lage, die Studenten beim Aufgabenlösen zu begleiten, jeden Lösungsschritt nachzuvollziehen und sämtliche Rechen- und Syntaxfehler, zu erkennen und zu protokollieren. Damit waren die Voraussetzungen zur Durchführung des Experiments geschaffen. An ihm nahmen 75 Studenten des zweiten Semesters teil. Jeder Student löste während einer ein- bis zweistündigen Sitzung am Bildschirm Aufgaben in boolescher Algebra. Der Computer registrierte pro Student ca. 100 Daten, u.a. Fehler und Denkzeiten. Die gesammelten Daten wurden zunächst durch einen *statistischen* Reduktor zu 20 metrischen Zwischenmerkmalen pro Student verdichtet, darunter waren absolute Werte, Mittelwerte und quadratische Abweichungen von Fehleranzahlen und Lösungszeiten. Sodann wurden die 20 Zwischenmerkmale eines Studenten durch einen *adaptiven* Reduktor auf eine der drei Noten gut, mittelmäßig, schlecht verdichtet. Auf diese Weise wurden die fünf in Abb.1 aufgeführten Dispositionsmerkmale des betreffenden Studenten benotet. Als adaptiver Reduktor diente das am Zentralinstitut für Kybernetik und Informationsprozesse in Berlin-Adlershof entwickelte Adaptive Klassifikationssystem STRAL (auch ESKKSA genannt), das nach dem Trennebenenverfahren [UW81] arbeitet, einem Mittelding zwischen neuronalen Netzen und Faktorenanalyse. In [BJ82] sind das Trennebenenverfahren und die Faktorenanalyse einander gegenübergestellt. Um LEDA in die Lage zu versetzen, verlässliche Noten zu vergeben, musste STRAL angelernt, d.h. sein Benotungsverhalten musste an das Verhalten einschlägiger „Experten“ angepasst werden. Die Rolle der Experten wurde von fünf Lehrbeauftragten übernommen. Jeder von ihnen benotete die Dispositionsmerkmale aller Studenten. Das gewogene Mittel der je fünf Expertennoten pro Merkmal und Student wurde zur *„wahren“ Note* erklärt. Die wahren Noten der Hälfte der Studenten wurden zum Anlernen von STRAL verwendet. Nach der Lernphase benotete STRAL die Dispositionsmerkmale der restlichen Studenten. Um die an das LEDA-Experiment gestellte Frage zu beantworten, wurden die Computernoten mit den „wahren“ Noten verglichen. Zu 82% stimmten die Noten überein. Diese Prozentzahl charakterisiert die *Computergüte* der Benotung. Sodann wurden die individuellen Expertennoten mit den „wahren“ Noten verglichen. Es ergab sich eine Übereinstimmung von 73% als Maß für die *Expertengüte* der Benotung. Sie lag 9% unter der Computergüte. In Abb.1 sind die Computer- und Expertengüten aufgeschlüsselt nach Dispositionsmerkmalen angegeben. Die Resultate erlauben eine positive Antwort auf die an das Experiment gestellte Frage. Die Aussagekraft der Antwort könnte angesichts der unterschiedlichen Expertenbenotungen zweifelhaft erscheinen. Doch ist zu bedenken, dass die Bedeutungen der Noten nicht klar definiert sind und dass die Semantik, mit denen ein Experte eine bestimmte Note belegt, beispielsweise die Note „schlechte Konzentrationsfähigkeit“, von seiner Erfahrung und Gewohnheit abhängt. LEDA *lernt* die Semantik der Noten aus den Äußerungen der Experten im Zusammenhang mit dem beobachteten studentischen Verhalten, also genau so, wie jeder Mensch die Bedeutung eines Wortes aus den Äußerungen seiner Mitmenschen *lernt*. Nach dem Lernprozess ist die Bedeutung einer Note in Form von Wertebereichen der 20 metrischen Zwischenmerkmale im Computer „abgespeichert“.

Dispositionsmerkmal	Computer-güte in %	Experten-güte in %	Anzahl der Meinungsänderungen zugunsten	
			Computer	Experten
Kenntnisse	79	72	11	4
Können	79	70	5	4
Arbeitstempo	71	76	4	4
Konzentrationsfähigkeit	95	73	7	1
Gleichmäßigkeit	87	72	3	2
Mittel / Summe	82	73	30	15

Abbildung 1: Beschreibung der Abbildung

Die Studenten, die an dem Experiment teilnahmen, benoteten sich auch gegenseitig. Die Noten wurden als *studentische Note* archiviert. 1982, in der Diplomphase, wiederholten die Studenten die gegenseitige Benotung. Das Ergebnis wurde mit den Noten von 1979 verglichen. Es wurden diejenigen Fälle herausgesucht, in denen die studentische Note vor drei Jahren *nicht* mit der „wahren“ Note übereingestimmt hatte, sich aber seitdem *geändert* hatte. 45 solche Fälle wurden gefunden. Überraschenderweise wurde festgestellt, dass sich die Meinungen der Studenten 15-mal zugunsten der Experten und 30-mal zugunsten des Computers geändert hatten. In Abb.1 sind die Anzahlen aufgeschlüsselt nach Dispositionsmerkmalen angegeben. Offenbar kannten sich die Studenten nach einem Semester gemeinsamen Studiums nicht so gut, wie der Computer sie nach einem kaum zweistündigen Dialog kannte.

Gleichzeitig mit dem Experiment wurde damit begonnen, LEDA vom PR 4000 auf ESER-Rechner zu transportieren. Außerdem wurde eine neue, erweiterte Konzeption erarbeitet. Sie sah ein Lehrsystem für die gesamte TU vor. Dafür wurden die Praktikumsaufgaben aller Sektionen studiert und die Lösungsmethoden katalogisiert. Im Rahmen eines Forschungsstudiums war die „LEDA-Systemsprache“ LESS entworfen und implementiert worden. Sie umfasste eine Sprache für den Übungsdialog, den die Studenten mit dem Computer führen, und eine Sprache für den Autorendialog, den die Lehrprogrammautoren mit dem Computer führen. Der Interpreter der Übungsdialogsprache enthielt einen Formelmanipulator für symbolische Rechnungen. Außerdem sah die neue Konzeption das Lehrgebiet Programmierungstechnik vor und es wurde die Entwicklung des Systems LEDA-PS in Angriff genommen. Eine erste Variante wurde implementiert und mit einem kleinen Übungsprogramm getestet. Auch dieses System unterschied sich von anderen uns bekannten Systemen durch seine Flexibilität. Das System unterstützte das Üben im modularen Programmieren. Im Gegensatz zu LEDA-BA war hier die Abspeicherung einer Antwort, in diesem Fall eines Musterprogramms, erforderlich. Die Antwortkontrolle erfolgte durch Prüfung der strukturellen Äquivalenz des studentischen und des Musterprogramms und durch Prüfung der Äquivalenz der

studentischen Modulen mit den Mustermodulen. Dabei kamen entsprechende Grammatikregeln zur Anwendung. In [He79] ist dargelegt, wie weit die Konzeption Ende 1979 realisiert war.

Zu dieser Zeit hatte LEDA im Kreise der vielen an Lehrsystemen arbeitenden Gruppen einen angesehenen Platz errungen. Wir pflegten Kontakte mit zahlreichen Institutionen und Arbeitsgruppen. Besonders intensiv waren die Kontakte mit dem Forschungszentrum für Technische Lehr- und Lernmittel an der TU Dresden, mit dem Rechenzentrum der TU Dresden, mit der damaligen TH Ilmenau, mit dem Kombinat Robotron und nicht zuletzt mit dem Zentralinstitut für Kybernetik und Informationsprozesse (ZKI) in Berlin-Adlershof. Namentlich möchte ich Siegfried Unger nennen, der mit seiner Arbeitsgruppe das System STRAL entwickelt und uns zur Verfügung gestellt hat. An dieser Stelle möchte eines verstorbenen Informatikers gedenken, mit dem mich eine nicht nur fachliche Freundschaft verband: Klaus Fritzsch. Er war Mitarbeiter des ZKI und hatte im LEDA-Kollektiv Forschungsseminare durchgeführt.

LEDA stand damals auf dem Höhepunkt seiner Laufbahn. Danach ging es sehr schnell bergab. Die sichtbare Ursache lag darin, dass das Thema keine Unterstützung vonseiten der Sektion mehr erfuhr, dass es nicht mehr die nötige Rechenzeit und auch nicht mehr die nötige Anzahl von Studenten erhielt. Dahinter stand der Konkurrenzkampf unter den neuberufenen Kadern um höhere Berufungen bei begrenzter Stellenzahl. Er konnte sehr unangenehme Züge annehmen, wenn er auf die Parteiebene hochgezogen wurde. Das möge ein geradezu groteskes Beispiel illustrieren. Der wohl fähigste Mann, den die Sektion damals hatte, und damit der gefährlichste Konkurrent für diejenigen, welche die eigene Karriere durch die anderer bedroht sahen, war Klaus Voss. Er war meiner Meinung nach prädestiniert, die junge Sektion zu leiten und ihr die Richtung zu geben. Stattdessen wurde er per Parteiverfahren ausgeschaltet und musste die Sektion verlassen. In dieser Atmosphäre hatte mein Projekt LEDA auf Dauer keine Chancen. Dass ich persönlich überlebt habe, verdanke ich zwei Menschen, Hans Heinold und Dieter Schubert. Prof. Heinold war bald nach Gründung der Arbeitsgruppe LEDA als deren Leiter eingesetzt worden. Er schirmte mich, wie er sich selbst ausdrückte „gegen den eisigen Wind, der in der Sektion wehte“, ab. Prof. Schubert ermöglichte mir, an meinem zweiten Vorhaben, den Operatorennetzen, in Ruhe weiterzuarbeiten, nachdem LEDA nicht mehr existierte. Von anderen Lehrsystemen mit ähnlichen adaptiven Eigenschaften, wie LEDA sie besaß, ist mir nichts bekannt.

Die Erwähnung des reglementierenden Eingreifens der SED in die Angelegenheiten der Sektion darf nicht als Beitrag zu Aufarbeitung der DDR-Vergangenheit verstanden werden und sollte bei niemandem bereits überwundene moralische Unter- bzw. Überlegenheitsgefühle zwischen Ost und West wieder aufleben lassen.

3 EDV oder Informatik?

Die beiden Vorhaben, Lehrsystem und Operatorennetze, wurden zunächst parallel bearbeitet. Das ergab sich aus der Notwendigkeit, das Programmsystem LEDA streng modular und sehr übersichtlich aufzubauen, weil die Programmierer jährlich wechselten.

Das wurde dadurch erreicht, dass dem Systementwurf das Operatorennetzkonzept zugrunde gelegt wurde. Dieses Konzept wurde später zu einer Methode der „Uniformen Systembeschreibung“ (USB-Methode) ausgearbeitet. USB erinnert an die Uniforme Modellierungssprache UML, von der wir erst viel später erfuhren. Die Kernidee beider Methoden ist das hierarchische Komponieren von Operatoren. UML ist in erster Linie auf die betriebliche Praxis orientiert, während USB eher theoretisch und auf einen sehr allgemeinen Anwendungsbereich orientiert ist. Mit der USB-Methode lässt sich jedes *kausaldiskrete* System beschreiben, d.h. jedes System, in dem Prozesse ablaufen, die kausale Ketten zeitdiskreter Zustände bilden. Der abstrakte Automat und das Petrinetz sind Musterbeispiele für abstrakte kausaldiskrete Systeme. Computer und Fertigungssysteme sind Beispiele für reale kausaldiskrete Systeme. Mit der USB-Methode können sowohl die Struktur als auch die Funktionsweise eines solchen Systems beschrieben werden, die Struktur in Form einer Operatorenhierarchie, die Funktionsweise, d.h. der Ablauf von Prozessen im System, in Form einer Petrihierarchie. In [Ju01] sind auf S. 506 das Operatorennetz und das Petrinetz für ein sehr einfaches System dargestellt. Ein Operator als Baustein einer Hierarchie stellt ein Netz aus Operatoren niederer Stufe dar. In den Verbindungen zwischen den Operatoren des Netzes dürfen Verzweigungen und Zusammenführungen liegen, die gegebenenfalls wie Weichen gesteuert werden können. Ein nach diesem Prinzip komponierter Operator heißt USB-Operator. In meinem Buch „Kausale Informatik“ [Ju01] habe ich gezeigt, dass alle Funktionen, die ein USB-Operator berechnen kann, dessen elementare Operatoren Inkrementierer oder die Elemente eines vollständigen Satzes boolescher Operatoren sind, rekursive Funktionen sind. Daraus ergibt sich eine neue Definition der rekursiven Funktion. Dieses Ergebnis halte ich persönlich für bedeutungsvoller als das Ergebnis des LEDA-Experiments. Dennoch hat es unter den Lesern keinerlei Reaktion ausgelöst. Offenbar hat mein Buch die von mir erhoffte Leserschaft nicht gefunden. Möglicherweise löst der Untertitel des Buches, der die Informatik als Lehre vom sprachlichen Modellieren durch Mensch und Computer bestimmt, bei vielen potenziellen Lesern, die das Buch zur Hand nehmen, einen ähnlich scharfen Protest aus, wie er bei mir seinerzeit die Umbenennung der „EDV“ (Elektronische Datenverarbeitung) in „Informatik“. Denn die Bezeichnung „Informatik“ suggeriert, dass es sich um Informationsverarbeitung durch den Computer handelt, während der Computer Daten verarbeitet, d.h. Zeichenketten, die vom Menschen interpretiert werden müssen, um zu „Informationen“ im üblichen Sinne des Wortes zu werden. Nebenbei sei bemerkt, dass die Bezeichnung „Informatik“ in der UdSSR - und auch in der DDR ursprünglich für eine Wissenschaft verwendet wurde, deren Gegenstand die wissenschaftliche Information und Kommunikation ist. Nach meiner Überzeugung wird aus der Zusammenführung von Computerwissenschaft und Gehirnforschung eine naturwissenschaftliche Disziplin hervorgehen, die zu Recht den Namen Informatik verdient. Begriffliche Sauberkeit könnte dadurch erreicht werden, dass zwischen *mentaler* und *physischer* Bedeutung unterschieden wird, d.h. zwischen der Wirkung einer Zeichenkette - oder allgemeiner eines Zeichenkonstrukts – im Bewusstsein eines Menschen einerseits und ihrer Wirkung im interpretierenden physischen System andrerseits. Dementsprechend wäre zwischen *bewusstseinsgebundener* und *systemgebundener* Information zu unterscheiden. Der Computer verarbeitet systemgebundene Information, der Mensch verarbeitet beide Informationsarten, wobei die verarbeitenden Prozesse mentaler bzw. neuronaler Natur sind.

4 Die LEDA-Konzeption – eine Horrorvision?

Neben dem Konkurrenzkampf sehe ich einen anderen, weniger sichtbaren Umstand, der LEDA angreifbar machte und möglicherweise dazu beigetragen hat, dass es dem Rotstift zum Opfer gefallen ist. Ich war schon damals der Ansicht, dass LEDA in das Gebiet der künstlichen Intelligenz einzuordnen ist. Als ich darüber mit Kollegen sprach, wurde ich gewarnt und belehrt, dass künstliche Intelligenz ein aus ideologischen Gründen verpönter Begriff sei, weil er der Parteilinie nicht entspräche. Zeitweilig gab es vonseiten der Partei sogar Einwände gegen den Begriff Kybernetik. Auf diesen Vorbehalt gegenüber modernen wissenschaftlichen Begriffen und die mit ihnen verbundenen Forschungsrichtungen war ich schon während meines Studiums in der UdSSR gestoßen, beispielweise gegen die Relativitätstheorie oder gegen die Psychologie, die, wenn auch nicht sehr lange, in die Rubrik „bourgeoise Pseudowissenschaften“ verbannt worden war. Den Grund für derartige Allergien sehe ich in der Angst der damaligen Machthaber, die Menschen könnten durch neue Ideen für die vorgeschriebene Ideologie verdorben und aufmüpfig werden. Die Wurzeln liegen aber wohl noch tiefer. Die kommunistische Ideologie umgab die menschliche Persönlichkeit mit dem Schein der Unantastbarkeit, fast der Heiligkeit. Diese Heiligsprechung stand in merkwürdigem Gegensatz zum offiziell verkündeten Atheismus. Doch sie war aus Sicht der Zielstellung des Marxismus-Leninismus verständlich. Denn, um die Massen zu ergreifen, musste ein hehres Ziel auf die Fahnen des Sozialismus geschrieben werden. Dieses Ziel war ein ideales, aber utopisches Bild vom Menschen in einer kommunistischen Gesellschaft. Aus der Apotheose des idealisierten Menschen wurden das Postulat seiner Unantastbarkeit und daraus eine Verurteilung aller Versuche, den Menschen und sein soziales Verhalten tiefer zu verstehen. Darin sehe ich die Wurzel des Misstrauens gegen alle human-, sozial- und geisteswissenschaftliche, ja sogar naturwissenschaftliche Gedanken und Bestrebungen. Diesem Misstrauen entsprang das Postulat, dass für die Forschung ideologische Kompetenz wichtiger ist als wissenschaftliche. Die Folge war eine restriktive Forschungspolitik, der viele Ideen und Ansätze zum Opfer gefallen sind. Aber ungeachtet der restriktiven Maßgaben durch die SED haben die Informatiker der DDR Leistungen hervorbracht, die sich in der Welt sehen lassen konnten bzw. hätten sehen lassen können, wenn der eiserne Vorhang nicht gewesen wäre. Das zeigen nicht zuletzt die Beiträge dieses Symposiums.

Wenn ich heute meine Idee betrachte, den Menschen zu einem „Objekt“ technischer Objekterkennung zu „deklassieren“, wundere ich mich, dass die damaligen Hüter der Parteilinie das LEDA-Konzept haben durchgehen lassen. Sie haben den „konterrevolutionären“ Inhalt wohl nicht erkannt. Aber wenn sie es verboten hätten, - womöglich hätten sie Recht damit getan? Das führt zu einer sehr prinzipiellen, von LEDA angeregten Frage. Ist es unbedingt als positiv zu werten, wenn der Computer Eigenschaften und Fähigkeiten lebender Personen beurteilt? Hier möchte ich einen Satz aus dem Vortrag zitieren, den Prof. Herrlich auf dem ersten Symposium gehalten hat. Er gab damals eine kurze Charakteristik von LEDA und sagte: „Alle gewonnenen Informationen über Nutzer, Autoren und Aufgaben sollten in einer Datenbank gesammelt und verwaltet werden. Eine Horrorvorstellung für jeden Datenschützer!“ Diese Horrorvorstellung will ich auf die Spitze treiben. Angenommen, es existiert ein Datenerfassungssystem, das sämtliche Bilder und Daten sowie sämtliche schriftlichen, akustischen, gestischen und mimischen Äußerungen aller Menschen erfasst und in einer Datenbank abspeichert. Außerdem existiere ein Erkennungssystem, das aus dem Inhalt der Datenbank und sogar schon allein aus der Physiognomie eines Menschen die Merkmale seines bewussten und unbewussten Innenlebens, seiner Veranlagung, seines Denkens und Handelns, seiner Neigungen und Ziele herauslesen kann. Jeder Mensch kann also rigoros aufgeblättert und für jeden anderen in jeder Hinsicht durchsichtig gemacht werden. Welche Möglichkeiten tun sich auf? Jeder kann sich über Verhalten und Eigenschaften jedes anderen Menschen genauestens informieren. Jedem Menschen kann sein optimaler Platz in der Gesellschaft zugewiesen werden. Die geeignetsten Kandidaten für Ministerposten und Präsidenten können gefunden werden. Aber auch potenzielle Verbrecher und Terroristen können erkannt werden. Derartige Phantasien könnten aus technischer Sicht früher oder später durchaus Wirklichkeit werden, es sei denn, der Gesetzgeber schiebt dem rechtzeitig einen Riegel vor. Mit dem Horrorszenario möchte ich auf eine negative, ja teuflische Seite des wissenschaftlich-technischen Fortschritts aufmerksam machen. Neue, von der Wissenschaft eröffnete technische Möglichkeiten bringen Gefahren mit sich, deren Wurzeln darin liegen, dass sich Gesetzgebung, Moral und Gewissen nur langsam an neue technische Möglichkeiten anpassen, während die kriminelle Energie sofort bei der Hand ist. Die Geschichte zeigt, dass die kriminelle Intelligenz schneller reagiert als die moralische. Eben daraus entspringen die Gefahren, wie sie der Menschheit aus der Kerntechnik erstanden sind und mit Sicherheit aus der Gentechnik entstehen werden. Sie werden auch aus der Informationstechnik erstehen. Sie existieren bereits durch die seuchenartige Ausbreitung sog. Schadsoftware. Noch viel größere Gefahren kündigen sich in der Finanzkriminalität an, der durch das Internet Tor und Tür geöffnet wird. Die Gefahren sind umso größer, je weiter sich der wissenschaftlich-technische Fortschritt dem Zugriff juristischer und moralischer Gesetze entzieht, je weiter er der Gesetzgebung und der Gewissensbildung davonläuft. Wenn der Abstand zu groß wird oder zu lange bestehen bleibt, kann der Fortschritt eine destabilisierende Wirkung haben. Er kann das soziale Gleichgewicht menschlicher Gesellschaften destabilisieren und er kann das ökologische Gleichgewicht auf unserem Planeten destabilisieren, an das sich der Homo sapiens angepasst hat. Die folgerichtige Gegenmaßnahme des Homo sapiens wäre, den Fortschritt zu bremsen. Das aber verbietet der Zwang zum Fortschritt, den sich die Menschen mit der Erfindung des Zinses aufgebürdet haben und der seitdem immer härter

geworden ist und heute fast die ganze Menschheit versklavt. Wo ist der Ausweg aus diesem Teufelskreis? Die Evolution hat ihn bisher nicht gefunden.

Literaturverzeichnis

[BJ82] Bergner, B.; Jungclaussen, H.: Lernverhaltensanalyse mittels Faktorenanalyse und adaptiver Klassifizierung. Wiss. Z. Techn. Univers. Dresden 31 (1982) H. 5.

[He79] Heinold, H. und Autorenkollektiv: Das Lehr- und Dateisystem LEDA. Wiss. Beiträge der Sektion Berufspädagogik der TU Dresden 1979, H. 27.

[Ju01] Jungclaussen, H.: Kausale Informatik. Einführung in die Lehre vom aktiven sprachlichen Modellieren durch Mensch und Computer. Deutscher Universitäts-Verlag, Wiesbaden, 2001; auch unter www.hardwin-jungclaussen.de zu finden.

[UW81] Unger, S; Wysotzki, F.:Lernfähige Klassifikationssysteme, Kap. 12. Akademie-Verlag, Berlin, 1981.

Strukturiertes Programmieren - strukturelles Denken

Mieczysław A. Kłopotek,
Institut für Grundlagen der Informatik der Polnischen Akademie der Wissenschaften,
ul. Ordona 21, 01-237 Warszawa, Polen , klopotek@ipipan.waw.pl

Abstract: In dem Beitrag wird auf die Rolle hingewiesen, die in der Ausbildung der Programmierer an der TU Dresden die Idee der Strukturierten Programmierung hatte. Aus der heutigen Sicht ist das Bestehen auf das Trennen des Programmierungsstils vom Programmierungswerkzeug (Programmierungssprache) als besonders wichtig anzusehen. Auch das Bestreben nach der Integration von Programmentwicklungsmethode und Darstellungsweise des Endprodukts gilt immer noch als moderne Denkweise.

1 Einleitung

Die Programmierung wird oftmals mit der Kunst verglichen. Mit Recht, denn das Entwickeln eines komplizierten Softwareproduktes benötigt jedes Mal einzigartige Ideen und bleibt wohl noch lange eine Herausforderung. Aber auch ein Künstler muss lernen, seine Werkzeuge gut im Griff zu haben, um diese einzigartigen Vorstellungen materialisieren zu können. Und dennoch dürfen die Werkzeuge seine Denk- und Arbeitsweise nicht einschränken, sondern im Gegenteil, eben seine Arbeitsweise sollte ein Ansporn für das Schaffen neuer Werkzeuge sein. Dies galt auch bei der Lehre an der TU Dresden, wo man strukturiertes Programmieren lehrte, ohne dass strukturierte Programmiersprachen verfügbar waren.

In den Jahren 1978-83 war ich Student und dann anschließend Aspirant an der Sektion 08 Informationsverarbeitung der Technischen Universität Dresden. Zu dieser Zeit, also Ende der 70er, Anfang der 80 Jahre des vorigen Jahrhunderts war die Zeit, wo die ersten sog. Höheren Programmiersprachen bereits entwickelt waren - man nenne nur Fortran (50-ger, IBM), Lisp (1958, John McCarthy) C (1972, Dennis Ritchie), Pascal (1970, Niklaus Wirth, Anfang 80er auch an der Technischen Hochschule Karl-Marx-Stadt implementiert), PL/1 (seit 60er Jahre), Algol 60, Algol 68 (das auch an der TU Dresden eine Implementation hatte). Und dennoch galt Assembler (der damaligen ESER-Rechner-Serie, aber auch von Z-80-Mikroprozessoren) als die Grundlage für die praktische Ausbildung eines Programmierers.

Aus der heutigen Sicht scheint es darum bemerkenswert, wie man es damals geschafft hat, uns die noch heute als modern geltenden Software-Entwicklungs-Regeln wie Abstraktion, Dekomposition und Wiederverwendung beizubringen.

Nachfolgend möchte ich kurz die Idee des Strukturierten Programmierens erläutern (Abschnitt 2), um dann ihre Umsetzung in der Lehre (Abschnitt 3), Forschung (Abschnitt 4) an der TU zu erörtern. Ich erlaube mir auch ein paar Worte für und gegen die Struktu-

rierte Programmierung. Insbesondere werde ich darauf hinweisen, wie wir als Studenten den Wert der strukturierten Programmierung sahen (Abschnitt 5). Ich werde kurz einen Beweis für die Strukturierbarkeit eines jeden Programms vorführen (Abschnitt 6) und auf die Schwierigkeiten hinweisen, die man mit Parallelprogrammen dabei hat (Abschnitt 7). Schließlich werde ich kurz darauf eingehen, wie die Idee der Strukturierten Programmierung auf die Darstellung von Wahrscheinlichkeitsverteilungsfunktionen angewandt werden kann (Abschnitt 8).

Insbesondere wurde die strukturierte (oder strukturelle) Programmierung Gegenstand des Ausbildungsprozesses.

2 Die Idee der Strukturierten Programmierung

Das strukturierte Programmieren verkörperte gleichzeitig die Art und Weise, wie das fertige Programm auszusehen hatte, wie es dokumentiert wurde, aber auch wie der Entwicklungsprozess der Software voranschreiten sollte.

Auf der einen Seite bestand also ein strukturiertes Programm aus den elementaren Anweisungen, Prozeduren (oder Funktionen) und Strukturen wie Sequenzen, Alternativen, Zyklen und (auch) parallelen Programmzweigen, die ineinander geschachtelt waren (siehe Abb. 1). .

Auf der anderen Seite war damit direkt eine (top-down) Programmentwicklungsmethode verbunden, die darin bestand, dass die anfangs informal gefasste Aufgabenstellung durch solche Strukturen (also Sequenzen, Alternativen, Zyklen, Parallelzweige, Einschließen in eine Prozedur) verfeinert wurde.

Es wurde dabei von uns abverlangt, dass jedes so verfeinerte Programmstück als auch die Hauptteile des Zielprogramms in den Begriffen von Eingangs-, Ausgangs- und transitiver Information (Parametern), sowie durch eine Kurzfassung der Aufgabe des Programmstücks dokumentiert wurde. Man hat dabei gefordert, dass z.B. in einer Prozedur nicht nur ihre Parameter, sondern auch jegliche globalen Variablen mitbeschrieben wurden, die gelesen oder verändert wurden. Dies betraf nicht nur den Entwicklungsprozess, sondern auch das fertige Programm.

3 Strukturierte Programmierung in der Lehre

Um uns die entsprechende strukturelle Disziplin beizubringen, wurde vom Lehrkörper eine künstliche strukturelle graphische (zweidimensionale) Programmiersprache - PAP („Programmablaufplan“) - entwickelt, die damals noch nirgends implementiert[1] und unsere erste Programmiersprache war, in der wir mit einem Bleistift und Radiergummi unsere Programme im ersten Semester geschrieben haben.

[1] ihre Implementation wurde noch zu meiner Studienzeit Gegenstand einer Dissertation von Dr. Czaja

Von außen her war PAP eigentlich eine Art der noch heute gängigen Flowcharts. Es gab allerdings einen wichtigen Unterschied zu den Flowcharts - die Syntax und Semantik war nach strikten Regeln vorgeschrieben und wurde im Laufe des Semesters erweitert, so dass typische Probleme des Assemblers als auch der höheren Programmiersprachen, und nicht zuletzt (in weiteren Semestern) der Parallelprogrammierung erörtert werden konnten. In dieser Sprache konnte man schließlich nicht nur einfache Datentypen behandeln, sondern auch Strukturen, Zeiger, Listen, dynamische Speicherverwaltung usw.

Das strukturierte Programmieren musste dann von den Studenten in die Praxis umgesetzt werden. Bevor die Assemblerprogramme oder später PL/1-Programme geschrieben wurden, musste jeder den entsprechenden PAP fertig stellen und anhand dessen dann die Implementation durchführen.

4 Die Strukturierte Programmierung in der Forschung

An der Sektion 08 blieb das strukturierte Programmieren nicht in der Welt der Ideen. Wie schon erwähnt, wurde die zweidimensionale PAP-Sprache, mit der Differenzierung der zugelassenen Konstrukte implementiert. Somit konnten die Studenten ihre ersten Programme auch an einem Rechner testen.

Die Idee der strukturellen Programmentwicklung wiederum wurde in Form eines Werkzeugs (geschrieben in Lisp, an dem auch ich als Hilfsassistent arbeitete) zur schrittweise Programmentwicklung materialisert. In einem praxisbezogenen Projekt hat man das Werkzeug auf Entscheidungstabellenentwicklung zugeschnitten, die von der Eisenbahn in Form von Schaltwerken realisiert werden sollten.

5 Der Wert der Strukturierten Programmierung

An dieser Stelle sei daran erinnert, dass der Begriff der strukturierten Programmierung von E. Dijkstra um 1968 [Dij68, Dij82] mit der Zielstellung vorgeschlagen wurde, durch die Berücksichtigung verschiedener methodischer Ansätze die Programmzuverlässigkeit und Wartbarkeit zu verbessern. Dies startete eine lang andauernde Diskussion, ob man auch jedes Programm in der strukturierten Form schreiben kann bzw. soll.

Durch Arbeiten von Bohm und Jacopini[BJ66], Ashcroft und Manna [AM71] sowie Kosaraju [Kos74] und schließlich Mills [Mil75] wurde der Beweis erbracht, dass jedes (nichtparallele) Programm in ein strukturiertes (nichtparalleles) Programm mechanisch umgewandelt werden kann.

Trotzdem enthielten die zuvor als auch danach entwickelte Programmiersprachen unstrukturierte Elemente, die sogenannten GoTo Konstrukte. Zum Beispiel können wir in C schreiben: „Hier: ... goto Hier;“. In FORTRAN 77 gibt es „GOTO Zeilennummer“. In PASCAL kann man schreiben: „If (x<0) then goto Sprung; Writeln(„Positive Zahl„); Sprung;“. ALGOL 60 erlaubt zu schreiben: „if x>q then goto STOP else if x>w-2 then

goto S;“. Demzufolge glaubten und glauben weiterhin viele nicht an Strukturiertes Programmieren.

Die Diskussion um Strukturiertes Programmieren ging an der TU Dresden nicht vorbei. In der Lehre hat man zwar den Studenten die Beweise für „Strukturierbarkeit“ nicht beigebracht, aber statt dessen anhand von konkreten Beispielen die Argumente der Befürworter der „Misch-Masch-Programmierung“ zurückgewiesen. Man behauptete damals, dass diese Art der Programmierung große Einsparungen an Ausführungszeiten und an Programmlänge erlauben würde. Dies widerlegten Studien, die zeigten, dass nur erfahrene Programmierer entsprechende Kürzungen erzielen würden, und dass durch den Einsatz von automatischen Optimierungswerkzeugen für wohl strukturierte Programme eine vergleichbare Zielcodeeffizienz erreicht werden kann, wobei die Programmentwicklungszeit als auch die Fehlerrate entschieden gesunken werden kann.

Als Studenten konnten wir den Beweis der Richtigkeit der These anhand eines Ereignisses erleben, dessen Kenntnis glücklicherweise dem Lehrkörper erspart wurde. Einer der Mitkommilitonen hatte ein Praktikum an einer anderen Sektion, wo eine Bibliothek von Steuerprogrammen entwickelt wurde. Die Grundbibliothek war da, mit einer ziemlich ordentlichen Dokumentation, und seine Aufgabe war es, zusammengesetzte Steuerelemente über dieser Bibliothek zu entwickeln. Die Schwierigkeit bestand darin, dass zwar die Ein- und Ausgangsparameter ordentlich beschrieben wurden, aber die Prozeduren hatten Seiteneffekte, die nicht beschrieben waren. Diese waren für die Einzelanwendung zwar unwichtig, aber als Komponenten waren sie nicht geeignet. Kurzerhand wickelte der Student die Bibliothekprozeduren in kurze Programme ein, die durch Speicherung auf dem Stack (Stapelspeicher) die Seiteneffekte eliminierten. Danach war die eigentliche Aufgabenstellung einfach mit der strukturierten Programmierung zu lösen. Dies gefiel den Auftraggebern nicht, denn nach ihrer Meinung war die Lösung ineffizient (besonders bei Mikrorechnern war damals jedes Byte sehr kostbar). Man wollte ihm vorführen, wie man es richtig anpacken soll. Aber man schaffte esnicht einmal, die einfachste zusammengesetzte Komponente aus der originalen Bibliothek zu fertigen. Man hat sich also mit der strukturierten Programmierung abgefunden. Für uns Studenten war es ein Nachweis, dass das strukturierte Programmieren etwas an sich hat.

Also galt der Grundgedanke, dass die fortschreitenden automatischen Optimierungstechniken für die optimale Codierung verantwortlich sein sollten, weil der Programmierer sich eher auf das korrekte Lösen der gestellten Aufgaben konzentrieren sollte. Das war ein wichtiger Gedanke in vielerlei Hinsicht. Erstens kam da ein wichtiges Prinzip des sich entfaltenden Softwareengineering zum Vorschein: die Werkzeuge sollten dazu dienen, den Menschen zu unterstützen und nicht vice versa. Zweitens machte der Ansatz klar, dass die Fähigkeit und der Stil des Programmierens gar nicht vom verfügbaren Werkzeug (der Programmiersprache) abhängen muss. Drittens war es eine wichtige Einsicht, dass man auf einem viel höheren Niveau als auf dem der Programmiersprache denken muss, um ein Problem richtig zu lösen.

Nicht zuletzt war aber für uns Studenten die Frage eine Herausforderung, ob man auch jedes einzelne Programm in der strukturierten Form aufschreiben kann. Mich faszinierte diese theoretische Frage, und da man uns den Beweis nicht vorgeführt hatte, baute ich eines Tages meinen eigenen kompletten Beweis auf, den ich allerdings nie veröffentlichte, also

auch mit den anderen nie verglichen habe. Stattdessen entwickelte ich ein Makrosystem, welches erlaubte, Assemblerprogramme strukturell zu schreiben, ohne Sprungmarken zu verwenden.

6 Strukturierbarkeit eines Programms

Nun wollen wir aber zu der Frage der Darstellbarkeit eines jeden GoTo-Programms in der Form eines strukturierten Programms, eines PAP, kommen. Ein PAP bestand aus (1) einem START-Knoten, (2) einem ENDE-Knoten, (3) mehreren Ausführungsknoten („einfachen" Anweisungen), (4) Verzweigungsknoten, (5) Zusammenführungsknoten, (6) Parallelanstoß knoten und (7) Paralleltermin ierungsknoten. Das Interessante an einem strukturierten Programm ist das Verschachtelungsprinzip: wenn man eine Sequenz, oder eine Alternative, oder einen Zyklus, oder einen Parallelanstoß nimmt, welche(r) nur aus einfachen Anweisungen besteht, (siehe die Knotenstrukturen der Form aus der Abbildung 1), und diese(n) durch eine einfache Anweisung ersetzt (zu einer einfachen Instrruktion reduziert), so erhält man wieder ein wohlstrukturiertes Programm. Wenn man diese Operation fortfährt, kommt man schließlich zu einem Programm, das nur aus einer einfachen Anweisung besteht. Dies war der Grundgedanke beim Aufbau des Beweises für die Möglichkeit, jedes Programm in eine strukturierte Form umzuwandeln. Es wurde immer ein Stück genommen, welches in eine strukturierte Form gebracht und danach durch einfache Anweisung ersetzt wird. Die Programmstruktur soll sich dabei nach einem Kriterium vereinfachen (z.B. die Anzahl der Zusammenführungen der alternativen Zweige wird verkleinert), so dass der Abschluss des Verfahrens gewährleistet wird.

Das Ziel der Umwandlung eines beliebigen Programmgraphen in ein strukturiertes Programm besteht darin, den Graphen so zu verändern, dass das resultierende Programm genau dieselbe Operation ausführt, aber wohlstrukturiert ist.

Zuerst wollen wir uns den Programmen zuwenden, deren Programmgraphen keine Zyklen und keine Parallelanstöße enthalten, dafür aber Verzweigungen, Zusammenführungen und sequenzielle Anweisungen (siehe Abb. 2). Man kann im Graphen immer den frühesten Zusammenführungsknoten J1 finden. D.h. dass man auf keinem Wege vom Startpunkt zu diesem Zusammenführungsknoten keinen anderen Zusammenführungsknoten findet (siehe 2). Zu ihm gehört immer ein Verzweigungsknoten T1, von dem aus auf zwei verschiedenen Wegen J1 erreicht werden kann und von dessen Nachfolgerverzweigungsknoten dies nicht mehr möglich ist. T3 sei jetzt der späteste Verzweigungsknoten auf dem Weg von T1 zu J1. Um jetzt zu einem strukturierten Programm zu gelangen, verlagern wir diesen Verzweigungspunkt hinter J1. Zu diesem Zweck wird eine zusätzliche logische Variable Z eingeführt, die mit false initiiert wird. Die Zusatzvariable Z wird direkt vor T3 auf den logischen Wert von T3 gesetzt und soll ihn über J1 hinaustragen. Anstelle von T3 setzen wir ein Test des Z-Wertes, wobei hier der Zweig true bis vor J1 geführt wird. Direkt hinter J1 wird wieder der Wert von Z getestet, um jetzt die Abzweigungen von T3 in die originale Richtung zu führen. Jetzt ist T3 aus dem T1-J1-Pfad entfernt und die erste Z-Alternative kann durch eine einfache Anweisung ersetzt werden, denn sie genügt der Forderung der Strukturierten Programmierung und braucht nicht mehr betrachtet zu werden. Auf die Art

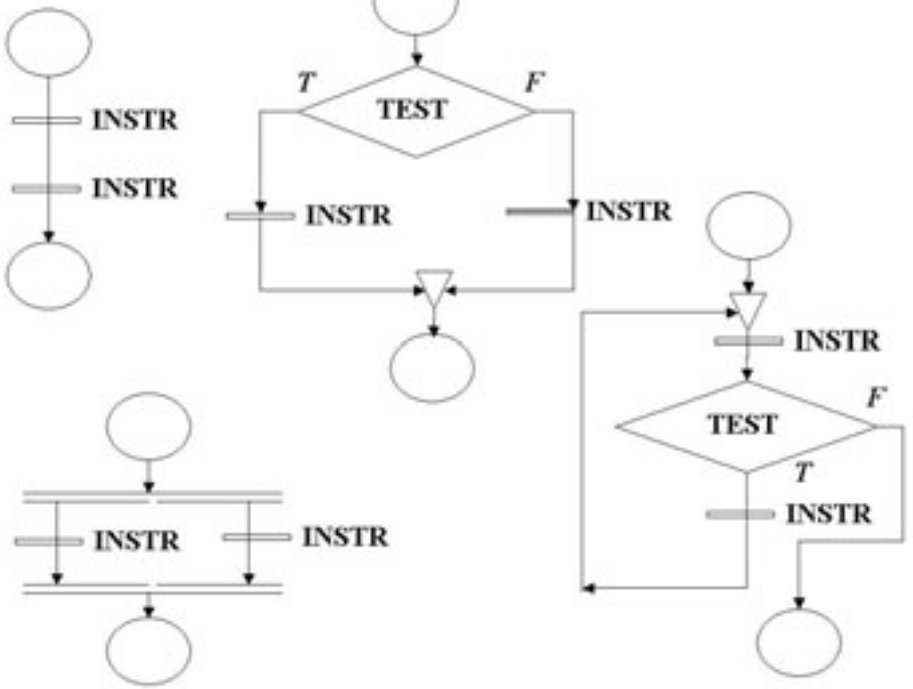


Abbildung 1: Strukturelle Elemente eines Programms

und Weise können alle Verzweigungen zwischen T1 und J1 entfernt werden Danach kann man die T1-J1-Alternative durch eine einfache Anweisung ersetzen und dieses Verfahren wiederholt durchführen, bis alle unstrukturierten Fragmente korrigiert worden sind.

Es sei an dieser Stelle bereits darauf hingewiesen, dass man durch Einführung der Variablen Z die Datenstruktur „kompliziert“, um die Programmstruktur zu vereinfachen.

Des Weiteren sei an der Stelle betont, dass man dasselbe Verfahren anwenden kann, wenn es sich um ein Programm handeln würde, welches keine Zyklen und keine Verzweigungen, dafür aber Parallelanstöße und Parallelterminierungen und sequenzielle Anweisungen enthält. Die eingeführten Hilfsvariablen haben als Aufgabe, in die richtigen Parallelzweige zu springen. Eine Verallgemeinerung auf azyklische Programmgraphen liegt auf der Hand. Man betrachte die Paare Verzweigung/Zusammenführung und Parallelanstoß/Parallelterminierung und dabei jeweils die früheste Terminierung oder Zusammenführung. Es muss jedoch gewährleistet sein, dass auf jedem Pfad vom Anfangsknoten des Graphen die Anzahl der Parallelterminierungen die Anzahl der Parallelanstöße nicht übersteigt.

Und jetzt zum komplizierteren Fall der Zyklen, wobei die Parallelanstöße zunächst unberücksichtigt werden (siehe Abb.3). Identifizieren wir einen Zyklus, wobei der Anfang des Zyklus eine Zusammenführung ist und die letzte diejenige Anweisung, von der aus zur ersten Anweisung des Zyklus übergegangen wird. Zuerst entfernen wir alle in den Zyklus eingehenden Pfeile (alle Zusammenführungspunkte außer dem ersten des Zyklus). Wir eliminieren zuerst den frühesten unerwünschten Eintrittspunkt. Eine Hilfsvariable Z wird eingeführt, die direkt vor dem Zyklus zu true initiiert wird und auf dem zu beseitigenden Pfeil zu false. Jetzt verlegen wir den unerwünschten Eintrittspunkt vor den Zyklus. Durch

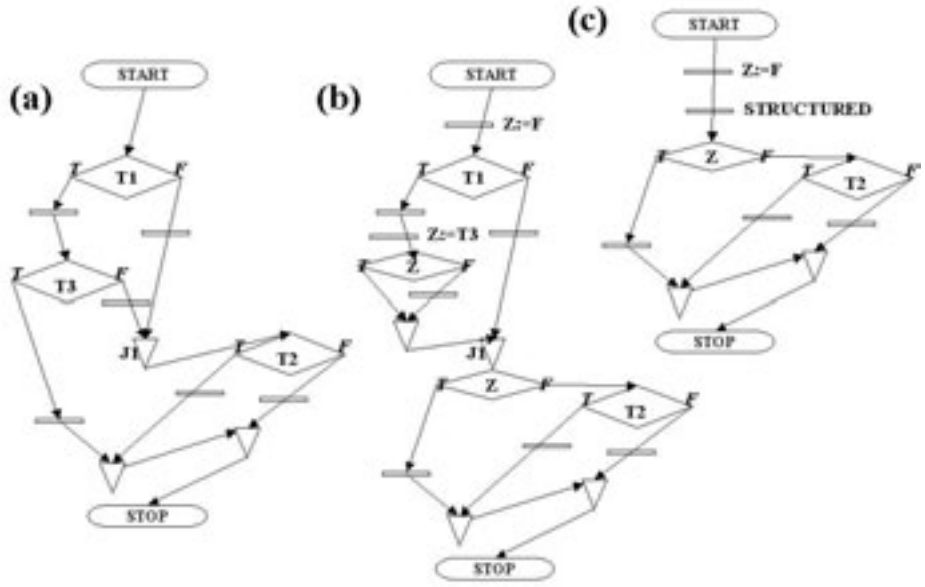


Abbildung 2: Behandlung zyklenfreier Programme

einen Test auf Z direkt nach dem Eintritt in den Zyklus verhindern wir die Ausführung der einfachen Anweisungen bis zu dem einstigen unerwünschten Eintrittspunkt. Auf diese Weise werden alle unerwünschten Eintrittspunkte entfernt. Ähnlich verlegen wir alle Austrittsverzweigungen bis zum letzten Austrittspunkt aus dem Zyklus.

Danach der Zyklus ist „sauber“ und kann durch eine einfache Anweisung ersetzt werden. In diesem Prozess wird die Anzahl der Zusammenführungen um 1 reduziert, so dass das Entfernen aller Zyklen garantiert ist. Jeder Graph lässt sich in einen zyklenfreien Graphen überführen, indem man einige Bögen (Pfeile) entfernt. Wir behandeln zuerst den Zyklus, der durch Rückerstellung des Bogens entsteht, der als „letzter“ unter den entfernten Bögen eingeht. Unerwünschte Eintrittspunkte sind die Kanten, die man in dem azyklischen Graphen vom Startpunkt auf dem Wege zum Endpunkt des Zyklus nicht durchquert, die aber in einen Knoten eingehen, der sich auf einem solchen Wege durchqueren lässt. Unerwünschte Austrittspunkte sind die Kanten, die man in dem azyklischen Graphen vom Startpunkt auf dem Wege zum Endpunkt des Zyklus nicht durchquert, die aber aus einen Knoten ausgehen, der sich auf einem solchen Wege durchqueren lässt.

Diese Vorangehensweise lässt sich auf Parallelanstöße und Parallelterminierungen ausdehnen, allerdings mit gewissen Einschränkungen. Erstens können bei der Erstellung eines azyklischen Graphen nur Bögen entfernt werden, welche in Zusammenführungen eingehen (und nicht etwa Terminrationen). Zweitens muss sich auf die Weise auch ein azyklischer Graph schaffen lassen. Drittens muss in dem azyklischen Graphen gewährleistet sein, dass auf jedem Pfad vom Anfangsknoten des Graphen die Anzahl der Parallelterminierungen die Anzahl der Parallelanstöße nicht übersteigt. Die Gründe dafür sind trivial: in einem wohlstrukturierten PAP kann man unendlich viele parallele Prozesse aus einer endlichen Struktur nicht generieren, und darum lassen sich nur solche Programmgraphen in eine gute Struktur umwandeln, die solche Prozesse nicht generieren können.

Damit sind wir also in der Lage, jedes Programm in ein strukturiertes Programm zu überführen. Allerdings ist darauf zu achten, dass man zusätzliche Variablen einsetzen muss, was als eine Konsequenz die Verlagerung der Programmstruktur auf die Datenstruktur bedeutet. Demzufolge ist eine so formal abgeleitete Struktur des Programms nicht unbedingt besser lesbar als das Original. Der Beweis hat hier eher die Bedeutung einer Anschauung, dass es keinen zwingenden Grund gibt, „GoTos“ einzuführen.

Nebenbei bemerkt ist die Beweisführung auch ein Hinweis darauf, wie man hierarchisch Sprungmarken für Anweisungen automatisch vergeben kann, so dass man sich auch in der Assemblerprogrammierung nicht besonders anstrengen muss, wenn man ein wohlstrukturiertes Programm in Code umsetzen will. Dies lag dem von mir erwähnten Makrosystem für Assembler zugrunde, in dem gänzlich auf Sprungmarken der Programmierer verzichten werden konnte, weil diese von den Makros automatisch vergeben wurden. Hierzu ist vorerst zu vermerken, dass „wilde Springe“ auch in höheren Programmiersprachen (lineares Rückgrat plus GoTos) Verwendung von Sprungmarken verlangen. Die strukturierte Programmierung aber kann in höheren Sprachen ohne Sprungmarken auskommen. Man verdankt dies der besonderen Graphstruktur, die sich auf eine besondere Weise linearisieren lässt.

7 Die Strukturierte Programmierung und Parallelverarbeitung

Die wohl wichtigste Eigenschaft des strukturierten Programmierens war, dass man sich anhand der Beschreibung eines Programmstücks sofort ein Bild über seine Einsetzbarkeit in einem größeren Programm machen konnte. Und wenn man sich ein algorithmisches Bild des Codes im Programmentwurf machen wollte konnte man das anhand der Unterkomponenten machen, ohne den Code zu durchwühlen. Hierfür war die Disziplin der Beschreibung der Ein- und Ausgangsvariablen von einer vorrangigen Rolle: Falls ein Zustand der Eingangsgrößen vorlag, konnte man den Ausgangszustand der Ausgangsgrößen aus der Beschreibung ablesen.

Problematisch war allerdings die Konstruktion der parallelen Verarbeitung. Man musste auf die interne Disziplin zurückgreifen, dass in den parallelen Zweigen nie dieselben Variablen verwendet wurden, bzw. dass man die Verwendung von gemeinsamen Variablen richtig bedenkt. Natürlich wurden die Synchronisationswerkzeuge, die auch heute angewandt werden, wie Semaphore, kritische Regionen oder Monitore auch gelehrt. Dies führte in die Welt der „Deadlocks“, die wieder durch „Struktur“ (Reihenfolge) der Ressourcenanforderungen (obgleich nicht effizient) gelöst werden konnten, was Gegenstand meiner Forschung wurde [Kło84, HK86].

Es muss hier noch ergänzt werden, dass die Parallelität der Prozesse nicht nur die Konkurrenz um Ressourcen, sondern auch Kooperation verlangen kann. Bei der vorhergehenden Diskussion über die Umwandlung in die strukturierte Form könnte man synchrone Kooperation in Form von Parallelterminationen mit anschließendem Parallelanstoß darstellen, während für die asynchrone Kooperation die Zwischenschaltung eines Prozesses ausreicht, der in einem Prozess angestoßen und in einem anderen zusammengeführt wird.

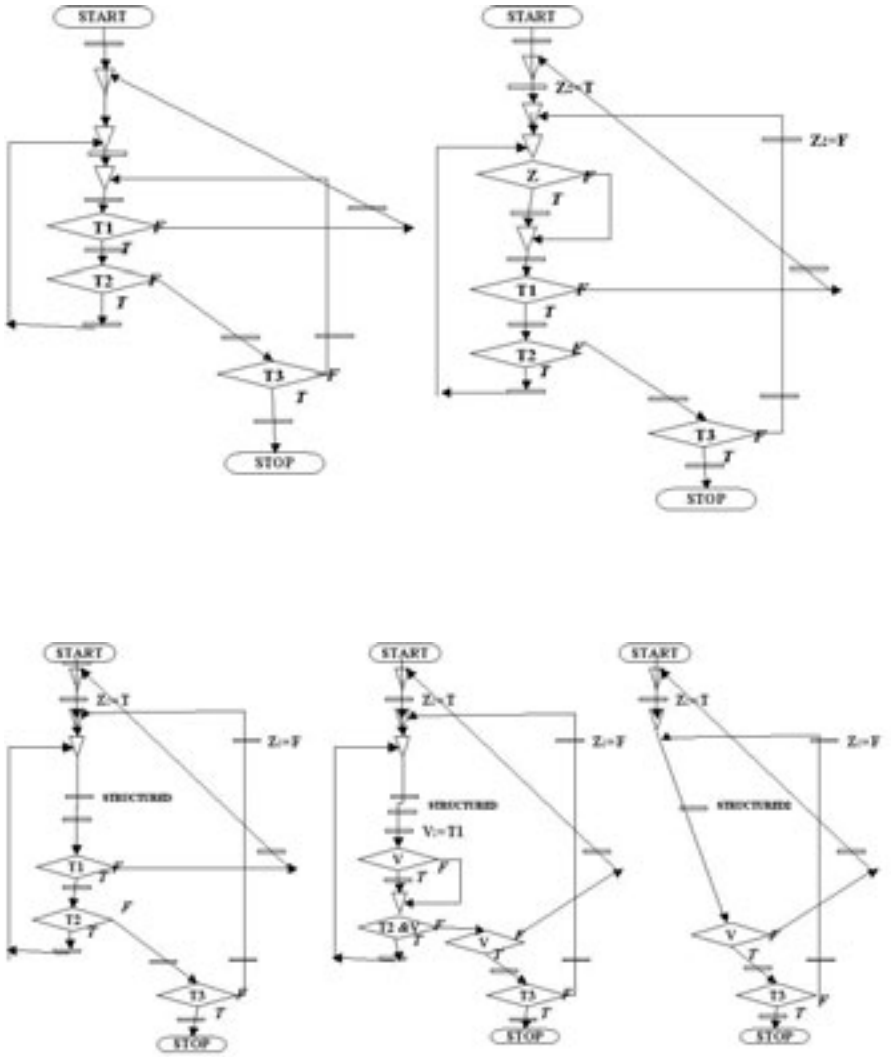


Abbildung 3: Behandlung der Zyklen

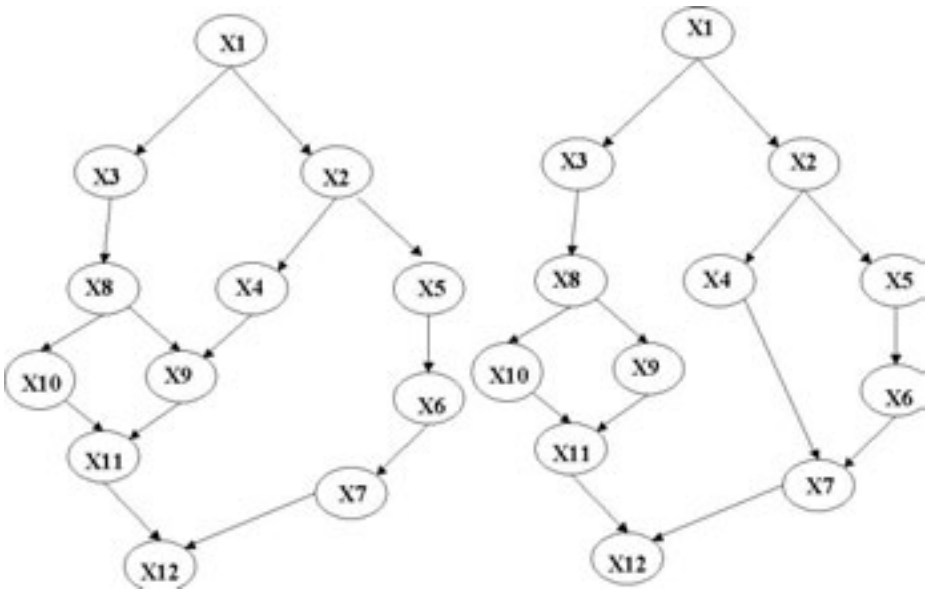


Abbildung 4: Allgemeine (links) und strukturierte (rechts) Bayesche Netzwerke

Natürlich ist der PAP-Parallelismus heute nicht die einzige und auch damals gekannte Form der Parallelverarbeitung. Ein Bestandteil der Lehre waren auch die Petri-Netze in verschiedenen Formen. Algorithmen mit massiver Parallelität (single instruction stream, multiple data stream) wurden auch in der Lehre angeschnitten.

8 Die Strukturierte Programmierung und andere Graphstrukturen

Überraschenderweise griff ich nach Jahren auf den Gedanken des Strukturierbarkeitsbeweises zurück, als ich an der Inferenz in den sogenannten Bayesschen Netzwerken arbeitete.

Die Bayesschen Netzwerke [Jen96] bieten eine kompakte Darstellung einer Wahrscheinlichkeitsverteilungsfunktion in mehreren Variablen, indem die relative Unabhängigkeit der Variablen ausgenutzt wird. .

Eine beliebige Wahrscheinlichkeitsverteilungsfunktion in diskreten Variablen kann man als

$$P(x_1, \ldots, x_n) = \prod_{i=1,\ldots,n} P(x_i | x_1, \ldots, x_{i-1})) \tag{1}$$

darstellen.

Falls wir wissen, dass X_i von manchen Variablen von den $X_1, \ldots, X_{i-1}$ (statistisch) nicht

direkt abhängt, vereinfacht sich die Struktur zu:

$$P(x_1, \ldots, x_n) = \prod_{i=1,\ldots,n} P(x_i|\pi(x_i)) \tag{2}$$

wo $\pi(X_i)$ den Satz der „Eltern“ der Variable X_i darstellen, d.h. der Variablen, von denen X_i direkt abhängt. Symbolisch wird diese Formel in der graphischen Form eines Orientierten Azyklischen Graphen (dag) dargestellt (Abb.4). Die Knoten stehen hier für Variablen und die Kanten verbinden die Eltern mit der von ihnen direkt abhängenden Variable.

Das Bayessche Netzwerk stellt zwar eine Wahrscheinlichkeitsverteilungsfunktion kompakt dar, aber trotzdem ist die Inferenz schwierig. Nur die graphischen Strukturen eines Baumes oder Polybaumes erlauben eine effiziente Inferenz.

Ich überlegte, welche Voraussetzungen nötig sind, um die Klasse der Bayesischen Netzwerke, für die die effiziente Inferenz relativ einfach ist, über die Baum- und Polybaumstrukturen hinaus auszudehnen. Bekanntlich muss man ein Bayessches Netzwerk in den sogenannten Markovschen Baum umwandeln, in dem die Inferenz einfach ist, aber die Umwandlung selbst ist NP-komplex (mit Ausnahme eben der Bäume und Polybäume). Das Charakteristische an der Inferenz im Markovschen Baum ist die „Reduktion“ der Baumstruktur im Laufe der Vermittlung der sogenannten Nachrichten. Das erinnerte an die Eigenschaft der wohlstrukturierten Programme.

So definierte ich strukturierte Bayesische Netzwerke nach dem Vorbild der strukturierten Programme [Kło03b], und bewies (nach dem Vorbild der strukturierten Programmierung), dass sich jedes Bayesische Netzwerk in einer strukturierten Form darstellen lässt [Kło04] Ich entwickelte auch Methoden, wie man solche strukturierte Bayesschen Netzwerke aus Daten gewinnt [Kło03a].

9 Abschließende Bemerkungen

Am Ende muss man noch einmal auf den wichtigen Unterschied zwischen der automatisch aus einem beliebigen Graphen erzeugten strukturierten Form des Programms oder Bayesschen Netzwerkes und der menschlischen Entwicklung des wohlstrukturierten Graphen hinweisen. Die Komplexität der inneren Struktur wird nämlich von dem Automaten aus der Programm- oder Netzwerkstruktur auf die Datenstrukturen verschoben. Durch die automatische Strukturierung wird also das Programm nicht übersichtlicher.

Aber das Komplexitätsphänomen ist auch bei menschlich erstellten Programmen ein Problem. Dies war bereits damals sichtbar, als man in das einheitliche Konzept der strukturierten Programmierung auch die Parallelverarbeitung integrieren wollte und die Synchronisation außer dem allgemeinen Rahmen fiel. Dieses Problem bleibt übrigens bis heute ungelöst. So bleibt also immer noch die wichtige Frage für die Forscher, wie man durchschaubare Strukturen sowohl für die Programme als auch für die Daten schaffen kann, in denen der Arbeitsaufwand des Programmierers in Grenzen gehalten wird. Einer der Gutachter dieses Beitrages hat darauf hingewiesen, dass Software-Objekte und

-Komponenten als die Mittel angesehen werden können, parallele Programmstrukturen zu finden, die den Arbeitsaufwand eines Programmierers bei der Entwicklung von parallelen Programmstrukturen in Grenzen halten. Ich stimme dieser Meinung zu, allerdings mit dem Vorbehalt, der dem bekannten Buch von S. Wolfram *The new kind of science* entnommen werden kann, dass die von dem Menschen beherrschten konzeptuellen Strukturen sehr klein im Vergleich zu den möglichen sind, wodurch auch das vom Menschen Schaffbare beschränkt ist. Es müssen neue strukturierte Formen auf ihre Brauchbarkeit und Beherrschbarkeit experimentell überprüft werden, wenn der technische Fortschritt voranschreiten soll. Und dies gilt auch für die Programmierung. Und dennoch (oder gerade deswegen) wird das Prinzip der Zusammensetzung, welches der Strukturierten Programmierung zugrunde liegt, noch lange ein wichtiger Bestandteil des Softwarengineering als auch der Forschung und Lehre bleiben.

Literatur

[AM71] Edward A. Ashcroft und Zohar Manna. The Translation of 'Go To' Programs to 'While' Programs. IFIP Congress (1) 1971: 250-255, 1971.

[BJ66] C. Bohm und G. Jacopini. Flow Diagrams, Turing Machines, and Languages with Only Two Formation Rules. Communications of the ACM 9:5, p. 266, May 1966., 1966.

[Dij68] E. Dijkstra. goto Statement Considered Harmful. Communications of the ACM 11:3, p. 147, March 1968., 1968.

[Dij82] E. Dijkstra. Stepwise Program Construction. Printed in Selected Writings on Computing: A Personal Perspective, Springer-Verlag, 1982, p. 2., 1982.

[HK86] O. Herrlich und M. A. Kłopotek. Hierarchische Betriebsmittelvergabe. Rechentechnik/Datenverarbeitung 23(1986)10, pp. 30-33., 1986.

[Jen96] J. Jensen. An Introduction to Bayesian Networks, Springer Verlag, 1996.

[Kło84] M. A. Kłopotek. Ein Beitrag zur statischen Verklemmungsvermeidung in Systemen nichtbinärer Betriebsmittel. Dissertationsarbeit, Technische Universität Dresden, Sektion Informationsverarbeitung, 1984.

[Kło03a] M. A. Kłopotek. Reasoning and Learning in Extended Structured Bayesian Networks. Fundamenta Informaticae. 58(2)2003, pp.105-137, 2003.

[Kło03b] M. A. Kłopotek. Reasoning in Structured Bayesian Networks IN Rutkowski, L., Kacprzyk, J., (Eds.) *Neural Networks and Soft Computing*. Sixth International Conference on Neural Network and Soft Computing, Zakopane, Poland, June 11-15, 2002, Advances in Soft Computing. Springer-Verlag, ISBN 3-7908-0005-8, pp. 418-423, 2003.

[Kło04] M. A. Kłopotek. Impact of Structuring on Bayesian Network Learning and Reasoning in:P. Grzegorzewski, M. Krawczak, S. Zadrozny eds.: Soft Computing Tools, Techniques and Applications. Akademicka Oficyna Wydawnicza EXIT Warszawa. pp.77-118, 2004.

[Kos74] S R Kosaraju. Analysis of structured programs J Comptr Syst Sct 9, 3 (1974), 232-254, 1974.

[Mil75] H. D. Mills. The New Math of Computer Programming. Com.ACM Jan. 1975, vol.18, Number 1, pp.43, 1975.

Lehrerbildung Informatik – Basis für die Informatik als Allgemeinbildung

Steffen Friedrich Bettina Timmermann

Technische Universität Dresden
Fakultät Informatik
AG Didaktik der Informatik/Lehrerbildung
01062 Dresden
Steffen.Friedrich@tu-dresden.de Bettina.Timmermann@tu-dresden.de

Abstract: Betrachtet man das heutige Mühen um informatische Bildung für alle Schüler unter historischer Sicht, dann muss man feststellen, dass seit der Etablierung von Kursen in der gymnasialen Oberstufe sich zwar Wandlungen in den didaktischen Konzeptionen vollzogen haben, aber ein Durchbruch zu einem Pflichtfach nur teilweise realisiert werden konnte. Neben Gründen, die eher bildungspolitischen Handlungsfeldern zuzuordnen sind, gab und gibt es in sehr vielen Bundesländern deutliche Defizite in der Aus- und Weiterbildung von Lehrern für das Fach Informatik. Erfahrungen aus der Lehrerausbildung vor 1990 in der DDR und deren Überführung in die neuen Strukturen zeigen heute, dass eine systematische und flächendeckende Lehrerbildung ein wesentlicher Schlüssel für den Erfolg der informatischen Bildung in der Schule darstellt.

1 Informatische Bildung in Schulen

1.1 Einiges über die Anfänge des Informatikunterrichts

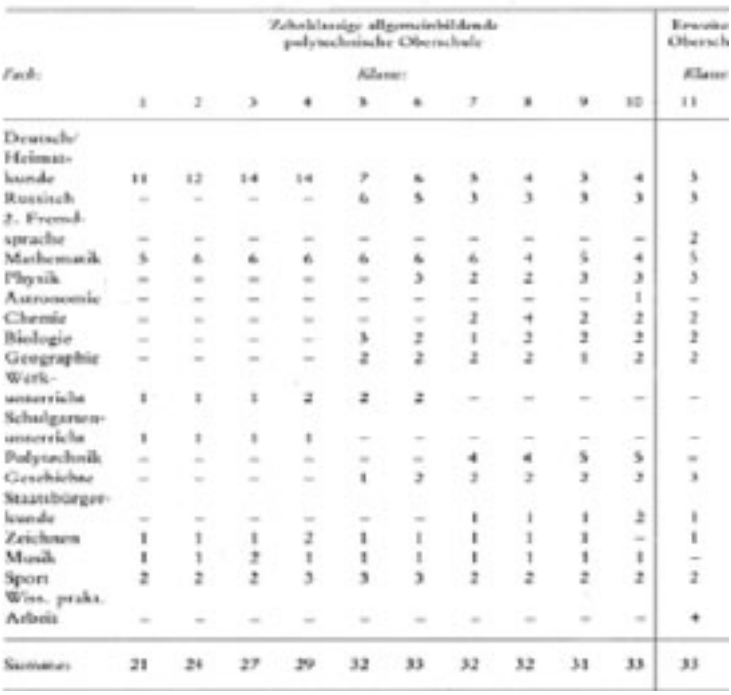
Die Anfänge informatischer Bildung in Schulen in Deutschland sind nur wenig später zu datieren, als die Einführung von Studiengängen zur Informatik bzw. zur Informationsverarbeitung in Universitäten und Hochschulen. Folglich finden sich diese Ansätze zuerst in der gymnasialen Oberstufe, auch weil zu diesem Zeitpunkt in den westlichen Bundesländern gerade eine Umorganisation der Kursstufe an den Gymnasien oder vergleichbaren Einrichtungen stattfand. In der DDR gab es diese Anfänge erst etwa 10 Jahre später, vor allem wegen der fehlenden technischen Voraussetzungen. Das didaktische Konzept damals ist einer „Abbild-Didaktik“ zuzuordnen, das zusätzlich das Programmieren in einer höheren Programmiersprache besonders betonte [vgl. Ko98]. Es schien, dass dies die Kernkompetenz zur Beherrschung informationsverarbeitender Maschinen sein wird und jeder damit konfrontiert ist. Bis zum Jahre 2004 waren dann auch Grundlagen der Programmierung und der Softwaretechnologie zentraler Bestandteil in den von der KMK erlassenen Prüfungsanforderungen des Faches Informatik für das

Abitur [vgl. EP89]. Mit deren Neufassung (2004) wurden schließlich moderne Aspekte berücksichtigt, ohne auf diese algorithmischen Grundlagen zu verzichten.

Mit der Einführung grafischer Benutzeroberflächen und vereinfachter Anwendungen für Alltagstätigkeiten, wie beispielsweise Textverarbeitung oder Rechenblätter, erwuchsen neue Anforderungen für die Bildung auf diesem Sektor. Die Beherrschung von aktuellen Produkten und deren Verwendung beim Lehren und Lernen trat in den Vordergrund. Auf der Basis von Festlegungen der Bund-Länder-Kommission zur informationstechnischen Grundbildung wurde sicher ein wichtiger Schritt zur Nutzung des Computers in den verschiedenen Bereichen der Allgemeinbildung gegangen. Es bestand vielerorts Konsens darüber, dass es notwendig war, für ein Grundverständnis der Computerbenutzung zu sorgen. Vielleicht hat man damals zu wenig zur Kenntnis genommen, dass die jeweiligen Anwendungen schnell veralten und damit eine Bedienung der im Moment aktuellen Software, die nach dem Ende der Schulzeit in der Regel so gar nicht mehr existiert, eben gerade kein ausreichender Inhalt einer informatischen Bildung sein kann. Es ist sicher ein Vorteil, wenn mit verschiedenen Anwendungen der Arbeitsprozess (auch das Lehren und Lernen) unterstützt wird. Aus didaktischer Sicht wurde die Ansicht vertreten, dass integrative Ansätze geeigneter sind, die Universalität des Computers in unterschiedlichen fachlichen Kontexten deutlich zu machen. Sie verdecken aber sowohl die für den richtigen Einsatz eines solch mächtigen Werkzeugs erforderlichen Grundlagen als auch die dafür in den Schulen oft nicht vorhandenen Rahmenbedingungen. Es fehlen auch hier die notwendigen Grundlagen, um Wirkprinzipien zu verstehen und künftige Entwicklungen einordnen zu können. Letztlich fehlen die Ressourcen, um in Produktschulungen den Lehrern der verschiedenen Fächer die Änderungen in den aktuellen Produkten zu vermitteln. Später waren es multimediale Oberflächen und an Schulen vorhandene Internetzugänge, die den Eindruck erweckten, dass die Nutzung neuer technischer Möglichkeiten in der Bildung lediglich eine Frage der modernen Ausstattung sei. Trotz kritischer Stimmen und konstruktiver Vorschläge [vgl. Ne97] wurden in der Initiative „Schulen ans Netz“ auf integrative Nutzungen im Unterricht, Fortbildung der Fachlehrer und fachübergreifende Projekte orientiert.

1.2 Zur Situation in der DDR

In den achtziger Jahren besuchte der überwiegende Teil der Kinder in der DDR die zehnklassige polytechnische Oberschule (POS) und erwarb mit dem erfolgreichen Ablegen der Abschlussprüfungen das Zeugnis der „Mittleren Reife“. Ein eher geringer Prozentsatz (unter 20 %) jedes Jahrgangs besuchte anschließend die erweiterte Oberschule (EOS) und legte nach zwei weiteren Unterrichtsjahren das Abitur ab. Anhand der Stundentafel wird deutlich, dass in den beiden Klassenstufen der EOS von den 33 Unterrichtsstunden pro Woche mehr als 50 % für den mathematisch-naturwissenschaftlichen Unterricht zur Verfügung standen. Im Zentrum heutiger Forderungen, beispielsweise des Verbandes für mathematisch-naturwissenschaftlichen Unterricht (MNU), steht die Sicherung von wenigstens einem Drittel der vorhandenen Unterrichtszeit für die MINT-Fächer (das sind Mathematik, Naturwissenschaften und Informatik) für alle Schüler. Einen Überblick vor Einführung des Informatikunterrichts als Pflichtfach an den EOS (1988/89) gibt die nachfolgende Stundentafel [Bu05]:



Fach:	Zehnklassige allgemeinbildende polytechnische Oberschule										Erweiterte Oberschule
	Klasse:										Klasse:
	1	2	3	4	5	6	7	8	9	10	11
Deutsche/ Heimatkunde	11	12	14	14	7	6	5	4	3	4	3
Russisch	–	–	–	–	6	5	3	3	3	3	3
2. Fremdsprache	–	–	–	–	–	–	–	–	–	–	2
Mathematik	5	6	6	6	6	6	6	4	5	4	5
Physik	–	–	–	–	–	3	2	2	3	3	3
Astronomie	–	–	–	–	–	–	–	–	–	1	–
Chemie	–	–	–	–	–	–	2	4	2	2	2
Biologie	–	–	–	–	3	2	1	2	2	2	2
Geographie	–	–	–	–	2	2	2	2	1	2	2
Werkunterricht	1	1	1	2	2	2	–	–	–	–	–
Schulgartenunterricht	1	1	1	1	–	–	–	–	–	–	–
Polytechnik	–	–	–	–	–	–	4	4	5	5	–
Geschichte	–	–	–	–	1	2	2	2	2	2	3
Staatsbürgerkunde	–	–	–	–	–	–	1	1	1	2	1
Zeichnen	1	1	1	2	1	1	1	1	1	–	1
Musik	1	1	2	1	1	1	1	1	1	1	–
Sport	2	2	2	3	3	3	2	2	2	2	2
Wiss. prakt. Arbeit	–	–	–	–	–	–	–	–	–	–	4
Stundensumme:	21	24	27	29	32	33	32	32	31	33	33

Abbildung 1: Stundentafel POS / EOS

Eine intensivere Ausbildung in diesem Bereich gab es an den Spezialschulen mit mathematisch-naturwissenschaftlicher Orientierung. In diese Gymnasien wurden die Schüler im Unterschied zur EOS schon früher (meist ab Klassenstufe 9) aufgenommen und bis zum Abitur geführt. Diese Spezialschulen hatten jeweils eigene Stundentafeln, die entsprechend der Spezialisierung sich von denen der erweiterten Oberschule unterschieden und durch entsprechende technische Ausstattung bereits frühzeitig (teilweise ab Klassenstufe 9) auch den Unterricht im Fach Informatik durchführen konnten.

Ferner spielte auch die Vermittlung von Kenntnissen in der Elektronik oder zum Programmieren in vielen Arbeitsgemeinschaften und anderen Einrichtungen eine Rolle, wie beispielsweise in dem in Dresden im Jahr 1984 in Zusammenarbeit mit dem Kombinat Robotron gegründeten „Schülerrechenzentrum (SRZ)“. Das SRZ wird heute unter dem Dach der TU Dresden in enger Zusammenarbeit mit der Stadt, dem Sächsischen Staatsministerium für Kultus und der Wirtschaft fortgeführt. Damit ist eine Tradition in der Begabtenförderung in den Fachgebieten Informatik und Elektronik bewahrt, die bundesweit einmalig ist. [vgl. He07]

Die seit 1985 in der DDR vorhandenen Kleinstrechner, mit denen auch Schulen (meist ein Computerkabinett) ausgestattet wurden, verstärkten das Bedürfnis nach einem fachsystematischen Unterricht, der nach damaligen didaktischen Konzeptionen eher einem algorithmenorientierten Vorgehen zuzuordnen ist. Das entsprach durchaus den Inhalten

von Lehrplänen der gymnasialen Oberstufe in den westlichen Bundesländern. Im Folgenden zwei Beispiele:

> **Lehrplan Berlin:** [LB85]
> In der Algorithmik steht die Einführung in grundlegende Elemente der Algorithmisierungsmethodik an Hand kleiner Programme im Vordergrund.
>
> **Lehrplan DDR:** [LD89]
> Die Schüler kennen den Begriff ‚Algorithmus' und wesentliche Eigenschaften von Algorithmen. Sie besitzen sichere Kenntnisse über Algorithmenstrukturen (...) und wesentliche Möglichkeiten ihrer Darstellung, ...

Nach entsprechenden Vorbereitungen und Erprobungen (einheitlicher Lehrplan, unterstützende Unterrichtsmaterialien für Lehrer und Schüler) wurde mit Beginn des Schuljahres 1988/89 in allen allgemeinbildenden erweiterten Oberschulen das Fach Informatik mit jeweils 2 Wochenstunden in den Jahrgangstufen 11 und 12 eingeführt.

> *Da es keine exakten Quellen mehr gibt, die die Veränderungen in der Stundentafel belegen, kann nur auf Aussagen von Lehrern zurückgegriffen werden, die in alten Klassenbüchern gestöbert haben:*
>
> *Im Zusammenhang mit der Einführung des Fachs Informatik für die 11. Klassen 1988/89 in der DDR wurde eine Stunde Geographie gestrichen und außerdem die Gesamtstundenanzahl um eine Stunde erhöht. Außerdem wanderte eine Stunde Geschichte [bisher 3h] zu Staatsbürgerkunde [bisher 1h]. Auf jeden Fall wurde keine der Stunden vom Fach Mathematik genommen, es blieb dort konstant (5 h).*
> *Die sonstige Stundenverteilung in der Klassenstufe 11 ist in der obigen Tabelle ersichtlich. Für die Klassenstufe 12 waren keine Daten auffindbar.*

Fast gleichzeitig wurden die vorgegebenen Lehrplanforderungen in Schülermaterialien und in Handreichungen für Lehrer umgesetzt. [vgl. LM]

1.3 Gegenwärtiger Stand – Bildungsstandards Informatik

Im sich gegenwärtig sehr rasch vollziehenden Wandlungsprozess zu einer wie auch immer beschriebenen Informations- und Wissensgesellschaft sind entsprechende Kompetenzen zu erwerben. Das macht eine informatische Bildung und darin enthalten den Informatikunterricht als Grundbestandteil der Allgemeinbildung erst recht notwendig, wobei durch die Verantwortung der Bundesländer dafür das Bild in Deutschland sehr unterschiedlich ist. [vgl. Ti08] Hier sollen Bildungsstandards ansetzen, eine Gesamtsicht auf eine informatische Bildung liefern und gleichzeitig die Komplexität in diesem Bereich deutlich machen. Die informatische Bildung hat längst den Ruf abgelegt, der Technologie nachzujagen, sondern beinhaltet genügend Grundlagen, die zur minimalen Bildung eines jeden Lernenden im heutigen gesellschaftlichen Umfeld gehören. Die Erfahrungen des Wirkens für einen verpflichtenden Informatikunterricht zeigen, dass dazu ein Spagat zwischen Wünschenswertem und Machbarem nötig ist.

Die Folgerung aus den bisherigen Darstellungen zum Stand der informatischen Bildung konnte eigentlich nur darauf gerichtet sein, unter Beteiligung vieler Fachleute aus Uni-

versitäten und Schulen solche Kompetenzen zu beschreiben und für Lehrende nutzbar darzustellen, die das Minimum informatischer Bildung für alle Schüler erfassen, also Mindeststandards zu formulieren. In diesem Sinne wurden in den inzwischen veröffentlichten Bildungsstandards Informatik [vgl. BI07] die Inhaltsbereiche „Information und Daten“, „Informatiksysteme“, „Algorithmen“, „Sprachen und Automaten“ sowie „Informatik, Mensch und Gesellschaft“ herausgearbeitet und Kompetenzen formuliert, die Schülerinnen und Schüler erwerben müssen. Diese Kompetenzen sind als Unterstützung für die Planung und Gestaltung des Unterrichts für die Klassenstufen 5-7 und 8-10 konkretisiert. In analoger Weise wurden informatikspezifische Handlungen in Form von Prozesskompetenzen zusammengestellt.

Bei der Formulierung von Standards für die Schulinformatik ist es natürlich auch notwendig, Inhaltsbereiche und prozessbezogene Kompetenzen in Einklang zu bringen. Eine überzogene Orientierung auf Ergebnisse (vielleicht typischer Aufgaben) bewirkt zusätzlich, dass Unterricht zur Testvorbereitung mutiert und auf eine automatische Ausprägung von Kompetenzen hoffend, die Lösung von Standardaufgaben trainiert wird. In Schulen sind deshalb der „Computerführerschein“ oder ähnliche Schulungen, die das Beherrschen aktueller Produkte testen, für die Ausprägung von Kompetenzen im Zuge einer informatischen Bildung ungeeignet. Außerdem zeigen die vorliegenden Ausarbeitungen zu den Standards erneut, dass informatische Bildung ins mittlere Schulalter gehört.

2. Zur Aus- und Weiterbildung von Informatiklehrern

2.1 Konzepte und Rahmenbedingungen in der DDR

Die Ausbildung von Lehrern erfolgte in der DDR sowohl an Universitäten als auch an Pädagogischen Hochschulen. Generell war für jede Einrichtung jeweils eine sehr begrenzte Anzahl von Fächerkombinationen festgelegt. So wurden zum Beispiel an der Pädagogischen Hochschule Dresden Mathematiklehrer nur in der Fachkombination Mathematik/Physik oder Mathematik/Geografie ausgebildet. An anderen Einrichtungen war Mathematik auch mit anderen Fächern kombinierbar, aber wieder nur mit maximal zwei verschiedenen Fächern. Die Studienpläne waren zentral abgestimmt und boten wenig Raum für Spezifizierungen. Nach einem fünfjährigen Studium mit zahlreichen Schulpraktika - insbesondere fand das letzte Studienjahr fast vollständig in der Schulpraxis statt - verließen die Studenten die Einrichtung mit einem Diplom für alle Schularten und wurden sofort als Lehrer in den zugewiesenen Schulen eingesetzt.

Die Ausbildung von Diplomlehrern der allgemeinbildenden polytechnischen Oberschulen in der Fachkombination Mathematik/Informatik an Universitäten und Hochschulen wurde im Zuge der Einführung des Unterrichtsfaches Informatik an den EOS und von Aktualisierungen im Rahmen des polytechnischen Unterrichts kurz vor der politischen Wende aufgenommen. Allerdings wurde die Einführung dieses Studienganges nur der TU Chemnitz und der PH Güstrow gestattet. Der für diese Studienrichtung vom Ministerium für Volksbildung und vom Ministerium für Hoch- und Fachschulen bestätigte Stu-

dienplan trat zum 1.1.1989 in Kraft [SP89]. Nach einer kurzen Formulierung des Erziehungs- und Ausbildungsziels sind in diesem Studienplan die Inhalte aller Bestandteile des Studiums beschrieben:

Das inhaltliche Konzept der Fachausbildung Informatik war auf der Grundlage der für den künftigen Informatikunterricht notwendigen Voraussetzungen entstanden und unterschied sich in verschiedenen Punkten von damaligen Studiengängen der Diplominformatiker. Das soll an der Darstellung einzelner Gebiete gezeigt werden:

> *„Im Lehrgebiet **Grundkurs Informatik** erwerben die Studenten grundlegende fachwissenschaftliche Kenntnisse, Fähigkeiten und Fertigkeiten auf den Gebieten Programmierungstechnik und Software-Technologie, Rechnersysteme, Betriebssysteme, Theoretische Informatik und Computergeometrie. Sie eignen sich die für die Informatik typischen Denk- und Arbeitsweisen systematisch an und werden mit der informationsverarbeitenden Technik vertraut gemacht. Zugleich erhalten sie vielfältige Anregungen für die Gestaltung eines wissenschaftlich fundierten obligatorischen und fakultativen Unterrichts. Sie erwerben Fertigkeiten im Umgang mit vorgefertigter Standard-Software und werden befähigt zur Wartung und Pflege der Schul- und Unterrichtssoftware."* [SP89]

Dieser Grundkurs *„Informatik"* umfasste 480 Stunden (davon 210 h Vorlesungen / 120 h Übungen / 150 h Praktika). Beispielsweise wird für das Gebiet „Programmierungstechnik und Software-Technologie" neben einer Vertiefung der Kenntnisse zum Algorithmusbegriff explizit gefordert, sowohl die Methoden der Programmentwicklung und die grundlegenden Konzepte einer imperativen höheren Programmiersprache als auch ein Überblick über wesentliche Konzepte objektorientierter, funktionaler und logischer Programmiersprachen zu vermitteln. Zu diesem Kurs gehörte ein Praktikum, in dem die Studenten praktische Fähigkeiten im Umgang mit Arbeitsplatzcomputern und entsprechender Standard-Software erwarben.

Im Gebiet „Rechnersysteme" waren u.a. sowohl die von-Neumann-Architektur eines Digitalrechners als auch Struktur und Funktion eines Prozessors und die Organisationsprinzipien des Speichers zu behandeln. Für die „Theoretische Informatik" werden Algorithmen- und Automatentheorie, Formale Sprachen, Syntax und Semantik von Sprachen, Abstrakte Datentypen, Entscheidbarkeit und Berechenbarkeit sowie Methoden der künstlichen Intelligenz explizit genannt. Auf dem Gebiet „Computergeometrie" sollten die Studenten Prinzipien und Anwendungsmöglichkeiten von Algorithmen in der Geometrie, der Computergrafik, der Bildverarbeitung und der Bildanalyse kennen lernen. Gegenstand des Lehrgebiets „Angewandte Informatik", welches ebenfalls ein Praktikum einschließt, waren Themen wie Informationssysteme, Modellierung und Simulation, Datenbanken, Software-Werkzeuge sowie Datenschutz und Datensicherheit. Schließlich ist das Lehrgebiet „Physikalische Grundlagen der Informatik" zu nennen, in dem wichtige physikalische und elektronische Kenntnisse als Voraussetzung für das Verständnis der Wirkprinzipien von Bauelementen und Baugruppen eines Digitalrechners vermittelt werden. Stark auf die Realisierung der künftigen Lehr-Lern-Prozesse an Schulen ausgerichtet war das Gebiet „Methodik des Informatikunterrichts" (heute als Didaktik bezeichnet), das auf die Befähigung für den obligatorischen Unterricht sowie dessen prob-

lemorientierte Gestaltung orientiert war. Bemerkenswert ist die Tatsache, dass dort auch ein Teilgebiet „Experimente für den Informatikunterricht“ beschrieben wird.

Da die beiden Studiengänge in Chemnitz und Güstrow den Bedarf an Informatiklehrern bei weitem nicht decken konnten, wurde ein Postgradualstudium für Informatiklehrer eingerichtet, um kurzfristig Fachlehrer für das neue Schulfach Informatik auszubilden. Teilnehmer waren ausgebildete Lehrer anderer Fächer, die für ein Schulhalbjahr vom Unterricht freigestellt waren und in einem Kompaktkurs Kenntnisse und Fähigkeiten für den Informatikunterricht an den EOS erwarben.

> *„Voraussetzung für die Zulassung am postgradualen Studium für das Fach Informatik war eine abgeschlossene Hochschulausbildung als Diplomlehrer, vorrangig für Mathematik, Physik oder Polytechnik, eine erfolgreiche Tätigkeit bei Führung der Bildungs- und Erziehungsarbeit und der geplante bzw. weiterhin vorgesehene Einsatz als Lehrer im obligatorischen und fakultativen Informatikunterricht der POS, der EOS und der Berufsausbildung mit Abitur.“ [SP87]*

Grundlage dieses Kurses waren der vom zuständigen Ministerium bestätigte Studienplan und eine einheitliche Lehrkonzeption für das postgraduale Studium Informatik, die von den beteiligten Einrichtungen unter Federführung der Akademie der Pädagogischen Wissenschaften erarbeitet wurde. Es stand im Rahmen eines Semesters mit 15 Wochen Lehrveranstaltungen (Vorlesungen, Übungen, Seminare, Praktika) ein Zeitfonds von 330 Stunden zur Verfügung. Hinzu kamen 5 Wochen vorlesungsfreie Zeit für Praktika, Selbststudium und eine selbständige wissenschaftliche Arbeit. Die Abschlussprüfung war eine mündliche Prüfung, über die ein entsprechendes Zeugnis ausgestellt wurde. An dieser Ausbildung konnten sich nicht alle Universitäten und Pädagogischen Hochschulen beteiligen. Es wurden besonders solche ausgewählt, an denen es bereits Aktivitäten zur Informatik im Rahmen der bestehenden Studiengänge für Mathematik- oder für Polytechniklehrer gab. In Sachsen konnte dieses Postgradualstudium ab Frühjahr 1988 nur an der PH Dresden und der TU Chemnitz aufgenommen werden.

Spezialisierungsrichtung *„Informatik und Mathematik“*							
Nr.	Lehrgebiet	Stunden	V	Ü	P	SWS	Abschluss
1	Weltanschaulich-philosophische, historische, ökonomische und soziale Aspekte der Informatik	15	15				Testat
2	Algorithmen, Daten, Programme	150	75	30	45	10	Prüfung
3	Hardware und Betriebssysteme	30	30			2	Prüfung
4	Mathematische Grundlagen der Informatik	30	30			2	Beleg
5	Mathematische Modellierung ausgewählter Prozesse	60	15	15	30	4	Beleg
6	Methodik des Informatikunterrichts	45	15	15	15	3	Testat
Σ		**330**	**180**	**60**	**90**	**22**	

Die zentralen Vorgaben für dieses postgraduale Studium und deren Umsetzung (einschl. Umfang, Art der Lehrveranstaltungen und der geforderten Abschlüsse) sind in der obigen Übersicht zusammengefasst. In der Spezialisierungsrichtung *„Informatik und Prozessautomatisierung“* wurden die Lehrgebiete 4 und 5 durch „Automatisierung von Produktionsprozessen“ ersetzt.

Auf der Grundlage dieser zentralen Vorgaben wurde an der Pädagogischen Hochschule Dresden eine konkretisierte Lehrkonzeption für das postgraduale Studium Informatik mit der Spezialisierungsrichtung Mathematik erarbeitet. Die Umsetzung dieser Konzeption lag in der Verantwortung des Wissenschaftsbereichs Numerische Mathematik/Informatik der Sektion Mathematik. Als Lehrkräfte waren vorrangig Hochschullehrer und Mitarbeiter dieses Bereiches, des Wissenschaftsbereiches Methodik des Mathematikunterrichts, aber auch Mitarbeiter der Rechenstelle der PH und Gastdozenten von der TU Dresden und aus dem Kombinat Robotron tätig.

In dieser Lehrkonzeption war zum Beispiel zum Thema „Programme/Weitere Aspekte“ im Lehrgebiet „Algorithmen, Daten, Programm“ festgehalten:

> *„Die Teilnehmer werden mit Grundkonzepten der Programmierung vertraut gemacht. Dabei werden Fähigkeiten zur imperativen Programmierung vorausgesetzt. ...Einführung in und exemplarische Anwendung von für die Schule relevanten Repräsentanten funktionaler bzw. logischer Programmiersprachen (LOGO und PROLOG) sind zur Erreichung dieses Ziels erforderlich. Die dieser deklarativen Programmierung zugrunde liegenden Abarbeitungsprinzipien legen den Übergang von sequentieller zu paralleler Verarbeitung nahe.“ [LK89]*

Im Zuge dieser Ausbildung entstanden durch die Beteiligten eine Reihe von Lehrbüchern, Unterrichtsmaterialien und Handreichungen für Lehrer, die auch an anderen Einrichtungen eingesetzt wurden. [LM] Eine Besonderheit an der PH Dresden war ein Lehrgebiet 5c: „Computer in der Bildung“. Dieses Lehrgebiet basierte auf Forschungen des verantwortlichen Wissenschaftsbereichs zur Entwicklung der Lehrprogrammformulierungssprache (LEFO) und des Autorensystems DAUS. [vgl. Ga08]

Der erste Kurs im vorgestellten postgradualen Studium fand an der PH Dresden im Frühjahrssemester 1989 limitiert mit 20 Lehrern statt, wobei die Anzahl der Bewerber diese Zulassungszahl bei weitem überstieg. Weitere Kurse folgten vom Herbst 1989 bis zum Frühjahr 1991. Es wurden in dieser Zeit bereits 106 Informatiklehrer an der PH in Dresden ausgebildet. Etwa gleichzeitig (ab Herbstsemester 1988) erhielt die PH Dresden die Genehmigung für jeweils eine Seminargruppe (30 Studierende) der Fachkombination Mathematik/Geografie, eine Drittfachausbildung Informatik durchzuführen. Bereits bei der Immatrikulation wurden die Studenten nach einer Bewerbung dafür ausgewählt. Die Lehrveranstaltungen des postgradualen Studiums standen teilweise auch diesen Studenten offen und es konnten erste Erfahrungen zur gemeinsamen Ausbildung von Lehrern und Direktstudenten gesammelt werden.

2.2 Weiterführung der Ausbildung nach 1990 in Sachsen

Mit der politischen Wende vollzogen sich auch im Hochschulwesen in Sachsen gravierende Änderungen. Am Studienort Dresden wurden die vorhandenen Einrichtungen unter dem Dach der TU vereint und eine Fachhochschule für Technik und Wirtschaft am Standort der früheren Verkehrshochschule gegründet. Das bedeutete für die Lehrerbildung eine volle Integration in die Fachfakultäten der TU und eine völlige Neuorganisation des Studienablaufs. In der Folge mussten natürlich Lehrangebote auf Diplomstudenten und Lehramtsstudenten des jeweiligen Faches zugeschnitten werden und waren nicht mehr primär auf die beruflichen Anforderungen ausgerichtet.

Grundlage der Lehrerausbildung wurde die im Jahr 1992 beschlossene Verordnung des Sächsischen Staatsministeriums für Kultus über die Erste Staatsprüfung für Lehrämter an Schulen im Freistaat Sachsen (LAPO I) [LA92]. In dieser LAPO I wurden zwei Fächergruppen definiert, aus denen nach festgelegten Bedingungen der Student seine Fachkombination wählen konnte. Es wurde nach dem Vorbild von Studienordnungen zum Lehramt in Bayern und Baden-Württemberg außerdem festgelegt, dass das Fach Informatik in Sachsen nur als ergänzendes 3. Fach studiert werden kann. Das war ein deutlicher Rückschritt, vor allem vor dem Hintergrund einer Stundentafel der Mittelschulen und Gymnasien in Sachsen, in der (anders als damals in Bayern und Baden-Württemberg) Informatik als Pflichtfach in der Sekundarstufe I verankert war.

Gerade deshalb wurden große Anstrengungen unternommen, um das postgraduale Studium für Lehrer unter den neuen Bedingungen fortzuführen, denn ein qualifizierter Fachunterricht an Schulen kann nur von fachlich entsprechend ausgebildeten Lehrern erteilt werden. Über Absolventen eines Direktstudiums war das nach den Festlegungen der LAPO I und den kaum vorhandenen Stellen in den Schulen in kürzerer Frist nicht zu erwarten. Mit großem Engagement vieler Beteiligter gelang es, diese Kurse bis zum Frühjahr 1991 mit Lehrern aus den sächsischen Mittelschulen und Gymnasien weiterzuführen. Diese Lehrer wurden allerdings nicht mehr für ein ganzes Schulhalbjahr, sondern nur noch für einen Wochentag vom Unterricht freigestellt und zwar für das gesamte Schuljahr. Bemerkenswert ist, dass teilweise fast 30 Lehrer pro Kurs teilnahmen, obwohl zu Beginn des Kurses der Abschluss noch völlig offen war. Durch die Einführung eines mehrwöchigen Kompaktkurses in der Semesterpause 1991 bzw. in den Schulferien im Februar 1991, gelang es schließlich, den Umfang und die Abschlussleistungen des postgradualen Studiums zu erreichen, was entsprechend zertifiziert wurde.

Im Laufe des Studienjahres 1990/91 waren dann die Gespräche mit den beteiligten Ministerien soweit fortgeschritten, dass ab Wintersemester 1991/92 der erste Jahrgang eines berufsbegleitenden Studiums zum Erwerb der Lehrbefähigung in einem weiteren Fach die Ausbildung beginnen konnte. Die dafür notwendige Verordnung war federführend vom Fach Informatik initiiert und wurde auf Fächer wie Englisch, Ethik und andere erweitert. Sie ist in dieser Form noch heute gültig und sieht vor, dass nach einem Studium über 4 Semester für das Lehramt an Mittelschulen und über 6 Semester für das Höhere Lehramt an Gymnasien bzw. an berufsbildenden Schulen das Studium mit einer modifizierten Staatsprüfung abgeschlossen wird. Eine Fortsetzung über ein berufsbegleitendes Referendariat mit 2. Staatsprüfung zur vollen Lehrberechtigung für das jeweilige

Fach ist ebenfalls vorgesehen. Den Teilnehmern des Kurses im Schuljahr 1990/91 wurde sogar ermöglicht, im Rahmen von Aufbaukursen den gleichen Abschluss zu erwerben. Über zusätzliche Regelungen konnten auch Lehrer mit Abschlüssen des Postgradualstudiums ihr Studium in verkürzter Form fortsetzen und den juristisch korrekten Abschluss erwerben.

Aus der heutigen Sicht spannend war die gemeinsame Erarbeitung einer Studienordnung in der Fakultät Informatik, die sich erstmalig mit der Lehrerausbildung konfrontiert sah. Dabei zeigt sich eine große Unterstützung auch für die Lehrer im berufsbegleitenden Studium, die in vielen Fällen nachhaltig die Entscheidungen von Schülern, in Dresden zu studieren, beeinflusst hat. So ist die Integration der Lehrerbildung in die Fakultät aus heutiger Sicht ein Gewinn für alle Seiten geworden, der von einer Anerkennung dieser Spezifik auf der einen Seite bis zur Bereicherung von Lehre und Forschung im Bereich von Medieninformatik oder E-Learning auf der anderen Seite reicht. Inzwischen haben seit 1992 insgesamt 591 Studierende (von 937 Anfängern) die erste Staatsprüfung für das Fach Informatik abgelegt und viele von ihnen pflegen auf den jährlichen Absolvententreffen den engen Kontakt zu ihrer Ausbildungseinrichtung.

Inzwischen kann seit dem Wintersemester 2000/01 in Sachsen die Fachkombination Mathematik/Informatik für das Lehramt im Direktstudium studiert werden, in einer Erprobung an der TU Dresden sogar noch in anderen Fachkombinationen.

3. Zur Lehrerbildung Informatik – einige Konsequenzen

Während in den westlichen Bundesländern neben Grundkursen auch viele Leistungskurse Informatik auf hohem Niveau für speziell interessierte Schüler angeboten wurden, war in der DDR ein obligatorischer Unterricht für alle Schüler der Oberstufe vorgesehen. Dies ging einher mit einem Konzept der Aus- und Weiterbildung von Lehrern, das sicher nur ein Anfang war und den Bedarf nicht so schnell gedeckt hätte. Die Rahmenbedingungen im Freistaat Sachsen ermöglichten es, dies in den neuen Bedingungen angepasster Form weiterzuführen, um einen Bestand ausgebildeter Informatiklehrer (mit Staatsprüfung) zu erreichen, der in keinem anderen Bundesland feststellbar ist.

Abgekoppelt von den politischen und gesellschaftlichen Rahmenbedingungen lassen sich die Erfahrungen auf folgende Aspekte zusammenfassen:

- Die Lehrerbildung befindet sich in Deutschland im Spagat zwischen notwendiger Fachausbildung, die den Diplom- bzw. Masterstudiengängen im Fach adäquat sind, und Orientierungen auf Schwerpunkte, die das Berufsbild stärker fokussieren. Es ist zu erwarten, dass durch die Studienumstellung und die Modularisierung, diese Probleme noch verschärft werden.
  Konsequenz: Es sind Ressourcen für lehramtsbezogene Fachmodule nötig, um eine fachlich solide Ausbildung zu sichern.
- Die freie Wahl von Fachkombinationen ist sicher für Studierende eine gute Basis und die Vielfalt hilft auch in den Schulen. Ein solches breites Angebot ist im Zuge der Modularisierung und stärkeren Pflichtbindungen bei Lehrveranstaltungen nicht

mehr im vorgegebenen Zeitraster zu organisieren.
Konsequenz: Universitäten sollten besonders geförderte Fächerkombinationen herausstellen und dafür auch Unterstützung erhalten.

- Im Vergleich zu Studiengängen vor 1990 hat sich besonders die Betreuungsrelation im Bereich der Fachdidaktik deutlich verschlechtert. Es ist in manchen Fächern kaum möglich, in der notwendigen Anzahl schulpraktische Übungen durch Mitarbeiter der Universität zu betreuen oder Praktika in Schulen zu begleiten. Ein Ausgleich durch eine Deputatsminderung bei der Übernahme durch Lehrer im Schuldienst sowie deren geeignete Qualifizierung ist bisher nicht erfolgt.
  Konsequenz: Ein Erweiterungsstudium zum Mentor kann die Voraussetzung liefern, um eine Betreuung entsprechend der Anforderungen der Ausbildung zu garantieren.

- Ohne Zweifel sollte Lehrerbildung auch künftig an Universitäten stattfinden. Für eine Beachtung im Wissenschaftsbetrieb sind Forschungsergebnisse und Drittmittel inzwischen Qualitätskriterium. Gerade im Bereich fachdidaktischer Arbeiten sind Förderungen zu unterrichtsbezogenen Projekten eher schwierig zu erreichen. In der Folge werden Mittel über andere Projekte eingeworben. Das führt nicht unbedingt zu Innovationen für den Fachunterricht, wie es gerade durch die vorgelegten Konzepte zu Bildungsstandards nötig wäre.
  Konsequenz: Ein Förderprogramm für empirische didaktische Forschungen zum Unterricht ist längst überfällig.

Betrachtet man abschließend die unter ganz anderen Bedingungen gemachten Anfänge, wird sichtbar, dass manche Erfahrungen geholfen haben, gerade die Entwicklungen zum Informatikunterricht in Sachsen und zur Lehrerbildung für das Fach Informatik anders zu gestalten. Trotz mancher Widerstände gab es seit 1992 das Pflichtfach Informatik in der Sekundarstufe I und wenig später dazu passende Lehrbücher. Die Lehrerausbildung konnte sich vor diesem Hintergrund in der Fakultät Informatik etablieren und wurde im Rahmen der Modularisierung der Studiengänge explizit berücksichtigt. Dazu war neben Hartnäckigkeit manchmal auch die richtige Aktivität zum passenden Zeitpunkt notwendig.

Literaturverzeichnis

[BI07] Grundsätze und Standards für die Informatik in der Schule - Bildungsstandards Informatik. in: LOG IN - Beilage, 27 (2007) 146/147

[Bu05] Bunke, F., Wir lernen und lehren im Geiste Lenins, in: Finke, K., u. a., Oldenburger Beiträge zur DDR- und DEFA-Forschung. Oldenburg, 2005

[EP89] Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik, Beschluss vom 1.12.1989, Sekretariat der KMK. Luchterhand, Neuwied 1991

[Ga08] Gahmig, F., Rohland, H., Unger, M.: Gab's gar schon eLearning? (in diesem Band)

[He07] Heinicke, S., Timmermann, B., Unger, M., Konzepte für die Begabtenförderung auf dem Gebiet der Informatik und ihre Umsetzung am Schülerrechenzentrum Dresden, in: Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS'03. Lecture Notes in Informatics (LNI), P-32, S. 77-87

[Ko98] Koerber, B., Peters, I.-R. (Hrsg.), Informatische Bildung in Deutschland, LOG IN Verlag, Berlin 1998

[LA92] Verordnung des SMK über die Erste Staatsprüfung für Lehrämter an Schulen im Freistaat Sachsen (LAPO I), Sächsisches Gesetz- und Verordnungsblatt Nr. 17/1992, Dresden, d. 11. Mai 1992

[LB85] Rahmenlehrplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe, Fach Informatik. Berlin, 1985, S.11

[LD89] Experimentallehrplan. Informatik Klasse 11. Volk und Wissen Berlin, 1989, S. 5

[LM] *Lehrmaterialien (Hrsg. Akademie der Pädagogischen Wissenschaften der DDR):*
Kerner, I.O., Wagenknecht, Ch., Einführung in das strukturierte Programmieren mit Basic und Kleincomputer KC 85-1. Berlin, 1986
ESP - Methodische Empfehlungen für den Computergrundkurs „Informationsverarbeitung" in Klasse 9 und den Einsatz des Computers in Klasse 10. Berlin, 1988
Anleitung zur Arbeit mit Benutzersystemen im obligatorischen Informatikunterricht, Klasse 11, als Manuskript gedruckt, Berlin, 1988
Entwickeln von Programmen, Klasse 12, Schülerarbeitsmaterial. Berlin, 1988
Informatik Klassen 9 und 10, Schülerarbeitsmaterial. Berlin, 1989
Informatik Klasse 11. Methodische Empfehlungen. Berlin, 1990
Weitere Bücher und Materialien:
Friedrich, S., Walter, J.-H.: Lehrmaterial Pascal - Foliensatz und Begleitmaterial. PH Mühlausen, Zentralstelle für Rationalisierungsmittel der Lehrerbildung. Erfurt, 1986
Kerner, I.O.: Numerische Mathematik mit Kleinstrechnern, MfL Bd.18, Deutscher Verlag der Wissenschaften. Berlin, 1989
Kerner, I.O., Informatik, MfL Bd. 9, Deutscher Verlag der Wissenschaften. Berlin, 1990
Schilling, A., Töpfer, W.: Informatik. Volk und Wissen Verlag. Berlin, 1988
Wenzel, R.-D., Programmierung in PASCAL. Erprobungsmaterial für einen Schulversuch im Schülerrechenzentrum Robotron beim Pionierpalast Dresden, 1987

[LK89] Lehrkonzeption für das postgraduale Studium Informatik an der Pädagogischen Hochschule „K. F. W. Wander". Dresden, 1989

[Ne97] Neupert, H. Friedrich, S.: Lernen mit Netzen – Lernen über Netze. in: LOG IN 17 (1997) 6, S. 18-24

[SP87] Studienplan für das postgraduale Studium zur Qualifizierung von Diplomlehrern auf dem Gebiet der Informatik an Universitäten und Hochschulen der DDR, Ministerrat der DDR. Berlin, 1987

[SP89] Studienplan für die Ausbildung von Diplomlehrer das allgemeinbildenden polytechnischen Oberschulen in der Fachkombination Mathematik/Informatik an Universität und Hochschulen der DDR, Ministerrat der DDR. Berlin, 1989

[Ti08] Timmermann, B., Weeger, M.: Informatische Bildung in Deutschland. in: LOG IN 27 (2007) 148/149, S. 4 – 7

Gab's gar schon eLearning?

Holger Rohland Michael Unger Frieder Gahmig

AG Didaktik der Informatik/Lehrerbildung
TU Dresden, Fakultät Informatik, Institut für SMT
01187 Dresden
hr8@inf.tu-dresden.de mu2@inf.tu-dresden.de

Linde-KCA-Dresden GmbH
Bodenbacher Straße 80
01277 Dresden
Frieder.Gahmig@linde-kca.com

Abstract: Bereits zu Beginn der 80-er Jahre wurde an der damaligen Pädagogischen Hochschule Dresden mit der Entwicklung eines Lehrsystems begonnen, welches die Potenzen des Computers als Unterrichtsmittel erschließen sollte. Basierend auf Großrechentechnik wurde zunächst eine Sprache entworfen, die "LEhrprogramm FOrmulierungssprache" - kurz LEFO - die es auch dem nicht informatisch gebildeten Lehrer ermöglichen sollte, kleine Lernprogramme zu schreiben. Später wurde diese Sprache in das "Dresdner AUtorenSystem" DAUS integriert, welches 1990 "Marktreife" erreicht hatte und auf PC - Basis verfügbar war.

Im Artikel soll dargestellt werden, welche Ideen in den Entwicklungen der 80-er Jahre steckten, wie konkrete Implementierungsprobleme aussahen und was letztendlich auf der Basis dieser Software an Beispielen für den Einsatz im Unterricht entwickelt werden konnte. Dabei soll deutlich gemacht werden, dass einige der damals angestellten Überlegungen zum didaktischen Hintergrund der Erstellung von Lernprogrammen durchaus noch heute ihre Gültigkeit haben. Andererseits wird auch zu zeigen sein, welche Grenzen - sowohl informatische wie auch didaktische - dazu geführt haben, dass letztlich das Grundkonzept der Gesamtanwendung heute nur in stark modifizierter Form als Learning Management System zum Einsatz kommen könnte - dann aber mit Potenzen, die den meisten aktuell verfügbaren Werkzeugen überlegen wären!

1 Wie alles begann

Ende der 70-er Jahre wurden weltweit Konzepte und Systeme entwickelt, die den Computer als Mittel und Gegenstand der Ausbildung im Bereich der Hochschulausbildung zum Inhalt hatten. Diese Entwicklung wurde besonders Anfang der 80-er Jahre in Amerika durch die rasante Computerisierung der Gesellschaft beschleunigt. Bereits 1983 wurden in den USA mehr Computer (10,3 Mill.) als Autos (6,3 Mill.) ausgeliefert. Wie sich später zeigte, war dies der Beginn eines technologischen Umbaus aller Lebensbereiche und revolutionierte somit auch das Bildungswesen.

Der Versuch, die Aus- und Weiterbildung von Schülern und Studenten durch den Computer zu unterstützen, ist vor dem Vorhandensein der PCs mit Entwicklungsarbeiten auf dem Gebiet des programmierten Unterrichts und der technischen Lehr- und Lernmittel verbunden. Wichtige Arbeiten von SKINNER, PRESSY u. a. [SK58; PR27] auf diesem Gebiet sind dafür der Beleg. Dieser Stand der Forschung und Entwicklung war unter anderem die Grundlage dafür, dass mit der weltweiten Verbreitung der Computer eine sehr große Anzahl von Lösungen für den computergestützten Unterricht entstanden.

Dabei wurde eine scheinbar unüberschaubare Breite von computergestützten Lehr- und Lerntechniken in Lehr- und Lernsystemen umgesetzt, die u. a. mit den Termini CAT (Computer Assisted Teaching), CBT (Computer Based Training) oder CAE (Computer Aided Education) in Verbindung zu bringen sind.

Rückblickend und stellvertretend für die vielfältigen Ansätze der Umsetzung computergestützter Lehr- und Lerntechniken sollen die Arbeiten von ZINN [ZI73] und Fisher [FI82] genannt werden. ZINN stellte einen Lifecycle für die Entwicklung von Courseware im Rahmen eines Autorensystems dar. Es zeigte sich später, dass nahezu alle später entstandenen Autorensysteme diesen Lifecycle abbildeten.

FISHER veröffentlichte Arbeiten, die sich mit der Individualisierung von Lernwegen innerhalb von Computersitzungen beschäftigten. BODE [BO75] stellte eine an der Universität Karlsruhe entwickelte Schülerkommandosprache für das System LEGIS vor, die, für die damalige Zeit typisch, als Fachsprache das Ziel hatte, Algorithmen und Workflows für spezielle Anwendungsfälle für eine Bearbeitung mit Computern zugänglich zu machen.

An der Pädagogischen Hochschule Dresden bestanden Anfang der 80 er Jahre gute Voraussetzungen dafür den Computer als Mittel und Gegenstand der Ausbildung besonders in die Forschungsarbeit zu integrieren. Drei wichtige Faktoren hiefür seien genannt.

Erstens kam es zur Bildung eines von Herrn Prof. Kerner geleiteten Lehrstuhls Informatik. Dieser Lehrstuhl war der erste Schritt um in der Ausbildung der künftigen Mathematiklehrer intensiver informatische Inhalte zu verankern, was für die spätere Entwicklung von Informatiklehrplänen sehr hilfreich war.

Zweitens standen aus der pädagogischen Forschung Ergebnisse zur Verfügung, die die Verbesserung von Lehrprozessen zum Ziel hatten. Aus empirisch ermittelten Tätigkeitsresultaten sind Symptome beschrieben worden, die als Grundlage für die Lenkung von Lernprozessen eingesetzt werden konnten. Dies sollte ein ganz wesentlicher Zugang für die Konzeption des mikrorechnergestützten Lehrsystems auf der Basis von Bürocomputern (MLS/BC) werden.

Drittens bot der IT- Standort Dresden mit der TU und Robotron gute Voraussetzungen dafür, die seit 1983 verfügbaren BC und PC in der DDR als eine technische Basis für ein zu entwickelndes Lehr und Lernsystem zu nutzen.

Die 1981 vorgeschlagene Lehrprogrammformulierungssprache LEFO [GA81] ist eines der ersten Ergebnisse, welches in der Verbindung von Informatik und Lehrerbildung entstanden ist. LEFO hob sich zum damaligen Zeitpunkt in der Art und Weise der Unterstützung zur Individualisierung der Lernwege von vergleichbaren Systemen, in denen beispielsweise Auswahlantwortkontrollen dominierten, ab. Weitere Arbeiten der Forschungsgruppe nahmen, zumindest gedanklich, Überlegungen voraus, die uns heute als Massenware in Form von Trainingsprogrammen und Tutorials bekannt sind.

Dass die Qualität dieser Courseware oft nicht über das Niveau der 70-er Jahre hinaus geht, unter didaktischer Sicht häufig sogar dahinter zurück bleibt, sollte Motivation genug sein, die damaligen Ansätze einer kritischen Weiterentwickelung zu unterziehen. Der Bedarf erschließt sich dem Betrachter besonders auch aus der Sicht von Industrieunternehmen, in denen vor allem auch die individuelle Aus- und Weiterbildung eine strategische Bedeutung habt.

2 LEFO - Steuerstrukturen und DAUS - Antwortkontrollverfahren - das didaktische Potenzial der 80-er Jahre

Die Weiterentwicklung der o. g. Ansätze führte Ende der 80-er Jahre zum Dresdner Autorensystem (DAUS). Das System wurde im Rahmen einer Dissertation [RU91] entwickelt und war besonders für die Erzeugung von Lehrprogrammen zur Unterstützung von Kontroll-, Übungs- und Festigungsprozessen konzipiert. Dabei lagen die besonderen Vorzüge in einer einheitlichen Behandlung aller Schülereingaben auf der Basis eines breiten Spektrums an Antwortkontrollverfahren und der dadurch möglichen Steuerung des Lerners in Abhängigkeit von seinen erbrachten Leistungen. Zur Erarbeitung eines Lehrprogramms gehörte die Vorgabe von geeigneten Daten (Fragen und Vorgabelösungen, Hilfestellungen, Grafiken) und einer in der Sprache LEFO (vgl. Kapitel 1) beschriebenen Steuerstruktur. Die Übersicht über ein DAUS-Lehrprogramm mit wissensbasierter Komponente zeigt Abbildung 1.

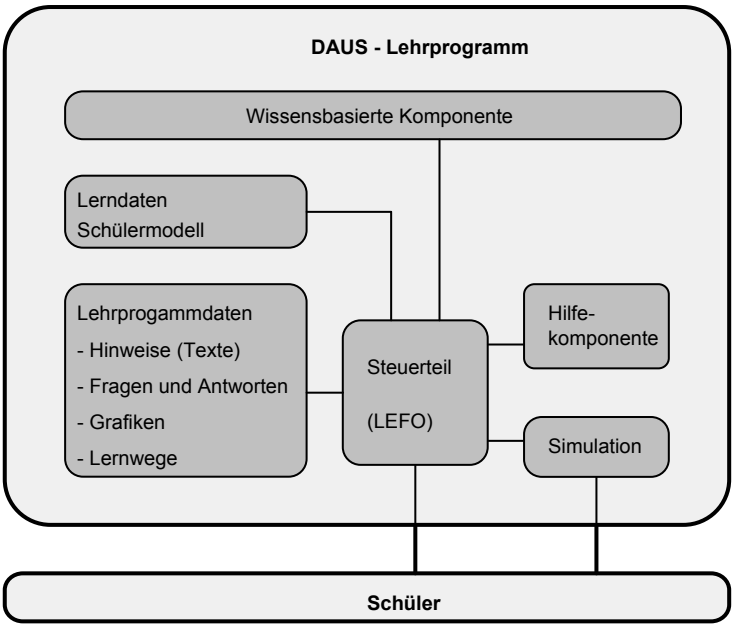


Abbildung 1: DAUS Lehrprogramm mit wissensbasierter Komponente

2.1 LEFO - Steuerstrukturen

Im Unterschied zum erforschenden, explorativen Lernen lag die Zielstellung bei der Entwicklung von LEFO / DAUS darin, Lehrprogramme zu erzeugen, die gezielt dem Erwerb von deklarativem Wissen dienen sollten. Der Schüler sollte nach Abarbeitung eines solchen Programms das vom Autor vorgegebene Lernziel erreicht haben, wozu eine genaue Analyse seiner fehlerhaften Lösungen und eine restriktive Steuerung seines Lernweges in Abhängigkeit vom erzielten Lernfortschritt und von den diagnostizierten Fehlern nötig war.

Bei jeder Schülerantwort wurden im sogenannten Schülermodell ausführliche Informationen abgelegt, die zur Steuerung des weiteren Lernweges verwendet werden konnten. Dazu gehörten neben der Schülerantwort selbst unter anderem

- der Bezeichner der Aufgabe und deren laufende Nummer,
- die erreichbaren Punkte und die erreichten Punkte,
- die Richtigkeit der Antwort,
- bis zu drei Resultateigenschaften („Symptome“) je Aufgabe,
- die bei der wissensbasierten Antwortkontrolle gewonnenen Regeln,
- die für die Lösung der Aufgabe benötigte Zeit.

Die Speicherung dieser Informationen erfolgte in einem standardisierten Datenbankformat, so dass eine individuelle Auswertung für den Lehrer oder auch den Programmautor möglich war. Im Dresdner Autorensystem war aber auch eine Auswertungskomponente vorhanden, die typische Anfragen an diese Datenbank sowohl zu einem konkreten Schüler als auch zum Lehrprogramm insgesamt erlaubte. Damit war es möglich das Programm selbst zu evaluieren und zum Beispiel typische Lernwege zu erkennen und bei der weiteren Entwicklung des Programms die Lernwegsteuerung noch spezifischer zu gestalten.

Die wichtigsten Steuerstrukturen der Sprache LEFO sind Verzweigungen und Zyklen, wobei die Besonderheit im Zugriff auf das Schülermodell liegt. So ist eine Verzweigung nicht nur in Abhängigkeit von der Richtigkeit der zuletzt gelösten Aufgabe möglich, sondern es kann auf den vom Schüler beschrittenen Lernweg (die Folge von Bezeichnern der gelösten Aufgaben), die für eine bestimmte Anzahl von Aufgaben benötigte Zeit, die Häufigkeit des Auftretens bestimmter Symptome oder auch das Auftreten bestimmter Regeln (bei der wissensbasierten Antwortkontrolle) zugegriffen werden.

2.2 DAUS - Antwortkontrollverfahren

Zur Beurteilung verschiedener Verfahren, mit denen eine Reaktion des Schülers erfasst und bewertet werden soll, ist es notwendig eine geeignete Methode zur Klassifizierung solcher Verfahren zu finden. In [RU91] wurde ein Formalismus entwickelt, der diese Einteilung auf der Basis allgemeiner Merkmale erlaubt. Dazu werden die bei der Antwortkontrolle auftretenden Mengen analysiert:

- die Menge **S** der möglichen Schülerantworten,
- die Menge **A** der vom System vorgegebenen Vorgabelösungen,
- die Menge **B** der Regeln, mit denen aus gegebenen Werten Vorgabelösungen erzeugt werden
- die Menge **L** der über den Lernenden zu speichernden Informationen (damals als „Schülermodell“ bezeichnet)

Auf dieser Grundlage erfolgte eine Einteilung in primitive, klassische, generative und wissensbasierte Antwortanalyseverfahren. Verfahren, bei denen die Vorgabelösungen direkt vom Autor erzeugt werden (also die Menge **B** leer ist), werden **primitiv** genannt. In diese Kategorie fallen die sogenannten "Multiple Choice Fragen", bei denen der Lernende lediglich eine oder einige von mehreren gegebenen Möglichkeiten als Antwort auswählt. Neben der einfachen Ja/Nein Frage stellt dieser Typ die einfachste Form der Antwortkontrolle dar. Rückschlüsse auf mögliche Ursachen bei falschen Schülerantworten sind hier praktisch unmöglich. Dieses Verfahren eignet sich aber sehr gut für solche Tests, bei denen die Fertigkeiten des Lernenden lediglich überprüft werden sollen. Eine automatisierte Auswertung ist sehr einfach zu realisieren, das Schülermodell wird durch Informationen wie richtig/falsch, der Nummer der gegebenen Antwort und vielleicht der benötigten Zeit gebildet.

Zu den **klassischen** Verfahren gehören insbesondere Texteingaben, die mit einem geeigneten Parser untersucht und ausgewertet werden. Da der Schüler hier freien Text eingeben kann, ist die Anforderung an den Lernenden höher und die Auswertung aufwändiger. Dafür ist eine genauere Analyse der Kenntnisse des Schülers möglich, insbesondere entfällt die Chance durch Raten bzw. Ausschließen offensichtlich falscher Antworten die richtige Lösung zu finden. Diese Verfahren können durch verschiedene Optionen wie Unterscheidung von Groß- und Kleinschreibung, Rechtschreibtoleranz, Nutzung von Jokerzeichen, Vollständigkeit und Reihenfolge bei mehrteiligen Antworten ergänzt werden. Die Menge **B** entspricht hierbei den Optionen, mit denen der Parser die Zeichenkette des Schülers analysiert. Typisch falsche Lösungen müssen aber vom Programmautor explizit vorgegeben werden.

Wenn die Vorgabeantworten erst durch Anwendung von Bildungsvorschriften aus der Menge **B** generiert werden, dann heißt das Verfahren **generativ**. Typische Vertreter sind Kontrollverfahren, bei denen eine numerische Antwort des Schülers analysiert wird. In diesem Fall liegen die gegebenen Werte meist als Zahlen (evtl. zufällig erzeugt) vor, die Regelmenge **B** wird durch mathematische Funktionen repräsentiert. Das Generieren der Vorgabeantworten erfolgt durch Anwenden dieser Funktionen auf die Zahlen. Zur ausführlichen Auswertung sollten im Schülermodell hier neben den bereits genannten Informationen auch die generierten Werte sowie die Schülerantwort selbst gespeichert werden.

Je nach Kontrollverfahren haben die Elemente der Mengen unterschiedliche Struktur. Im Interesse eines einheitlichen Steuerungs- und Auswertemechanismus sollte aber die Menge **L** so konstruiert werden, dass die Analyseergebnisse für alle Kontrollverfahren damit erfasst werden können.

Besonders genau kann die Lösung eines Schülers dann untersucht werden, wenn es gelingt diejenigen Elemente aus **B** zu bestimmen, deren Anwendung auf die gegebenen Größen die konkrete Schülerlösung ergibt. Dazu muss die Vorgabelösung nicht mehr nur die gegebenen Größen enthalten, sondern auch die Regeln, nach denen daraus die richtige Antwort gebildet werden kann. Wichtigstes Element dieses Verfahrens ist die Regelmenge **B**, zu deren Vorgabe sich beispielsweise die Programmiersprache Prolog anbietet. Da darin das Wissen erfasst sein soll, welches der Schüler zur Lösung der Aufgabe benötigt (auch das „falsche“ Wissen, also solche Regeln, die zu typisch fehlerhaften Lösungen führen), werden Verfahren dieser Art als **wissensbasiert** bezeichnet.

An einem Beispiel sollen das Vorgehen bei diesem Antwortanalyseverfahren dargestellt und seine Vorteile aufgezeigt werden. Zur Vereinfachung wird nur eine Aufgabe betrachtet. Sie heißt: Berechne: $\frac{2}{3}+\frac{2}{5}$. In Vorbereitung der Erstellung eines Lehrprogramms sollte eine Schülergruppe mehrere solche Aufgaben lösen um nicht nur typisch falsch Antworten zu erkennen, sondern auch die falsch von den Schülern angewandten Regeln zu identifizieren. Ein solcher Test könnte folgendes Ergebnis haben:

$\frac{16}{15}$	richtiges Ergebnis; Der Schüler hat den Hauptnenner gebildet, beide Brüche erweitert, die Zähler addiert und den Hauptnenner beibehalten.
$\frac{1}{2}$	Der Schüler hat die beiden Zähler und Nenner addiert und das Ergebnis gekürzt.
$\frac{4}{8}$	Der Schüler hat die beiden Zähler und Nenner addiert und das Ergebnis nicht gekürzt.
$\frac{4}{3}, \frac{4}{5}$	Der Schüler hat die beiden Zähler addiert und einen Nenner beibehalten.
$\frac{1}{4}$	Der Schüler hat die beiden Nenner addiert und einen Zähler beibehalten und das Ergebnis gekürzt.
$\frac{2}{8}$	Der Schüler hat die beiden Nenner addiert und einen Zähler beibehalten und das Ergebnis nicht gekürzt.

Eine wissensbasierte Antwortkontrolle muss nun auf die gegebenen Größen (hier: die beiden Brüche, also 4 ganzzahlige Werte) die richtigen und „falschen" Regeln so lange anwenden, bis sie das Ergebnis des Schülers erhält. In diesem Fall kann davon ausgegangen werden, dass die verwendeten Regeln die gleichen sind, die auch der Schüler verwendet hat. Damit ist ein methodisch wesentlich präziseres weiteres Vorgehen im Programm möglich. Zum einen können dem Lernenden gezielt Hinweise für die „richtige" Regel gegeben werden und andererseits Übungsaufgaben so ausgewählt werden, dass genau diese Regelanwendung geübt wird.

Die technische Umsetzung basierte in DAUS auf Prolog, die „richtigen" und „falschen" Regeln wurden als Prolog-Klauseln formuliert und die jeweils tatsächlich benutzten Regeln in eine Liste eingetragen. Zur Übergabe von Parametern wurden Textdateien benutzt. Damit konnte die Liste der verwendeten Regeln an das Lehrprogramm übergeben und dort sowohl innerhalb von Steuerstrukturen als auch zur nachträglichen Auswertung genutzt werden. Da auch das Schülerergebnis selbst gespeichert wurde, war es möglich, die vorgegebenen Regeln auf ihre Relevanz hin zu testen und gegebenenfalls zu ändern bzw. zu erweitern.

Die Formulierung der Klauseln entspricht der Vorgabe von Antwortvarianten und stellte den Schwerpunkt der Arbeit des Programmautors dar, wobei nicht das Umsetzen in Prolog-Klauseln schwierig ist, sondern das exakte Finden und Beschreiben der Regeln. Der hohe Aufwand für diese didaktische Aufbereitung des Lehrstoffes, die der Lehrer im Unterricht aufgrund seiner im Studium erworbenen breiten „Wissensbasis" oft intuitiv vornimmt, verhindert vermutlich bis heute den Einsatz von Lehrsystemen mit solchen Antwortkontrollverfahren. Die technische Realisierung mit DAUS war jedenfalls schon vor 18 Jahren möglich.

3 LEFO und DAUS - Potenzial und Grenzen aus heutiger Sicht

Betrachtet man die dargestellten Aspekte von LEFO und DAUS aus heutiger Sicht, so wird zweierlei deutlich: Der damals verfolgte Ansatz deckt nur einen sehr begrenzten Teil dessen ab, was wir heute unter eLearning verstehen. Andererseits scheitern eine Vielzahl moderner eLearning Angebote gerade an didaktischen Unzulänglichkeiten - die möglicherweise auch auf dem Wege der Bewahrung des damals erreichten Forschungsstandes vermeidbar wären.

3.1 Was können moderne Learning Management Systeme was DAUS nicht konnte?

Zunächst wäre zu klären, in welchem Verhältnis eigentlich die Begriffe "Dresdner Autorensystem" und "Learning Management System" (LMS) stehen. Folgt man der Erklärung von Baumgartner [BHH2002, S. 27], so besitzt ein LMS fünf grundlegende Funktionsbereiche:

- Präsentation von Inhalten wie Text, Grafik, Bild, Ton und Film
- Kommunikationswerkzeuge wie Mail, Forum, Chat und Application sharing
- Werkzeuge zur Erstellung von Aufgaben und Übungen
- Evaluations- und Bewertungshilfen
- Administration von Lernenden, Inhalten, Kursen, Lernfortschritten und Terminen

Deutlich davon abgegrenzt (bis auf den Bereich der Aufgaben und Übungen) geht Baumgartner davon aus, dass das Erstellen der Inhalte mit separaten Autorenwerkzeugen erfolgt, deren Ergebnisse dann in das LMS eingebunden werden. Als Beispiele hierfür nennt er HTML - Editoren.

Analysiert man nun noch einmal die grundlegenden Funktionen von DAUS, so wird sehr schnell deutlich, dass es eben mehr als ein AUTORENsystem war. Neben den Möglichkeiten zur Erstellung von Inhalten (Texte, Grafiken) und Aufgaben verfügte DAUS über zahlreiche Features, die man heute als Funktionen eines LMS bezeichnen würde:

- Es war möglich, das fertige Lernprogramm aus DAUS heraus zu starten (entspricht der Präsentation von Inhalten)
- Es gab Evaluations- und Bewertungshilfen wie die oben erwähnten Antwortkontrollverfahren, deren Resultate in einer Datenbank gespeichert wurden und damit für die Analyse des Lernfortschritts zur Verfügung standen (heute steht hierfür der Begriff des "Lerner Tracking")
- Es war möglich - und sogar besonders exponiert (vgl. die Ausführungen zur Steuerung in Kapitel 2) - den Lerner in Abhängigkeit vom Lernfortschritt auf dem Weg durch das Lernprogramm zu steuern (Baumgartner nennt dies "Learning Flow Management")

Letztlich war das Nichtvorhandensein von Kommunikationswerkzeugen und damit aller Möglichkeiten der Kommunikation, Kooperation und Koordination zwischen den Lernern und zwischen Lerner und Tutor die einzige grundlegende Funktion, welche DAUS, dem Entwicklungsstand der Technik geschuldet, nicht aufzuweisen hatte.

Diese fehlende Funktion stellt natürlich im Sinne des heutigen Verständnisses von eLearning eine ganz gravierende Einschränkung dar. Teilt man eLearning Szenarien in solche für das Individuallernen und solche für kooperatives Lernen, so ist klar, dass sich die zweite genannte Gruppe gar nicht realisieren ließ. Aber auch im Individuallernen ist das Fehlen von Kommunikationsmöglichkeiten - vor allem mit einem Tutor - ein nicht unerheblicher Nachteil. Da damit auch der Lernfortschritt erst im Nachhinein bewertet werden kann, ist eine tutorielle Steuerung des Lernweges, wie sie heute präferiert wird, nicht möglich.

Damit war natürlich der didaktische Spielraum des Einsatzes von DAUS - Programmen auf einige wenige Lernszenarien beschränkt. Insbesondere dort, wo es im Sinne der behavioristischen Lerntheorie um den Erwerb von deklarativem Wissen ging, hatte der verfolgte Ansatz aber eindeutige Stärken. Und auch wenn es von manchem "modernen" Pädagogen geleugnet wird: das Erlernen von Geschichtszahlen, des kleinen Einmaleins oder von Vokabeln, das "Einpauken" grundlegender physikalischer Formeln, der Elemente des Periodensystems, anatomischen oder topografischen Faktenwissens lässt sich auch heute noch nach diesem lerntheoretischen Ansatz realisieren - der Beweis, das alternative Ansätze bessere Ergebnisse bringen, steht jedenfalls aus!

Es soll an dieser Stelle natürlich nicht verschwiegen werden, dass es auch hinsichtlich der Gestaltung der Benutzeroberfläche und des Einsatzes multimedialer Elemente schon allein wegen der zur Verfügung stehenden Hard- und Software Restriktionen gab - aber dies stellt im didaktischen Sinne nur eine weniger schwerwiegende Einschränkung dar.

3.2 Welche Ansätze wären einer Weiterverfolgung wert gewesen?

Nur wenn man die Frage der Eignung des Einsatzes von eLearning in Individuallernszenarien zum Erwerb und zur Festigung deklarativen Wissens prinzipiell negiert, wird man zu der Aussage gelangen, dass es keine Ansätze gibt, deren Weiterverfolgung lohnenswert wäre. Betrachtet man allerdings die potenziellen Einsatzgebiete von eLearning unvoreingenommen und begibt man sich nicht auf die Position derer, die Lernen nur noch im konstruktivistischen Kontext für möglich halten, so geraten zwangsläufig auch Lernszenarien, wie sie mit DAUS realisierbar waren, in den Fokus der Aufmerksamkeit.

Neben den schon weiter oben genannten Einsatzmöglichkeiten im Bereich allgemeiner Bildung spielen solche Szenarien auch und gerade in der betrieblichen Weiterbildung eine ganz bedeutende Rolle. Im Sinne lebenslangen Lernens wird erwartet, dass der Mitarbeiter sich weitgehend selbständig in neue Wissensgebiete einarbeitet um den sich wandelnden Anforderungen im Arbeitsprozess gewachsen zu sein. Häufig kommen solche Fortbildungsangebote allerdings nicht über eine lose Sammlung von - zugegeben oft multimedial gut aufbereiteten - Lernmaterialien hinaus.

Die didaktische Qualität solcher Angebote ist äußerst zweifelhaft. Eine Expertenbefragung [BR03, S. 77] ergab, dass die unzureichende Didaktik die entscheidende Marktbarriere für eine rasche Ausweitung von eLearning Angeboten überhaupt ist. Eine der entscheidenden didaktischen Hürden dabei ist, dass die Mehrzahl aller Angebote auf das selbstorganisierte Lernen setzt. Hält man sich dann vor Augen, dass diese Art des Lernens für die Mehrzahl der Menschen mit Anforderungen verbunden ist, denen sie nicht gewachsen sind, wird die Notwendigkeit von eLearning Angeboten, die den Intentionen von Daus folgen, schnell ersichtlich.

Welche Aspekte betrifft dies nun im Einzelnen? Der alles übergreifende Aspekt ist die Frage der Motivation des Lernenden. Gerade das Gefühl mit dem Lernstoff völlig allein gelassen zu sein und keinen "roten Faden" zu erkennen, wirkt zutiefst demotivierend. Aber auch das undifferenzierte Feedback auf die Lösung von Übungsaufgaben - wie es die meisten Learning Management Systeme bieten - ist der Motivation des Lernenden äußerst abträglich.

Was die Steuerung des Lernablaufs betrifft, so bieten die meisten heutigen Systeme dem Autor außer der Strukturierung der Lernmaterialien in Form eines "Lernpfads" - dargestellt durch die baumartige Anordnung der Kapitelüberschriften - keinerlei weitere Möglichkeiten. Auch die wenigen Systeme, die wie WebCT konditionale Freigaben ermöglichen, kommen dabei nicht an die Bedingungsvielfalt von LEFO heran. In der Regel ist es lediglich möglich neue Lerninhalte in Abhängigkeit der erreichten Punktzahl bei einem Test oder zu einem bestimmten Zeitpunkt freizugeben. Differenzierungen nach Symptomen (im Sinne des Erkennens typischer Fehler), nach der Anzahl der notwendigen Wiederholungen oder nach der für die Lösung einer Aufgabe benötigten Zeit gibt es praktisch nicht. Auch unterstützt kein System eine mehrfache Verzweigung in Abhängigkeit vom Lernfortschritt.

Im Bereich der Antwortkontrolle ist neben der in den meisten Systemen stark eingeschränkten Auswahlmöglichkeit bezüglich verschiedener Aufgabentypen vor allem die Frage des Feedbacks häufig unzureichend gelöst. Vielfach ist vom Autor außer einem "Richtig" oder "Falsch" keinerlei Reaktion des Systems einstellbar, geringfügige Schreibfehler oder die Verwendung von Synonymen führt zur Falschbewertung der Schülerlösung. Schon das Verwenden von Jokerzeichen in Vorgabelösungen wird kaum unterstützt, nur wenige Systeme bieten im Bereich numerischer Aufgaben die Möglichkeit generativer Antwortkontrollen und eine wissensbasierte Antwortkontrolle existiert praktisch in keinem der auf dem Markt befindlichen Systeme.

Zusammenfassend kann man festhalten: es existiert heute kein Learning Management System, welches dem Autor "intelligente" Antwortkontrollen und differenzierte Steuerungsmechanismen zur Verfügung stellt. Betrachtet man diese Unzulänglichkeiten, so kommt man zu der Erkenntnis, dass die entsprechenden Verfahren von DAUS auch heute noch ausgesprochen nützlich für die Gestaltung von Individuallernszenarien wären.

Dazu wären selbstverständlich Weiterentwicklungen des Systems notwendig gewesen, aber auch diesbezüglich gab es bereits Denkansätze. So ist die Frage einer Lernwegsteuerung, insbesondere auf der Basis von mehreren und im Lernprozess auch schon weiter zurückliegenden Schülerantworten, natürlich nicht einfach. Hilfreich hierfür wäre ein grafischer Programmeditor wie er in ähnlicher Weise im System Autorware von Macromedia/Adobe realisiert wurde.

Auch die Erstellung der PROLOG - Regelbasis für die wissensbasierte Antwortkontrolle war natürlich dem Nutzer auf Dauer so nicht zuzumuten. Hierfür hätte es eines Editors bedurft, der die Eingabe der Regeln über eine grafische Benutzeroberfläche ermöglicht hätte aus der automatisch die entsprechenden PROLOG - Klauseln generiert worden wären.

Insgesamt bestanden natürlich wie oben erwähnt auch bezüglich der Frage der Gestaltung der kompletten Nutzerschnittstelle, der Einbindung anderer Medienarten (mit Bildplatte wurde schon damals experimentiert!) und der Schaffung von Möglichkeiten zur Kommunikation Defizite. Wenn aber das System DAUS in diesen Bereichen im Einklang mit den hard- und softwaretechnischen Entwicklungen der 90-er Jahre weiter entwickelt worden wäre, so stünde nach unserer Überzeugung heute ein Werkzeug zur Verfügung, mit dessen Hilfe ein nicht zu unterschätzendes Spektrum von eLearning Szenarien hervorragend umgesetzt werden könnte.

Literaturverzeichnis

[BO75] Bode, A.: Lernersteuerung im Rechnergestützten Unterricht: Entwicklung und Erprobung des Systems LEGIS, Karlsruhe, 1975

[BR03] Bredemeier, W.: "Die Entwicklung der deutschen Informationswirtschaft bis 2007 – Ergebnisse einer Expertenumfrage"; Hattingen, März 2003

[BHH02] Baumgartner, P.; Häfele, H.; Maier-Häfele, K.: E-Learning Praxishandbuch. StudienVerlag, Innsbruck, Wien, München, Bozen, 2002

[FI82] Fisher, F.D.: Computer-Assisted Education – What's not Happening?, Journal of Computer based Instruction 1982(9) Seiten 19-27

[GA81] Gahmig, F.: Mensch-Maschine-Kommunikation in pädagogischen Prozessen, Pädagogische Hochschule Dresden, Dissertation, Dresden, 1981

[PR27] Pressey, S.L.: A machine for automatic teaching of drill material, school and society, 1927, 25, 1-14

[RU91] Rohland, H; Unger, M.: Ansätze zur Integration wissensbasierter Komponenten in ein Autorensystem, Pädagogische Hochschule Dresden, Dissertation, Dresden, 1991

[SK58] Skinner, B.F.: Teaching Machines, Science 1958, 128, 969-977

[ZI73] Zinn, K.L.: Requirements for Effective Authoring Systems and Assistance, University of Michigan, 1973

Das Schülerrechenzentrum Dresden von 1984 bis heute

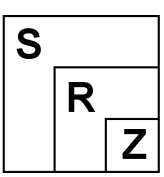
Steffi Heinicke Michael Unger

Schülerrechenzentrum
Fakultät Informatik der TU Dresden
01062 Dresden
sh5@inf.tu-dresden.de mu2@inf.tu-dresden.de

Abstract: In diesem Beitrag wird die Entwicklung des Schülerrechenzentrums (SRZ) an der Technischen Universität Dresden von seiner Gründung bis heute vorgestellt. Diese Einrichtung leistet einen Beitrag zur Förderung begabter Schüler auf den Gebieten Informatik und Elektronik. Dazu werden aufeinander aufbauende Kurse zu verschiedenen Themengebieten angeboten. Das SRZ hat sowohl eine lange Tradition, als auch eine vielversprechende Perspektive. Erreicht wird das durch die Unterstützung vieler Partner, wie die Landesregierung, die Stadt Dresden, die Technische Universität und auch viele regionale und überregionale Wirtschaftsunternehmen. Schon die Gründung des SRZ im Jahre 1984 erfolgte mit dem Ziel, besonders begabte Schüler auf dem Gebiet der Mikroelektronik und Informatik zu fördern. Und auch damals war die Unterstützung durch viele Partner (Bezirksleitung der SED des Bezirkes Dresden, Pionierpalast Dresden, Kombinat Robotron, Institut für Polytechnische Bildung der Akademie der Wissenschaften, Technische Universität, Pädagogische Hochschule und Ingenieurhochschule) Voraussetzung und Grundlage für die erfolgreiche Umsetzung dieses Ziels. Einige Beispiele aus der Arbeit des SRZ damals und heute sollen die Kontinuität der Ausbildung im Kontext der sich entwickelnden Wissenschaft Informatik und der rasanten Entwicklung des Computers als ihr wesentliches Werkzeug verdeutlichen.

1 Das SRZ von 1984 bis heute

1.1 Die Entstehung des SRZ



Die Gründung des Schülerrechenzentrums erfolgte nach einem Beschluss vom 4. Februar 1983 der Bezirksleitung der SED des Bezirkes Dresden. Danach war neben dem bereits bestehenden „Technischen Kabinett Mikroelektronik“ das „Jugendobjekt Schülerrechenzentrum – Robotron“ beim Pionierpalast Dresden aufzubauen und als eine außerschulische Einrichtung der Abteilung Volksbildung der Stadt Dresden zu installieren. Im Schülerrechenzentrum sollten ausgewählte Schüler aus Dresdner Schulen, vorwiegend ab der Klassenstufe 7, auf dem Gebiet der Mikroelektronik und Informatik gefördert werden. [SRZ04]

An dem Aufbau des Schülerrechenzentrums waren (wie auch heute) verschiedene Institutionen beteiligt. Neben dem Rat des Bezirkes Dresden waren auch das Kombinat Robotron, das Institut für Polytechnische Bildung der Akademie der Wissenschaften, die Bezirksleitung der FDJ, der Rat der Stadt Dresden und der Pionierpalast Dresden eingebunden. Zusätzlich erfolgte die Einbeziehung wissenschaftlicher Einrichtungen der Stadt Dresden, wie der Technischen Universität, der Pädagogischen Hochschule und der Ingenieurhochschule. Um den Unterricht am SRZ zu begleiten und zu unterstützen, gründeten diese Betriebe am 17. Januar 1984 ein Konsortium, mit dessen Leitung der VEB Robotron - Bürotechnik Dresden beauftragt wurde. Neben dem „Gesellschaftlichen Rat“, der für die politisch-pädagogische Beratung verantwortlich war, wurden in diesem Gremium bis 1989 die Schwerpunkte für die Arbeit am SRZ festgelegt.

Die technische Ausstattung wurde vom VEB Bürotechnik Dresden (BTD), unter Mitwirkung der Betriebe VEB Robotron-Messelektronik (MKD) und VEB Robotron-Elektronik Dresden (RED) geleistet.

Die technische Erstausstattung bestand aus:
- 16 Taschenrechnern MR 609,
- 5 Tischrechnern K 1003,
- 5 Polycomputern,
- 1 Bildschirmcomputer A 5120 und
- 1 Industrieroboter PHM 1.

Durch das Kombinat Robotron erfolgte eine laufende Modernisierung und Wartung der Erstausstattung. Durch Neuanschaffungen, wie beispielsweise Personalcomputer, Typenraddrucker, Matrixdrucker und Plotter, wurde die Technik ständig erweitert und ergänzt. Im Jahre 1989 standen den Schülern auch einige „Schneider PC“ zur Verfügung, die auf großen Umwegen angekauft worden waren. [SRZ04]

Um interessierte Schüler zu finden, warb man in allen Schulen der Stadt Dresden. Für diese war ein vierjähriger Kurs am SRZ in den Ausbildungsgebieten Informatik und Mikroelektronik vorgesehen. Als weitere Perspektive stand die Möglichkeit offen, beim VEB Kombinat Robotron eine Berufsausbildung mit Abitur aufzunehmen. Auch ein Übergang nach der 8. Klasse in die Spezialschule „Martin-Andersen-Nexö“ in Dresden bei gleichzeitigem Besuch des Schülerrechenzentrums war möglich.

Zu Beginn des Schuljahres 1983/1984 nahmen die ersten Arbeitsgemeinschaften (AG) ihre Arbeit auf. Damals bezogen sich die Themen der AG auf das Kennen lernen von Bauelementen der Elektronik und deren Zusammenwirkung in Grundschaltungen. Der Probebetrieb des SRZ begann mit etwa 50 Schülern (überwiegend der Klassenstufe 7).

1.2 Die „offizielle Eröffnung“

Das Schülerrechenzentrum nahm nach einjähriger Probezeit, mit Beginn des Schuljahres 1984/1985, offiziell seine Tätigkeit auf. Die feierliche Eröffnung vollzog sich am 15. Oktober 1984 im Seitenflügel der 10. Polytechnischen Oberschule, dem Sitz des SRZ. Das Schülerrechenzentrum wurde als „Jugendobjekt“ im Rahmen des Bezirksjugend-

objektes „Mikroelektronik und Robotertechnik“ übergeben. An der Eröffnungsveranstaltung nahmen unter anderem Hans Modrow, der damalige 1. Sekretär der SED-Bezirksleitung, Mitglieder des Rates des Bezirkes Dresden und Dr. Dieter Walter, damaliger stellvertretender Generaldirektor des VEB Kombinat Robotron teil. Nach dem feierlichen Teil erfolgte eine Besichtigung der Arbeitsräume des SRZ und des Technischen Kabinetts Mikroelektronik durch die Ehrengäste und ausgewählte Teilnehmer des SRZ. Die Schüler demonstrierten ihr Können wie heute beim jährlichen Tag der offenen Tür.

1.3 Arbeit in den achtziger Jahren

Über den laufenden Betrieb des SRZ in den Achtziger Jahren liegen kaum schriftliche Zeugnisse vor. Ein Schlaglicht wirft die Beratung des „Gesellschaftlichen Rates“ im Oktober 1987, der neben dem „Konsortium“, das zweite bedeutende Beratungsgremium des Schülerrechenzentrums bis zum Umbruch 1989/90 war. Auf dieser Sitzung wurde diskutiert, wie noch mehr geeignete Jugendliche für das SRZ begeistert werden könnten. Ziel war es, so viele begabte Schüler wie möglich aus den Schulen der Stadt an das SRZ zu führen. Zum damaligen Zeitpunkt lagen etwa 100 Neuanmeldungen für das Schülerrechenzentrum vor. Dabei wurde festgestellt, dass zu wenige Elektronik-Talente unter den Bewerbern waren. Außerdem wurde moniert, dass Lehrer, Eltern und Schüler nur unzureichend über das Schülerrechenzentrum informiert waren. Durch die Erarbeitung eines Handblattes und Pressemitteilungen über das Schülerrechenzentrum sollte ein noch besserer Bekanntheitsgrad erreicht werden. Die Probleme von damals sind den heutigen doch verblüffend ähnlich.

1.4 Das Jubiläum - 5 Jahre SRZ

Am 17. November 1988 feierte das Schülerrechenzentrum Robotron sein fünfjähriges Bestehen. Das SRZ war zwar offiziell erst am 15. Oktober 1984 eröffnet worden, doch die „Grundsteinlegung“ erfolgte ja bereits ein Jahr zuvor. Das Jubiläum wurde im Festsaal des Rathauses der Stadt Dresden mit einem großen Erfahrungsaustausch eingeleitet und am Nachmittag im SRZ mit einem Rundgang des Generaldirektors des Kombinats Robotron gemeinsam mit dem Stadtschulrat Dresdens fortgesetzt.

1.5 Die Wende und die Folgen für das SRZ

Noch vor den politischen Veränderungen im Herbst 1989 wurde im September 1989 der Abtrennungsprozess des SRZ vom Pionierpalast eingeleitet. Diese Trennung vollzog sich sowohl aus organisatorischen als auch aus inhaltlichen Gründen. Am 24. Oktober 1989 fand eine Zusammenkunft des Gesellschaftlichen Rates des Schülerrechenzentrums Robotron statt. Der Stadtschulrat gab auf dieser Sitzung bekannt, dass mit dem Schuljahr 1989/90 das SRZ direkt dem Rat der Stadt Dresden zugeordnet wird. Mitarbeiter und die Haushaltsordnung sollten vorerst weiter unter der Regie des Pionierpalastes verbleiben. Die endgültige Abtrennung des SRZ vom Pionierpalast Dresden erfolgte erst im Januar 1990. Das SRZ wurde dem Schulamt als nachgeordnete Einrichtung unterstellt und dem

Schulverwaltungsamt angeschlossen. Das Kombinat Robotron und sein Nachfolgeunternehmen finanzierte weiterhin die technische Ausrüstung und gewährleistete die entsprechenden Reparaturen und Nachrüstungen. Im Juni 1991 wurde ein Kabinett des SRZ mit PCs ausgestattet, die auf INTEL-80286-Prozessor-Plattform basierten.

1.6 Gründung des Fördervereins des SRZ e.V.

Um die weitere Entwicklung des Schülerrechenzentrums unter Mitwirkung von Schülern, deren Eltern und Freunden tatkräftig zu unterstützen, wurde am 12. April 1994 der Förderverein des Schülerrechenzentrums Dresden e.V. gegründet. Der Förderverein selbst ist Mitglied im Landesverband Sächsischer Jugendbildungswerke e.V. (LJBW).

Die Bemühungen des Fördervereins liefen bzw. laufen unter anderem auf folgende drei Schwerpunkte hinaus:

- Einwerben von Spenden und Fördermitteln,
- Unterstützung der Öffentlichkeitsarbeit (Mitwirkung an Wettbewerben und Veröffentlichungen),
- Herstellung von Kontakten mit wissenschaftlichen Einrichtungen und Firmen zur fachlichen Zusammenarbeit mit dem SRZ.

1.7 Die Neugründung

Während Bundespolitiker mehr Geld für Bildung propagierten, strich die Stadt Dresden dem Schülerrechenzentrum im Jahre 1999 für das folgende Haushaltsjahr die Gelder. Die Einrichtung stand kurz vor dem Aus. Ohne Beteiligung der Mitarbeiter und Lehrer des SRZ wurde vom Schulverwaltungsamt eine Vorlage erarbeitet, die zwangsweise zur Schließung des Schülerrechenzentrums geführt hätte. Diese sah die Weiterführung des SRZ im Rahmen der Jugendvolkshochschule vor. Dank des guten Rufes des Schülerrechenzentrums fand die bevorstehende Schließung in den regionalen Medien einen starken Widerhall.

Im letzten Moment gelang durch die Initiative des Fördervereins die Rettung des SRZ. Obwohl noch nicht alle Details geklärt waren, stand Ende Mai 2000 fest, dass das Schülerrechenzentrum in seiner ursprünglichen Form erhalten bleibt und die Technische Universität Dresden ab dem Schuljahr 2000/2001 die Trägerschaft übernimmt. Unter der Leitung des Kanzlers der TUD, Alfred Post, einigten sich Vertreter der Stadt Dresden, des Freistaates Sachsen, der Wirtschaft und der Fakultät Informatik der TUD mit dem Leiter des SRZ auf ein gemeinsames Konzept. Die Technische Universität Dresden ist laut Vereinbarung der inhaltliche Träger, die Stadt sichert den räumlichen, das Kultusministerium den personellen und private Firmen den technischen Betrieb ab.

Die neue Trägerschaft bedeutet neben politischer Unterstützung auch eine bessere Anbindung an wissenschaftliche Prozesse durch die fachliche Zusammenarbeit mit den Fakultäten Informatik, Elektrotechnik und Informationstechnik und Wirtschaftswissenschaften der TU Dresden. Die feierliche Wiedereröffnung fand am 1. Februar 2001 in den Räumen des SRZ statt. Zur Wiedereröffnung unterstützte Infineon Technologies das

SRZ mit moderner Computertechnik im Werte von 150.000 Mark. Das SRZ wurde eine Betriebseinheit der Fakultät Informatik.



Abbildung 1 - Übergabe einer Spende durch Herrn Vetter (IBM)

Im Dezember 2004 feierte das Schülerrechenzentrum (SRZ) der Technischen Universität Dresden sein 20jähriges Bestehen. Bei einer feierlichen Veranstaltung würdigte der Kanzler der TU Dresden, Herr Post, die Arbeit des Schülerrechenzentrums vor geladenen Gästen, die das SRZ über viele Jahre begleitet haben. Im Rahmen dieses SRZ-„Geburtstagsfestes" übergaben IBM Deutschland als Spende ein komplettes PC-Kabinett mit 13 Arbeitsplätzen. Die Firma AMD sponserte einen Server mit AMD Opteron sowie Rechentechnik für ein weiteres Kabinett. Ebenso beteiligte sich Thyssen Krupp an der Neuausstattung. [HEI00]

Das Schülerrechenzentrum ist heute eine Betriebseinheit der Fakultät Informatik der Technischen Universität Dresden. Für die pädagogische Arbeit sind drei Lehrer mit jeweils der Hälfte Ihrer Pflichtstunden an das SRZ abgeordnet. Durch die TU Dresden wird weiteres Personal zur Verfügung gestellt. Die gegenwärtige Unterbringung des SRZ in der ehemaligen 10. Mittelschule beruht auf einem Mietvertrag zwischen der TU Dresden und der Stadt Dresden, die eine mietfreie Überlassung der Räume beinhaltet. Durch diese Struktur findet das SRZ die Unterstützung der TU Dresden, der Stadt Dresden, des Sächsischen Kultusministeriums und verschiedener Firmen.

TECHNISCHE UNIVERSITÄT DRESDEN

S R Z

Der Förderverein des Schülerrechenzentrums e.V. unterstützt heute aktiv die Arbeit des Schülerrechenzentrums. Er hält insbesondere Kontakt zu Vertretern der Wirtschaft, um Aktivitäten anzubahnen, die das Schülerrechenzentrum befördern. In Zusammenarbeit mit verschiedenen Landesverbänden organisiert er die Einbindung des SRZ in übergeordnete Projekte.

2 Das SRZ als Institution zur Begabtenförderung

2.1 Ziele und Anliegen des SRZ

Das Schülerrechenzentrum Dresden ist seit seiner Gründung ein Zentrum zur Begabtenförderung von Schülern in den Bereichen Informatik und Elektronik. Aufbauend auf der Tradition dient es der außerunterrichtlichen Bildung in den genannten Gebieten aus der Sicht der jeweiligen Fachwissenschaften und trägt dazu bei, Wissen zu vertiefen und zu erweitern, einschlägige Arbeitsmethoden zu erlernen und zu festigen und Voraussetzungen für Studium oder Berufstätigkeit in diesen Gebieten zu verbessern.

In diesem Sinne ist das SRZ auch Konsultationszentrum für Schüler und Lehrer. Es unterstützt verschiedene Aktivitäten an den Schulen für leistungsstarke Schüler wie Facharbeiten, „Besondere Lernleistungen“, schulinterne Projekte und die Teilnahme an Wettbewerben wie „Jugend forscht“, „Bundeswettbewerb Informatik“ oder „Sächsischer Informatikwettbewerb“. Bei der Erkennung und Förderung von Begabungen berät und unterstützt das SRZ Eltern und Lehrer.

2.2 Wirksamkeit des SRZ in der Öffentlichkeit

Über die verschiedensten Formen der Öffentlichkeitsarbeit werden heute Schüler, deren Eltern und Lehrer auf das SRZ aufmerksam gemacht. Jedes Jahr im Juli findet ein Tag der offenen Tür statt. Er dient der Vorstellung der Pläne für das folgende Schuljahr und der in den Praxiskursen der Arbeitsgemeinschaften entstandenen Projektarbeiten.

Das SRZ präsentiert sich einmal im Jahr an den Fachmessen Karriere-Start (Messe für Aus- und Weiterbildung) und COMTEC (Messe für Informations- und Kommunikationstechnologien). Dabei werden herausragende Schülerarbeiten öffentlich vorgestellt. Weitere Aktivitäten fanden im letzten bzw. aktuellen Schuljahr anlässlich folgender Ereignisse statt: Woche der Informatik, Jahrestagung der Gesellschaft für Informatik, Output 2007 (TU Dresden), Uni-Tag der TU Dresden 2007, Tag der Fakultät Informatik 2007, Lange Nacht der Wissenschaften. Außerdem wurden Poster zu ausgewählten Schülerarbeiten im Sächsischen Staatsministerium für Kultus ausgestellt.

2.3 Zusammenarbeit mit Schulen und Universität

Seit dem Schuljahr 2005/2006 können Schüler, die einen der Kurse Informatik 2 oder Informatik 3 besuchen, diesen in ihre Abiturwertung an der Schule einbringen. Durch diese Konzentrierung der interessierten bzw. begabten Schüler einer Stadt wie Dresden ist es möglich, an einer Einrichtung eine Vielfalt von Möglichkeiten anzubieten, die eine einzelne Schule nicht leisten kann. Insbesondere können Informatiklehrer eine Teilnahme an einem Kurs im SRZ solchen Schülern empfehlen, die durch Vorkenntnisse und besondere Begabungen im „normalen“ Unterricht nicht oder unzureichend gefordert werden. Eine einzelne Schule hat in der Regel nicht die Möglichkeit, für solche Schüler einen geeigneten Kurs anzubieten.

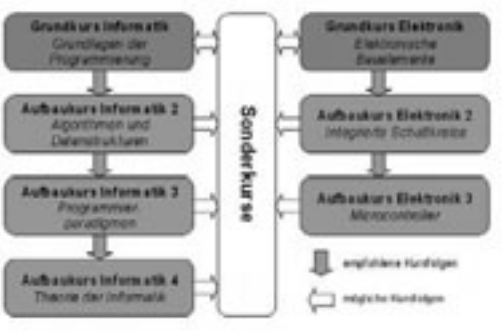
Für ausgewählte Schüler, die auch den Kurs Informatik 3 besonders erfolgreich abgeschlossen haben gibt es die Möglichkeit, an Lehrveranstaltungen der TU Dresden teilzunehmen. Durch die gute Zusammenarbeit mit den Hochschullehrern der Fakultät Informatik ist es bereits mehrmals gelungen, die organisatorischen und terminlichen Probleme zu lösen. Bei erfolgreicher Teilnahme an der zugehörigen Prüfung wird dem Schüler ein Schein ausgestellt, der bei einem späteren Studium an der TU angerechnet wird. Im Schuljahr 2006/2007 legte ein Schüler des SRZ die Prüfung zur Lehrveranstaltung „Logik I“ ab, ein anderer Schüler nimmt zurzeit an den Vorlesungen und Übungen „Algorithmen und Datenstrukturen“ teil.

Seit 1994 ist das Schülerrechenzentrum maßgeblich an der Initiierung, Organisation und Durchführung des „Sächsischen Informatikwettbewerbs“ beteiligt. Die Mitarbeiter des SRZ arbeiten sowohl im Organisationskomitee als auch in den Aufgabenkommissionen dieses Wettbewerbes mit. In diesem Schuljahr haben sich über fünftausend Schüler aus mehr als 250 Schulen aller Schularten daran beteiligt.

2.4 Inhalte und Organisationsformen

Solide Grundlagenausbildung sowie eine hohe, regelmäßig an aktuelle Entwicklungen anzupassende Themenvielfalt in den verschiedenen Kursen sind die Voraussetzung für die Realisierung der Ziele. Besonderer Wert wird darauf gelegt, dass die Schüler Einblick in die jeweilige Fachsystematik erhalten und ihr Wissen an ausgewählten Themen umzusetzen lernen.

Schwerpunkt der Informatikausbildung ist die Softwareentwicklung/ Programmierung. In der Elektronikausbildung werden Kenntnisse über die Funktion und Anwendung moderner elektronischer Bauelemente vermittelt. Die Vermittlung von Basiskenntnissen erfolgt im ersten Jahr in Grundkursen mit unterschiedlicher Schwerpunktsetzung (Algorithmik, Internet, diskrete Bauelemente).



Darauf aufbauend existiert ein breites Angebot modular aufgebauter Kurse, die eine Länge von einem halben bis zu mehreren Jahren haben. Daraus kann jeder Schüler sich ein seinen individuellen Neigungen entsprechendes Programm zusammenstellen, wobei er auf der Grundlage seiner Leistungen im Grundkurs durch die pädagogischen Mitarbeiter beraten wird. Durch unterschiedliche Wahl der Kurse ist sowohl eine vertiefte Beschäftigung mit einem Spezialgebiet möglich als auch eine breitere Auswahl verschiedener Themen. Daneben werden verschiedene Zusatzangebote (Kurzkurse, Internet-Nutzung) unterbreitet, die die kurzfristige Reaktion auf spezielle Entwicklungen und den intensiven Erfahrungsaustausch der Schüler ermöglichen.

In Anlehnung an universitäre Ausbildungsmethoden bestehen die Kurse in der Regel aus einem Theoriekurs und einer Arbeitsgemeinschaft, in der der im theoretischen Teil vermittelte Stoff praktisch umgesetzt wird. Bei der Lösung der praktischen Aufgaben werden sowohl das selbständige Handeln als auch die Fähigkeit zur Arbeit im Team gefordert und bewertet. Ein individuelles Eingehen auf jeden Teilnehmer wird durch die Arbeit in kleinen Kursen ermöglicht.

Im Laufe eines Jahres fertigt jeder Schüler im praktischen Teil eine Projektarbeit an. Während in den Grundkursen die selbständige Arbeit im Vordergrund steht, können in den höheren Kursen mehrere Schüler ein gemeinsames Projekt erstellen. Zu der Jahresarbeit gehören eine Dokumentation und die Verteidigung vor allen Schülern. Bewertet

werden die Qualität der Arbeit und der Dokumentation, die Kreativität beim Finden eines geeigneten Themas und die Verteidigung. Der erfolgreiche Abschluss eines Kurses wird mit einem Zeugnis bestätigt.

Beispiele für Themen von Jahresarbeiten des letzten Schuljahres sind LAN-Schach, Verschlüsselungsprogramm, Mensch ärgere dich nicht, Time Manager, Stundenplanmanager STUPS, Aktienkursanalyse, Siedler von Catan, Wirtschaftsrechner, LCD-Master. Einige Projekte werden zur Präsentation des Beitrages genauer vorgestellt.

Schüler, die einen der höheren Kurse erfolgreich abgeschlossen haben erhalten die Möglichkeit, ihr erworbenes Wissen als AG-Leiter eines Grundkurses zu nutzen und dabei weiter zu festigen. Sie werden dabei von den pädagogischen Mitarbeitern des SRZ angeleitet und unterstützt.

2.5 Das SRZ zwischen Kontinuität und Aktualität

Kaum eine Wissenschaft entwickelt sich derart schnell wie die Informatik. Was heute noch Zukunftsmusik ist, ist morgen schon alltäglich und übermorgen hoffnungslos veraltet. Wie aber können die Schüler von heute auf das Studium von morgen und die Aufgaben von übermorgen vorbereitet werden? Seit Jahren diskutieren Lehrer, Wissenschaftler, Praktiker und andere Personen über dieses Thema. Dabei zeichnen sich zwei Strömungen ab: Einerseits sollen die Schüler mit den aktuellsten Entwicklungen vertraut gemacht werden, darunter fallen Begriffe wie Web2.0, virtuelle Welten, Multimedia. Überhaupt scheint das Thema Internet bei dieser Richtung zu dominieren. Andererseits ist es in allen Wissenschaften üblich, gerade die Inhalte als Grundlagen zu vermitteln, die sich über einen langen Zeitraum als stabil und grundlegend erwiesen haben. Die Festlegung dieser Inhalte bezüglich der Informatik ist nicht einfach; selbst wenn man etwa die Definition der Turingmaschine als Geburtsstunde der Informatik ansieht, so ist diese Wissenschaft heute noch nicht mal 100 Jahre alt.

Sicher sind solche Fähigkeiten wie das schnelle Eingeben von Texten, das souveräne Bedienen eines Internet-Browsers oder die möglichst blinde Beherrschung des Menüs eines bestimmten Programms wichtig und nützlich, um sich heute behaupten zu können, aber genauso sicher sind sie in zehn Jahren unbrauchbar. Wer wusste schon vor vielleicht 20 Jahren, was das Internet bzw. ein Browser ist (letzteres niemand, denn das World Wide Web ist erst seit etwa 18 Jahren bekannt). Texte haben vor 30 Jahren höchstens einige Spezialisten mit dem Computer geschrieben und vor 40 Jahren saßen die Computer-Bedien-Experten noch vor schreibmaschinenähnlichen Terminals und Druckern. Wer wagt also zu sagen, was die heutige Jugend in 20, 30 oder 40 Jahren erwartet? Und wie können wir sie heute darauf vorbereiten?

Ein Vergleich der Inhalte und Zielstellungen des SRZ von 1988 und 2008 soll zeigen, dass die Schwerpunktsetzung auf bewährte und langfristig stabile Inhalte und Verfahren wie Algorithmik als inhaltlicher Schwerpunkt, problemorientiertes Herangehen und die Verwendung wissenschaftlich fundierter Arbeitsmethoden die Grundlage für den Erfolg dieser Einrichtung ist. Daneben ist natürlich die Einbeziehung aktueller Entwicklungen wichtig, dazu gehören die Einrichtung von Sonderkursen (3D-Animation mit Blender,

Web-Design mit servergestützten Datenbankanwendungen), aber auch die Verwendung elektronischer Medien und aktueller Forschungsergebnisse in Projektarbeiten.

In [AUS88] werden folgende Ziele und Schwerpunkte für den Theoriekurs und die Arbeitsgemeinschaften Informatik im Schuljahr 1988/89 festgelegt:

- „Erkennen der großen volkswirtschaftlichen Bedeutung der Informationsverarbeitung
- Erwerben der Fähigkeiten und Fertigkeiten, für Problemstellungen Datenobjekte und Lösungsalgorithmen zu entwerfen, zu beschreiben und programmtechnisch umzusetzen
- Erwerben von Kenntnissen zu Standardalgorithmen und deren Anwendung
- Erlernen einer höheren Programmiersprache bis zur aktiven Beherrschung
- Erwerb von Fertigkeiten zur selbständigen Arbeit am Rechner
- Anwendung des erworbenen Wissens und Könnens zur Lösung gesellschaftlich relevanter Problemstellungen“

Weiter heißt es in [AUS88]:

> „Für die Erreichung dieser Ziele ist die Entwicklung des algorithmisch-dynamischen Denkens der Schüler von großer Bedeutung. Andere Denkweisen wie das funktionale oder heuristische Denken sind ebenfalls weiterzuentwickeln. Als höhere Programmiersprache wird (T-)PASCAL verwendet. ... Besonderer Wert wird auf die Vermittlung einer modernen Programmiermethodik gelegt:
> - Programmentwicklung als komplexer Prozeß (von der Problemanalyse bis zur Dokumentation),
> - strukturierte Programmierung,
> - schrittweise Verfeinerung.“

Im Abschnitt „6.4 Ausbildungsprogramm des zweiten Ausbildungsjahres“ werden dann die fachlichen Inhalte aufgelistet:

- „Arbeit mit dem Pascal-Compiler
- Datentypen (Aufzählungstypen, Zeichenketten, Verbund)
- Arbeit mit selbstdefinierten Funktionen und Prozeduren
- Arbeit mit Textdateien
- Nutzung von Units“

Die Kontinuität bei der inhaltlichen Schwerpunktsetzung der wichtigsten Kurse am SRZ wird an diesem Beispiel deutlich, wenn man das Programm für den Aufbaukurs Informatik 2 von heute liest:

- „algorithmische Grundstrukturen
- einfache und strukturierte Datentypen
- Binärzahlen und Umwandlung
- technische Umsetzung mit Relais und Transistoren
- Prozeduren, Funktionen, Schnittstellen, Parameter

- Rekursion und Sortierverfahren
- Verschlüsselungsverfahren und Komprimierverfahren
- dynamische Datenstrukturen
- rekursive Algorithmen auf Listen
- Graphen und Bäume
- Algorithmen auf binären Bäumen“

Für alle Schüler war in den ersten Jahren seit der Gründung des SRZ die Teilnahme sowohl an Informatikkursen als auch an Elektronikkursen obligatorisch, da auch damals schon eine breite Ausbildung als wichtig erkannt wurde. Ebenso waren das Anfertigen einer Jahresarbeit einschließlich einer begleitenden Dokumentation und das Auftreten vor Mitschülern zur Verteidigung von Beginn an wichtiger Bestandteil der Ausbildung. Die besten Schüler konnten ihre Arbeiten bei verschiedenen Anlässen (Messen, Ausstellungen, wissenschaftliche Konferenzen) öffentlich vorstellen. Zur Bedeutung der Jahresarbeit heute heißt es in [FR06]: „Ein wichtiger Bestandteil der Arbeit ist die Durchführung und Mitarbeit an Einzel- und Gruppenprojekten. Die Schüler sollen die Möglichkeit haben, ihre Arbeit in einem größeren Rahmen vorzustellen und zu verteidigen. Dadurch wird auch das SRZ einer breiteren Öffentlichkeit bekannt gemacht und Interesse bei Schülern und Eltern geweckt.“

Auch die Beteiligung an verschiedenen Wettbewerben (Messe der Meister von Morgen, Erfinderwettbewerb und andere) war schon immer für viele Schüler eine gute Gelegenheit, ihre im SRZ gewonnen Kenntnisse und Fähigkeiten anzuwenden. Dabei errangen einzelne Schüler und auch Schülergruppen zahlreiche Urkunden und Preise. Die Teilnahme am „Sächsischen Informatikwettbewerb“ ist heute für alle SRZ-Schüler selbstverständlich, fast regelmäßig gehören auch Schüler des SRZ zu denjenigen, die für ihr erfolgreiches Abschneiden im Landeswettbewerb Urkunden und wertvoll Preise erhalten.

In [HEI07] wird die Ausbildung am SRZ im Rahmen der aktuellen Forderungen an ein Unterrichtsfach Informatik betrachtet. Im Vergleich mit den „Einheitlichen Prüfungsanforderungen Informatik“ [EPA04] wird gezeigt, dass sich die Organisationsformen und Inhalte am SRZ bewährt haben und auch unter den veränderten technischen und gesellschaftlichen Verhältnissen dazu geeignet sind, „der außerunterrichtlichen Bildung in den genannten Gebieten aus der Sicht der jeweiligen Fachwissenschaften zu dienen und dazu beizutragen, Wissen zu vertiefen und zu erweitern, einschlägige Arbeitsmethoden zu erlernen und zu festigen und Voraussetzungen für Studium oder Berufstätigkeit in diesen Gebieten zu verbessern.“ [FR06].

Solche bewährten und nachgewiesenermaßen erfolgreichen Inhalte und Formen sind

- inhaltliche, problemorientierte Herangehensweise an Aufgaben
- Kontinuität bei den inhaltlichen Schwerpunkten
- Einbeziehung aktueller Entwicklungen und Erkenntnisse in ausgewählten Zusammenhängen
- enge Verzahnung von Elektronik und Informatik
- „breites“ Wissen mit Vertiefungen an ausgewählten Inhalten

- Nutzung unterschiedlicher, jeweils problemangepasster Softwareprodukte (dazu gehört auch die variable Nutzung verschiedener Programmiersysteme)
- gemeinsame Erarbeitung theoretischer Inhalte, aber selbständige Realisierung von Projekten
- öffentliche Auftritte, Teilnahme an Wettbewerben

Literaturverzeichnis

[AUS88] Ausbildungsprogramm für die Fördergruppen des Schülerrechenzentrums robotron, Dresden, 1988 (unveröffentlicht)

[EPA04] Einheitliche Prüfungsanforderungen Informatik (Beschluss der Kultusministerkonferenz vom 01.12.1989 in der Fassung vom 05. Februar 2004)

[FR06] Friedrich, S.; Heinicke, St.; Unger, M.: Konzepte der Weiterentwicklung des SRZ Dresden, 2006 (unveröffentlicht)

[HEI00] Heinicke, St., Das Schülerrechenzentrum wurde 20. LOG In Heft Nr. 133 (2005) [AB00] Abel, K.: Bibel, U: Formatierungsrichtlinien für Tagungsbände. Format-Verlag, Bonn, 2000

[HEI03] Heinicke, St.; Timmermann, B.; Unger, M.: Konzepte der Begabtenförderung auf dem Gebiet der Informatik und ihre Umsetzung am Schülerrechenzentrum Dresden. In: Hubwieser, P.(Hrsg.): Informatische Fachkonzepte im Unterricht, Gesellschaft für Informatik, Bonn 2003

[HEI07] Heinicke, St.; Unger, M.: Kompetenzen im Informatikunterricht und ihre Umsetzung am Schülerrechenzentrum Dresden. In: Stechert, P. (Hrsg.): Informatische Bildung in der Wissensgesellschaft - Praxisband der 12. Fachtagung Informatik und Schule - INFOS 07 - Siegen 2007

[SRZ04] 20 Jahre Schülerrechenzentrum - Chronik des SRZ, TU Dresden 2004

Das Softwarehaus des VEB Kombinat Robotron

Hans-Jürgen Lodahl

Zum Stöckicht 4j
01458 Ottendorf-Okrilla
hans.lodahl@t-online.de

1 Die Vorgängereinrichtungen

Im Dezember 1961 hielt im Zentralinstitut für Automatisierung Dresden ein Abteilungsleiter des Hauses einen Vortrag über Aufbau und Nutzungsmöglichkeiten des vom VEB Elektronische Rechenmaschinen entwickelten Kleinrechners SER 2. Den etwa 30 interessierten Mitarbeitern wurde erklärt, dass die Maschine 64 Speicherplätze für ein Programm und ebenso viel Speicherplätze für Zahlen habe. Eingabe über Lochband. Die Befehle wurden einzeln verschlüsselt und in eine Liste eingetragen; dabei musste verfolgt werden, wie die Speicherplätze dann im Einzelnen belegt waren. Viele solcher Blätter ergaben das Programm, und es dauerte viele Stunden, ehe es fehlerfrei lief. Beim ZRA 1, der im Institut dann intensiv genutzt werden konnte, war dann schon vieles deutlich einfacher. 47 Jahre sind seitdem vergangen – aber vielleicht ist es doch ganz gut, man erinnert einmal an die Anfänge.

Das Institut für Datenverarbeitung (idv)

Am 03.07.1964 wurde von der Regierung der DDR das erste Datenverarbeitungsprogramm beschlossen, im gleichen Jahr wurde aus dem ZIA heraus das Institut für Datenverarbeitung in Dresden formiert und der Vereinigung Volkseigener Betriebe Datenverarbeitungs- und Büromaschinen (VVB DuB) unterstellt. Im Beschluss heißt es:

"Die VVB DuB erfüllt die ihr auf dem Gebiet der Einsatzvorbereitung und des Kundendienstes für Organisationstechnik und der Wartung des Maschinensystems für Datenverarbeitung übertragenen Aufgaben, indem sie

- *bei der Erarbeitung der wissenschaftlichen Grundlagen für die Anwendung der maschinellen Datenverarbeitung, besonders in Betrieben und VVB, mitwirkt,*
- *Musterbeispiele besonders für den Einsatz neuentwickelter datenverarbeitender Maschinen oder für die Erschließung neuer Anwendungsgebiete der maschinellen Datenverarbeitung schafft,*
- *auf der Grundlage von Verträgen den Einsatz von datenverarbeitenden Maschinen in Betrieben, VVB und Staatsorganen vorbereitet, ...*

Die Aufgaben zur Erarbeitung wissenschaftlicher Grundlagen und zur Schaffung von Musterbeispielen für die Anwendung der maschinellen Datenverarbeitung löst die VVB mit Hilfe des Instituts für Datenverarbeitung." [Be64]

Die Softwareaktivitäten konzentrierten sich im Bereich EDVA-Anwendungen auf folgende Gebiete:

1. Beratung und Betreuung von Ersteinsatzfällen des R 300 in Korrespondenz mit der Erarbeitung von Typenprojekten für verschiedene Wirtschaftszweige. 1967 wurden die Typenprojekte Maschinenbau Großserienfertigung, Chemische Industrie, Textilindustrie zur kostenlosen Nutzung übergeben. Später folgten dann Industriebau, Lebensmittelindustrie, Energieversorgung, Maschinenbau/Serienfertigung, Kraftverkehr, Information/Dokumentation.

2. Für die Lösung spezieller Aufgaben, die in einer großen Anzahl von Einrichtungen zu lösen waren, wurden Bausteine und Programme zur Unterstützung der EDVA-Anwender geschaffen:

   . Simulatoren für die Systemsimulation wie Lagerhaltung, Marktforschung, Fertigungssteuerung,
   . Mathematische Standardprogramme,
   . Programmbausteine aus dem Bereich Operations Research wie Optimierung, Statistik, Netzplantechnik,
   . Systembausteine der Datenverarbeitung wie Strukturstückliste, Arbeitszeitbilanz, Nettolohnrechnung usw.

Bei dem Einsatz von Prozessrechnern waren die Schwerpunkte bei der Entwicklung von Software:

- die Modellierung kontinuierlicher technologischer Prozesse und die Möglichkeiten ihrer Steuerung und Regelung,
- die Schaffung von Musterbeispielen mit den daraus dann abzuleitenden methodischen Unterlagen, die Erarbeitung von Programmierungshilfen und den Aufbau einer Programmbibliothek.

Einsatzbetriebe waren z. B. VEB Kombinat Schwarze Pumpe, VEB Erdölverarbeitungswerk Schwedt.

Vom idv wurden, beginnend bereits 1964, Fachtagungen zur EDV wie auch internationale Tagungen zur Thematik elektronische Rechentechnik initiiert und mit gestaltet. Das idv organisierte die Herausgabe von Fachliteratur und übernahm Herausgeberfunktionen (ab 1969 vom VEB Kombinat Robotron/ZFT fortgeführt bis 1990) z. B. für die Monatszeitschrift „rechentechnik/datenverarbeitung“ (rd) und die Schriftenreihe „Informationsverarbeitung“.

Dem Auftrag folgend, fachlich-inhaltlich zu verbesserten Bedingungen für die Schaffung von EDVA-Einsatzlösungen beizutragen, stellte das idv Informationen und Dokumentationen zur Rechentechnik zusammen. Es beteiligte sich an der Erarbeitung von Lehrschauen und Filmmaterial im Niveau der Erwachsenenqualifizierung. Von besonderer Breitenwirkung waren die vom Deutschen Fernsehfunk (DFF) 1968 bis 1970 ausgestrahlten EDV-Bildungsprogramme „Elektronische Datenverarbeitung“, die bis zu einer Prüfung von 135 000 daran interessierten Teilnehmern führten [Me06].

VEB Kombinat Robotron / Zentrum für Forschung und Technik

Am 01.04.1969 wurde der VEB Kombinat Robotron durch Fusion mehrerer Betriebe und Institute gebildet; in die damals als Großforschungszentrum (GFZ) bezeichnete FuE-Einrichtung des VEB Kombinat Robotron war auch das idv mit seiner Aufgabenstellung übernommen worden. Ende 1971 wandelte sich das GFZ vom Namen her in das Zentrum für Forschung und Technik (ZFT), ab 1974 bis 1984 dann auch als juristisch selbständiger VEB geführt. In dieser Einrichtung wurde, aufbauend auf die Ergebnisse der im idv erarbeiteten Lösungen für die EDVA R 300, die Entwicklung von Software für die EDVA, Prozess-, Klein- und Mikrorechner auf ständig steigend höherem Niveau fortgeführt.

Für den ab 1968 produzierten Rechner R 300 wurden zur Unterstützung der Programmierung neben ASSEMBLER nur MACRO-MOS und ALGOL-Compiler entwickelt. Der R 21, ESER-kompatibel ab 1972 produziert und mit dem Plattenbetriebssystem DOS/ES ausgestattet, erhielt ASSEMBLER, RPG, Basis-FORTRAN und ein PL/1-Subset. Danach wurde das Angebot an Programmiersprachen und Compilern mit der Bereitstellung neuer Rechnersysteme im Niveau OS erweitert: PL/1-Normalcompiler, FORTRAN IV, RPG, COBOL und ALGOL 60.

Programmiersprachen für Prozess-, Klein- und Mikrorechner wurden ab etwa 1969, zunächst für das Rechnersystem R 4000/4200, später für die robotron K 1600-Varianten entwickelt, jedoch waren, vornehmlich durch Echtzeitforderungen und begrenzte Ressourcen dieser Computer, der Anwendung von höheren („problemorientierten“) Programmiersprachen enge Grenzen gesetzt. Deshalb wurde längere Zeit fast ausschließlich die jeweilige (maschinenorientierte) Assemblersprache verwendet.

In Kooperation mit der Sektion Informationsverarbeitung der Technischen Universität Dresden wurde 1973 für das System Robotron 4000 ein Compiler für FORTRAN IV bereitgestellt. Für das Klein(Mikro)-Rechnersystem K 1600, Einsatz nach 1981, konnten dann auch Compiler für FORTRAN (Standard), COBOL, PASCAL, BASIC, CDL und C entwickelt werden.

Datenbank-Systeme und Software für Informationsrecherchen waren Schwerpunkte der Anwendersoftware-Entwicklung. Die Produkte erreichten die höchsten Nachnutzungszahlen und wurden teilweise mehrere Jahre nach Auflösung des Kombinates und der Entwicklungsabteilungen durch Anwender noch genutzt [He04]. Bei Datenbank-Systemen begann es um 1970 mit der Entwicklung von BASTEI („Bankspeicherung Technischer Informationen“ nach dem Vorbild BOMP von IBM, bis 1978 etwa 100 Anwendungen in der DDR und osteuropäischen Ländern), es folgten DBS/R in verschiedenen Versionen. Für Informations-Recherche-Aufgaben wurde AIDOS entwickelt und ständig weiterentwickelt.

Ausgehend von Erfahrungen mit der Entwicklung und Umsetzung von Typenprojekten R 300 wurde ab etwa 1968 begonnen, eine neue Generation von Anwendersoftware zu konzipieren und zu entwickeln, die Sachgebietsorientierten Programmiersysteme („SOPS“).

Zur Lösung mathematisch-ökonomischer und wissenschaftlich-technischer Probleme bzw. zur Lösung von Problemen des Operations Research wurde „Verfahrensorientierte Software“ bzw. „Software für mathematische Verfahren“ entwickelt, meist in Kooperation mit wissenschaftlichen Einrichtungen bzw. Universitäten. Ihre Bereitstellung erfolgte zu Beginn unter den Bezeichnungen Verfahrensorientierte Programmpakete (VOPP) / Verfahrensorientierte Programmiersysteme (VOPS), später unter den Bezeichnungen Programmpakete (PP)/ Programmiersysteme (PS). Solche Softwareprodukte liefen unter den Bezeichnungen z. B. DISKO, STATISTIK, SIMDIS, NUMATH und NPT. Anwender dafür gab es in vielen Wissenschafts- und Wirtschaftsbereichen; Anwendungen im Ausland erfolgten in Ungarn, in der CSSR, in der UdSSR, im Irak, in Indien und Kuba [JM06]. Die Bearbeitung dieser umfangreichen und langlebigen Softwarentwicklungen erforderte auch Leistungsnormative für Entwickler und eine „Software-Technologie“ mit den Komponenten Methodologie, Werkzeuge, Entwicklungsorganisation; es entstand das „Handbuch der Software-Technologie Robotron“.

Ein weiterer Arbeitsschwerpunkt der Softwareentwicklung im ZFT war die Mitwirkung an Projekten der Export-/Import-Koordinierung (EIK). Auf diese Aufgaben wird später eingegangen.

2 Das Softwarehaus VEB Robotron-Projekt Dresden (RPD)

Im Zeitraum nach 1980 wurde in der DDR zunehmend besser auch in den obersten Leitungsgremien die Bedeutung der Software für die weitere Entwicklung der „Produktivkräfte“ erkannt; im RGW-Bereich stemmte man sich aber immer noch gegen die Anerkennung von Software als Ware, für die man nicht nur die Kosten für Druck und Vervielfältigung zu zahlen hatte. Software erwies sich als Defizit-Position, und es häuften sich die Beschlüsse und Anordnungen der zentralen staatlichen Organe. Anreiz war ein Vergleich der Software-Entwicklerkapazitäten in den USA und in der DDR: Das Softwareherstellungspotenzial in den USA soll 1979 insgesamt 1,1 Millionen Personenjahre, in der DDR 15 000 Personenjahre betragen haben. Bis 1985 entwickelte sich dieses in den USA auf das 2,4-fache, in der DDR um etwa 50 %. Man konstatierte einen Mangel an Softwareherstellungskapazität und einen für die Softwareentwicklung extrem hohen finanziellen und personellen Aufwand. Beispiele für Beschlüsse/Anordnungen:

- Ministerratsbeschluss 01-111/4/85 vom 04.04.1985 zur weiteren Entwicklung und multivalenten Nutzung von Software in der DDR;
- Anordnung über die Planung, Bilanzierung und Abrechnung von Software vom 13.01.1986 (Gesetzblatt I/4 – 1986 der DDR);
- Anordnung über die Informations- und Beratungsleistungen zur Entwicklung, Produktion und Mehrfachnutzung von Software in der DDR vom 14.02.1986 (Gesetzblatt I/9 1986);
- Preisverfügung Nr. 9/86 vom 30.12.1985 über die Preisbildung von Software;
- Beschluss des Sekretariats des ZK der SED vom 15.05.1988: Maßnahmen zur weiteren Qualifizierung der staatlichen Leitung sowie zur Erhöhung der Effektivität der Softwareentwicklung und –herstellung;
- Staatliche Leitergruppe CAD/CAM-Systeme: Richtlinie zur Verwendung von höheren Programmiersprachen, Betriebssystemen und Datenbank-Betriebssystemen.

Der Einfluss der Zentrale gipfelte schließlich in einer „Gesamtvolkswirtschaftlichen Softwarekonzeption“ (Vertrauliche Verschlusssache B 153-43/89), sie bildete nach Auffassung des Ministers für Elektrotechnik und Elektronik „die abgestimmte Grundlage für die themen- und produktkonkrete Bilanzierung der Softwareherstellung mit den Volkswirtschaftsplänen und die Voraussetzung zur Sicherung von Vorhaben großer Bedeutung für die gesamte Volkswirtschaft“. Wie konkret waren die Festlegungen? Ein Beispiel: Auf Blatt 28, Tabelle 2 ist festgelegt, dass der Compiler Common LISP, lauffähig auf dem PC ES 1834 (entspricht IBM XT), im 4. Quartal 1989 an die Anwender auszuliefern ist; Entwickler Kombinat Robotron / VEB Robotron-Projekt Dresden.

Der Ministerratsbeschluss vom 4. April 1985 enthält Festlegungen zur:

- Erweiterung der Mehrfachnutzung von Software,
- Zielgerichteten Verstärkung des Potenzials für die Entwicklung, Produktion und Wartung von Software,
- Qualitätssicherung und Standardisierung,
- Erweiterung des Exports von Software,
- Planung, Bilanzierung, Abrechnung und Preisbildung

und legt die Verantwortung von Ministerien, Generaldirektoren, des Präsidenten des Amtes für Standardisierung, des Präsidenten der Akademie der Wissenschaften u. a. für die Einleitung geeigneter Maßnahmen zur Erfüllung des Ministerratsbeschlusses fest.

In der dortigen Anlage 1 wurde die Grundlinie für die Umsetzung des Beschlusses dargelegt. Im Einzelnen werden z. B. Ausführungen zur Erhöhung des Potenzials zur Entwicklung, Produktion und Wartung von Software, der Erhöhung der Produktivität der Softwareentwicklung und zur internationalen Zusammenarbeit gemacht. Die Anlage 2 enthält Festlegungen zur Arbeitsteilung zwischen den Kombinaten. So war das Kombinat Robotron z. B. für Datenbanken und Informationssysteme verantwortlich, während das Kombinat Mikroelektronik z. B. für Betriebssysteme für Mikroprozessoren und Ein-Chip-Mikrorechner zuständig wurde.

Im Ergebnis dieses Ministerratsbeschlusses wurde am 01.07.1984 der VEB Robotron Projekt Dresden gegründet. Im Rahmen des Projektes zur Industriegeschichte der Stadt Dresden von 1945 bis 1990 wurde u. a. ein Firmenportrait des VEB Robotron Projekt erarbeitet [LMM06], auf dem die folgenden Ausführungen in wesentlichen Teilen beruhen.

Das Leistungsspektrum

RPD hatte bei Betriebssystemen des ESER die in der DDR in jener Zeit nur noch in Importanlagen genutzten Ausführungen im Niveau DOS/ES entwicklerseitig zu begleiten. Gemeinsam mit der Firma VUMS/Prag wurden Compiler für alle Betriebssysteme, wie z. B. FORTRAN, COBOL, entwickelt und in den Vertrieb übergeleitet. Weiterhin wurde an der Entwicklung von wesentlichen Komponenten des Betriebssystems OS/ES unter der Gesamtregie des Entwicklerkollektivs in Karl-Marx-Stadt (Fachgebiet Geräte Karl-Marx-Stadt des FuE-Bereichs des Stammbetriebes) mitgearbeitet.

Das Leistungsspektrum der Standardsoftware umfasste insbesondere Datenbanksoftware, die der Speicherung, Verwaltung und gezielten Wiederauffindung von Daten diente. Das Datenbanksystem DBS/R (für ESER-EDVA) wurde im In- und Ausland bei mehr als 100 Kunden erfolgreich genutzt. Gleiches gilt für die DB-Lösung DABA 1600 für den Rechner robotron K 1630. Mit Oracle als Vorbild wurde das DB-System ALLDBS zur Nutzung für den 32-Bit-Rechner robotron K 1840 geschaffen. Zur Speicherung und zum Wiederauffinden unformatierter Daten wurde Software für Informationsrecherchesysteme zur Verfügung gestellt, die auch im Ausland im Zusammenhang mit Hardwarelieferungen zur Anwendung kam.

Zur Lösung von Aufgaben des Operations Research wurde verfahrensorientierte bzw. Software für mathematische Verfahren entwickelt, wie z. B. Optimierung, Statistik, Simulation und Netzplantechnik. Ausgewählte Verfahren wurden jeweils für mehrere Betriebssysteme bereitgestellt. Speziell für die vom VEB Kombinat Robotron gefertigten Personalcomputer PC 1715, A 7150 und EC 1834 sowie den Bildungscomputer BIC A 5105 wurden Programme für die Tabellenkalkulation und die Textverarbeitung geliefert. Die Entwicklung von Software durch die Anwender wurde durch die Bereitstellung der bereits im ZFT entstandenen Softwaretechnologie unterstützt.

Einen besonderen Schwerpunkt stellte die Bereitstellung von Branchensoftware dar. Für ausgewählte Branchen, wie das Hotelwesen oder Banken und Sparkassen, wurden vorgefertigte Softwarepakete bereitgestellt, die die Anwender ohne weiteren Entwicklungsaufwand nutzen konnten. So wurde die Hotelsoftware erfolgreich in den Hotels Bellevue und Dresdner Hof in Dresden und im Grand Hotel Berlin genutzt. Auch die Staatsbank der UdSSR arbeitete mit der von RPD entwickelten Software.

Zum Leistungsspektrum von RPD gehörte wie auch bereits im ZFT die Erarbeitung von Projekten und Programmen auf der Basis von Kundenaufträgen. Überwiegend handelte es sich dabei um sogenannte IEK (Import-Export Koordinierung) -Vorhaben, bei denen der Kunde in der DDR ein Gesamtpaket von Hardware, Software und Projektierungsarbeiten von einem Partner in einem nichtsozialistischen Land kaufte und dabei vereinbart wurde, dass dieser Partner möglichst viele seiner Leistungen an DDR-Firmen im Unterauftrag vergab. Die Besonderheit dieser Projekte bestand darin, dass die finanziellen Aufwendungen für den Import von Ausrüstungen aus dem „kapitalistischen Ausland“ mit den Leistungen des RPD verrechnet wurden und so eine Einsparung von Valutamitteln erreicht wurde. Die Aufträge zur Softwareentwicklung wurden sowohl für Robotron-Gerätetechnik, für programmkompatible Rechner, wie z. B. IBM, als auch für nicht kompatible Rechner realisiert. Beispiele dafür sind die Prozessführung pfannenmetallurgischer Anlagen im Edelstahlwerk Freital, die Unterstützung der Automatisierung im Düngemittelwerk Rostock und die rechnergestützte Produktionsautomatisierung im Metallgusswerk Leipzig.

Neben den IEK-Objekten wurde aber auch Software im Auftrag westdeutscher Firmen entwickelt. So entstand im RPD ein Softwareprogramm Warenwirtschaftssystem im Auftrag des Volkswagenwerkes in Wolfsburg. Bankensoftware für den Auslandszahlungsverkehr wurde im Auftrag der SITEC, einer in Westberlin ansässigen Tochter des Siemenskonzerns, entwickelt. Die Mitarbeit an diesen Projekten hatte den Vorteil, dass sich Projektleiter und Mitarbeiter Wissen zu Methoden des Projektmanagements und zu Technologien der Auftraggeber erarbeiten konnten. Zugleich erhielten sie Kenntnisse über die eingesetzte Technik und deren Nutzung. Damit erwarben sie Fachwissen, das ihre Zukunftschancen verbesserte.

Leitungsstruktur und Personalverteilung

Eine Übersicht über den strukturellen Aufbau und die Verteilung der ca. 1100 Beschäftigten zeigt die Abbildung 1.

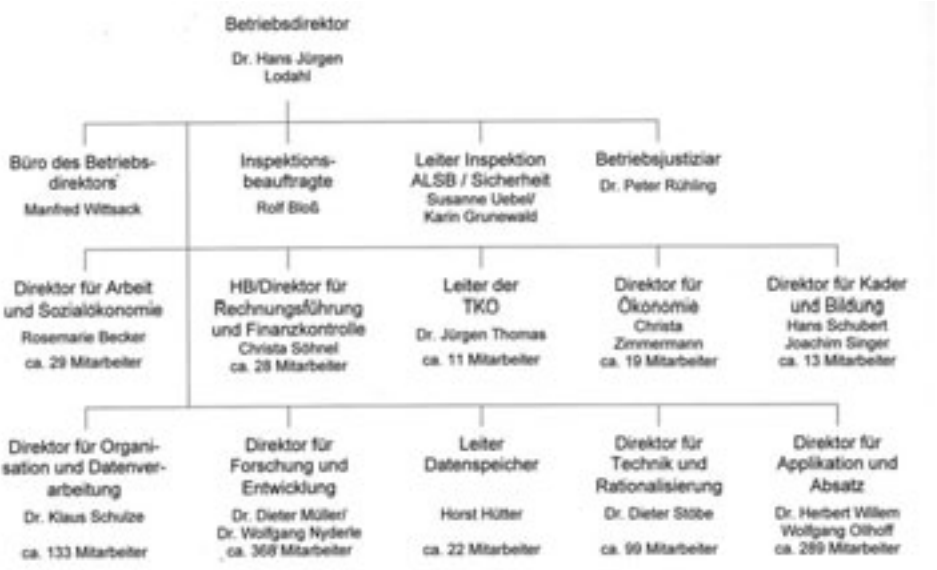


Abbildung 1: Leitungsstruktur und Personal [LMM06]

Ausstattung

Als technische Grundlage für die Entwicklung und Erprobung der Software verfügte RPD über ein leistungsstarkes Rechenzentrum, das 1988/1989 wie folgt ausgestattet war: 1 Stück ES 1055, 2 Stück ES 1057, 1 Stück ES 1040, 4 Stück K 1630, 2 Stück K 1840 (Daten hierzu siehe www.robotrontechnik.de). Hinzu kam eine umfangreiche Anzahl von Personal- bzw. Arbeitsplatzcomputern der Typen 1715, 1834 und 7150 für die Lösung der eigenen betrieblichen Aufgaben sowie die Entwicklung der Standardsoftware für die 8- und 16 Bit Technik.

Das Rechenzentrum besaß ein Mikrofilmzentrum, das neben der Bereitstellung von Literaturauszügen auch im Auftrag Dritter die Verfilmung von Dokumenten, wie z. B. die Krankenakten des Krankenhauses Rostock, durchführte. Neben Service-Aufgaben wurden auch Arbeiten zur Erprobung von Hardware-Konfigurationen durchgeführt.

Da zu den Aufgaben von RPD nicht nur die Entwicklung von Software, sondern auch die Bereitstellung der Handbücher, Dokumentationen und Disketten für die Software gehörte, verfügte RPD über eine eigene Druckerei auf der Franklinstraße in Dresden, in der ca. 60 Mitarbeiter tätig waren. Die technische Ausstattung befand sich für die damalige Zeit auf einem hohen Niveau und umfasste den gesamten technologischen Prozess von der Reprografie für die Druckvorlagen, A3- und A2-Druckmaschinen, Zusammentragstrecken, Falzmaschine, Schneidemaschine, Rückstichbindemaschine und halbautomatische Klebemaschine. Für die Vervielfältigung der Softwaredisketten waren mehrere Kopiermaschinen vorhanden, die aus dem „Nicht-Sozialistischen Wirtschaftsgebiet“ (NSW) importiert worden waren.

Kooperation

DDR: Auf dem Gebiet der Programmiersprachen erfolgte eine enge Zusammenarbeit mit der TU Dresden, der Humboldt-Universität Berlin und der Akademie der Wissenschaften der DDR. Mit der Sektion Arbeitswissenschaften der TU Dresden wurde bei der benutzerfreundlichen Gestaltung der Softwaredokumentation zusammen gearbeitet. Eine arbeitsteilige Zusammenarbeit bei der Entwicklung von Software erfolgte mit dem VEB Kombinat Datenverarbeitung, speziell mit dem Leitzentrum für Anwendungsforschung (LFA) mit Sitz in Berlin. Im Rahmen dieser Zusammenarbeit entstand eine Richtlinie für die Anwendung von Betriebssystemen, universellen Programmiersprachen und Datenbanksoftware, die den Versuch darstellte, die Entwicklung von Software und deren Anwendung effektiv zu gestalten.

UdSSR: Im März 1989 erfolgte der Abschluss eines Regierungsabkommens DDR/UdSSR über die Gründung und Tätigkeit eines gemeinsamen Produktionszentrums zur Entwicklung, Lieferung und Wartung von Software und Datenverarbeitungssystemen. Diese Firma hatte ihren Sitz in Kalinin, dem heutigen Twer, auf dem halben Weg zwischen Moskau und St. Petersburg gelegen. Der Name der gemeinsamen Firma von Zentroprogrammsistem Kalinin und RPD lautete ZENTRON. Diese Firma war das einzige Gemeinschaftsunternehmen DDR/UdSSR auf dem Gebiet der UdSSR, beschäftigte ca. 200 Mitarbeiter und hatte ein Stammkapital von 5 Millionen transverablen Rubeln, die zu je 50 % von RPD und von Zentroprogrammsistem Kalinin aufgebracht wurden.

Arbeitsschwerpunkte des gemeinsamen Betriebes waren:

- Entwicklung von Software für die Robotron-Technik,
- Ausbildung der Anwender,
- Technische Betreuung der Robotron-Technik,
- Anfertigung von Dokumentationen speziell in russischer Sprache,
- Herausgabe von Werbe- und Informationsmaterial.

Ein besonderer Schwerpunkt der gemeinsamen Arbeit war die Entwicklung des Datenbankbetriebssystems INTERBAS, mit dem das Ziel verfolgt wurde, ein Datenbanksystem als Arbeitsgrundlage für alle daran interessierten sozialistischen Länder zu schaffen.

BRD: Die Bearbeitung der Projekte für westdeutsche Auftraggeber führte dazu, dass RPD Anfang Dezember 1989 an das Ministerium für Außenhandel den Antrag stellte, das erste deutsch-deutsche Joint Venture der Firmen Data Print (Berlin/West) und dem VEB RPD zu gründen (Stammkapital 100 000 DM, Aufteilung 50/50 %, Sitz in Berlin/West).

Die Aufgaben der gemeinsamen Firma waren:

- Entwicklung und Vertrieb von Software,
- Installation von Hard- und Software.

Mit der Bereitstellung eines Devisenkredits für RPD durch die Staatsbank der DDR und dem Unterschreiben des Notarvertrages nahm die Firma im Januar 1990 ihre Arbeit auf. Das Presseecho war beträchtlich und reichte von *Frankfurter Rundschau* bis *Die Welt*. Die Zeitung *Junge Welt* machte sich Gedanken, warum der RPD-Anteil nicht wenigsten 51 % betragen würde.

Die Gründung des gemeinsamen Betriebes in Berlin/West war nicht der einzige Versuch, die Zusammenarbeit mit westdeutschen Firmen zu vertiefen. So wurden bereits zum Jahresanfang 1989 die Vorbereitungen für die Gründung eines Softwareentwicklungsstandortes bei RPD durch die SIEMENS AG getroffen. Softwareentwickler von RPD wurden bereits in Berlin/West ausgebildet, sie sollten die Keimzelle des Softwareentwicklungsstandortes bilden.

Der Neuanfang

Im November 1989 fand die letzte Betriebskonferenz des RPD statt. Die dem Referat des Betriebsdirektors folgende Diskussion war bereits durch die sich in der DDR vollziehenden Veränderungsprozesse gekennzeichnet. Die Diskussionen beschäftigten sich eingehend und überwiegend konstruktiv mit der Verbesserung der Effektivität der Arbeit von RPD und reichten von der Veränderung der Leitungsstruktur bis hin zur Gestaltung von Gehalt und Prämien. Von der Belegschaft wurde ein Bürgerkomitee gewählt, das im Dezember 1989 seine Arbeit aufnahm und der Vertrauensleutevollversammlung im Januar 1990 seine Vorschläge vorlegte. Im Februar 1990 wählten die Mitarbeiter des Betriebes einen Betriebskontrollrat (BKR), dessen Aufgabe es sein sollte, in die Rechte des staatsnahen Freien Deutschen Gewerkschaftsbundes (FDGB) einzutreten. Die staatlichen Leiter des RPD bildeten im Februar 1990 den „Sprecherausschuss der leitenden Angestellten“.

Ein weiteres Ergebnis der Betriebskonferenz vom November 1989 war die Gründung von Arbeitsgruppen, die sich mit der Entwicklung von RPD befassten. Die untersuchten Problemkreise waren:

- Wirtschaftspolitische Veränderungen
- Kader
- Leitungsstrukturen
- Stimulierung der Arbeit
- Rechenbetrieb.

Die Ergebnisse der Arbeit wurden am Runden Tisch, einer in der DDR im Zeitraum November 1989 bis März 1990 praktizierten Form des Zusammenwirkens von staatlich eingesetzten Leitern und Vertretern verschiedenster politischer Formierungen, diskutiert und ausgewertet.

Im Ergebnis des Beschlusses der Regierung der DDR vom 1. März 1990 zur Umwandlung von volkseigenen Kombinaten, Betrieben und Einrichtungen in Kapitalgesellschaften wurde im August 1990 die Robotron-Projekt Dresden Gesellschaft mit beschränkter Haftung in das Handelsregister eingetragen. Das Stammkapital betrug 5.858.000 Mark der DDR.

Die Zeit bis zur Gründung der RPD GmbH und auch danach war geprägt durch die Sorge, wie sich die weitere wirtschaftliche Entwicklung vollziehen wird und wie die ca. 1100 Arbeitsplätze gesichert werden können. Die Zeitschrift *Die Zeit* druckte am 08.12.1889 einen ganzseitigen Artikel unter der Überschrift „Fast ein sächsischer SIEMENS. Das High-Tech Kombinat aus Dresden sucht jetzt Partner im Westen“. Es stellte sich aber sehr schnell heraus, dass alle Versuche, das Kombinat Robotron als Ganzes zu retten, erfolglos sein würden. Jetzt galt es für die Leitung von RPD, selbst aktiv zu werden und die in der Vergangenheit aufgebauten Verbindungen und Kontakte zu nutzen. Es begannen Gespräche mit dem Ziel, die vorangegangenen Überlegungen zum SIEMENS-Softwareentwicklungsstandort Dresden als Grundlage zu nehmen, um eine gemeinsame Firma in Dresden zu gründen. Zum Jahresende 1989 waren die vorbereitenden Gespräche sehr weit gediehen, als die SIEMENS-Vertreter auf einer Beratung in München einen leitenden Mitarbeiter der Firma SAP vorstellten und darüber diskutiert wurde, SAP an dem zu gründenden Gemeinschaftsunternehmen zu beteiligen.

Auf der Frühjahrsmesse 1990 in Leipzig gab der Vorstand der SAP seine Zustimmung zum Einstieg der SAP in das zu gründende Unternehmen und ca. 150 Mitarbeiter von RPD fuhren ab sofort am Sonntag Nachmittag mit Bussen nach Walldorf, wurden dort mit den Grundlagen der Marktwirtschaft und den SAP-Produkten vertraut gemacht und kamen am Sonnabend Nachmittag wieder in Dresden an. Und das ein halbes Jahr lang. Das Software- und Systemhaus SRS GmbH wurde mit 320 Mitarbeitern als Gemeinschaftsunternehmen von SIEMENS, ROBOTRON und SAP und einem Stammkapital von 6 Mio. DM am 15.10.1990 notariell beurkundet. Heute firmiert SRS als SAP SI mit ca. 500 Mitarbeitern in Dresden im alleinigen Besitz der SAP AG.

Die Gründung von SRS ist nicht der einzige Erfolg, mit gut ausgebildeten, engagierten Softwareentwicklern den Neuanfang zu meistern. Parallel zu Gesprächen mit SIEMENS und SAP erfolgten Verhandlungen mit einer Reihe weiterer Unternehmen mit dem Ziel, den Mitarbeitern der RPD GmbH eine Perspektive zu bieten. Im Folgenden wird ein kurz gefasster Überblick gegeben.

- Robotron Datenbank-Software GmbH mit 26 Mitarbeitern.
  Geschäftsführer: Dr. Heinemann
  heute: erfolgreich tätig

- SQL GmbH Gründung 1992 mit ca. 25 Mitarbeitern, nach zeitweiliger Tätigkeit der Gründer bei der Sybase Germany GmbH
  Geschäftsführer: Bittner
  heute: erfolgreich tätig

- Sächsisches Druck- und Verlagshaus GmbH mit ca. 30 Mitarbeitern
  Geschäftsführer: Dr. Stöbe, Deutsch
  heute: als AG erfolgreich tätig

- Robotron Projekt GmbH mit rund 175 Mitarbeitern
  Geschäftsführer: Becker, Krönert
  1993 von Peter C. Adenauer von der Treuhand gekauft
  heute: erfolgreich tätig

- Robotron CAMA-System GmbH mit 30 Mitarbeitern, unter Beteiligung der AENI Software GmbH Köln
  Geschäftsführer: Dr. Wildenhain
  bis 1992 tätig

- Data-Print Robotron System GmbH mit ca. 30 Mitarbeitern, unter Beteiligung von Data-Print GmbH
  Geschäftsführer: Ollhoff, Böttcher
  bis 1996 tätig

- b + r Compusoft Dresden mit ca. 8 Mitarbeitern
  Geschäftsführer: Krampen, Ruef
  bis Ende 1990 tätig

- DI Schule mit 10 Mitarbeitern als Bestandteil der SIEMENS AG
  Leiter: Neels
  bis 1992 tätig.

Der Versuch, ZENTRON im Jahre 1990 der SAP anzugliedern, war nicht von Erfolg gekrönt, aber eine Reihe der russischen Mitarbeiter bildete später den Kern der Aktivitäten der SAP in Russland. Das Joint Venture in Berlin/West war ebenfalls noch eine Reihe von Jahren erfolgreich tätig.

Literaturverzeichnis

[Be64] Beschluss des Ministerrates der DDR vom 03.07.1964: Programm von Maßnahmen zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR in den Jahren 1964 – 1974. SAPMO Barch JI 2/2A/1038

[Me06] Merkel, G.: Institut für Datenverarbeitung (idv); Ein Beitrag zur Industriegeschichte der Stadt Dresden. 28 Seiten. Stadtarchiv Dresden und http://robotron.foerderverein-tsd.de.

[He04] Heinemann, R.: Geschichte der Entwicklung von Datenbank-Systemen im Kombinat Robotron; Informationsschrift „robotron-Geschichte“ der Robotron Datenbank Software GmbH Dresden; Stand 06.10.2004, 6 Seiten.

[LMM06] Lodahl, H., Merkel, G. und Müller, D.: Der VEB Robotron-Projekt Dresden. Ein Beitrag zur Industriegeschichte der Stadt Dresden. 22 Seiten. Stadtarchiv Dresden und http://robotron.foerderverein-tsd.de.

[JM06] Junge, S., Merkel. G.: Die zentrale FuE-Einrichtung des VEB Kombinat Robotron Dresden, Ein Beitrag zur Industriegeschichte der Stadt Dresden. 209 Seiten. Stadtarchiv Dresden und http://robotron.foerderverein-tsd.de.

Mit Prototyprekonstruktion zum Welthöchststand? PC-Software in den letzten Jahren der DDR

Detlev Fritsche

Technische Universität Dresden
Institut für Festkörpermechanik
01062 Dresden
fritsche@mfk.tu-dresden.de

Abstract: Die öffentliche Wahrnehmung der EDV in den letzten Jahren der DDR wurde von der Bereitstellung einer neuen Generation von Hardware geprägt: den Personalcomputern. Auch hier war man (wie auch auf anderen Gebieten) auf der Jagd nach dem im Westen verorteten Weltniveau, auch hier erlag man aber bezüglich der Software ähnlichen Fehleinschätzungen wie die westlichen Vorbilder.
Ausgehend von einer Analyse von Beiträgen in DDR-Fachzeitschriften beschäftigt sich der Vortrag mit der Entwicklung der PC-Software-Landschaft in der DDR. Aus technikhistorischer Sicht werden die Strategien etablierter EDV-Unternehmen bei der Bereitstellung von PC-Software hinterfragt, die veränderten Nutzeranforderungen beim Umzug der EDV auf den Schreibtisch der Anwender aufgezeigt und die Reaktionen der volkswirtschaftlichen Steuerungsorgane der DDR auf das dabei auftretende Defizit anhand von Archivunterlagen dargestellt.

1 Einleitung

„Welthöchststand" war ein beliebter und viel strapazierter Begriff im offiziellen Sprachgebrauch der DDR. Trotz eingeschränkter weltweiter Kommunikations- und Vergleichsmöglichkeiten diente er als virtuelle Zielmarke für technische Entwicklungen.[Fö97] [St01] Welthöchststand wurde auch für die Branche der Elektronischen Datenverarbeitung (EDV) in der DDR proklamiert und reklamiert. [K87] [1]
Wirft man einen Blick auf die Software der in den letzten Jahren der DDR hergestellten Personalcomputer, insbesondere auf die verwendete System- und Basissoftware, so fällt auf, dass große Teile davon Adaptionen westlicher Vorbilder waren. Dieser Umstand war nicht geheim oder mit einem Tabu belegt, [K87b] [2] wurde aber in Fachpublikationen nur in Nebensätzen oder Fußnoten erwähnt.[He86] [3] Die für die Anpassung verwendeten Euphemismen waren vielfältig: So sprach man von „Prototyprekonstruktion", „rechtsmängelfreier Adaption" [Ke90] oder „fremdmusterorientierter Entwicklung".
Diese Konstellation scheint zunächst nicht zum Selbstverständnis der DDR zu passen, wollte man doch um jeden Preis eine wirtschaftliche Abhängigkeit vom Westen verhin-

1 Krakat berichtete als westlicher Korrespondent von der Leipziger Messe und dem zuvor stattgefundenen Seminar Günter Mittags mit den Generaldirektoren der Kombinate, auf dem dieser forderte, neue Erzeugnisse sollten bereits zum Zeitpunkt ihrer Produktionseinführung dem Weltstand entsprechen.
2 Krakat berichtet z. B. kommentarlos über die „Anlehnung" von Systemsoftware aus der DDR an internationale Standards.
3 Heydenbluth spricht ganz offen aus, dass das Betriebssystem SCP „zu dem international weit verbreiteten Betriebssystem CP/M (Version 2.2) kompatibel ist".

dern.[Ge90] [4] Hätte man nicht gerade im Bereich der wissensintensiven Softwaretechnologie eigene, den Bedingungen in der DDR entsprechende Produkte erwarten können? Wie kam es zur Übernahme westlicher Rechner-Programme und welche Gründe hat es dafür gegeben?
Der inhaltliche Rahmen der Untersuchung wird dabei durch die Beschränkung auf kommerzielle Softwareprodukte für Büro-, Personal- und Arbeitsplatzcomputer gesetzt – im Folgenden als PC bezeichnet. Software für Großrechner, Prozessrechner oder Heimcomputer findet keine Berücksichtigung. Bei der genannten Hardware wird der Fokus wiederum auf deren System- und Basissoftware gerichtet.[5]
Untersuchungen zur DDR-Geschichte haben bisher oft unter dem Problem gelitten, dass sie im Hinblick auf das Ende der DDR das Interpretationsmuster einer Vorgeschichte des Scheiterns kultivieren. [Br96] [Ko93] Des Weiteren wird in vielen Fällen ein Vergleich mit der Bundesrepublik Deutschland gezogen, der aber nur selten angemessen scheint.[Ge94] Um diese methodischen Klippen zu umschiffen, soll der regionale Untersuchungsrahmen hier weiter gesteckt werden. Diese Vorgehensweise bietet sich auch deshalb an, weil die zentralen Entwicklungen auf dem Gebiet der PC überwiegend in den USA stattgefunden haben. [Di95] [6] Durch die Integration der Bundesrepublik in den westlichen Wirtschaftsraum waren dortige Unternehmen jederzeit in der Lage, Hard- und Software auf legalem Wege zu beschaffen bzw. mit (und ohne) Lizenz nachzubauen bzw. nach zu entwickeln. [Kr86] Eine eigenständige Entwicklung, insbesondere auf dem Gebiet der System- und Basissoftware für PC, fand nur in Ansätzen statt [FM94] und wurde auf Grund der Marktmacht US-amerikanischer Unternehmen bald aufgegeben. [M02] [7]
Im geplanten Vergleich [Ko93b] soll es darum gehen, Ähnlichkeiten zu entdecken und diese zu Thesen zu verdichten, so dass Zusammenhänge des Einsatzes von Hard- und Software in den letzten Jahren der DDR sichtbar werden. Im Rahmen des Vergleiches werden zwei Fragestellungen verfolgt: Welche Gründe gab es einerseits für die Übernahme und Anpassung westlicher Softwareprodukte durch DDR-Betriebe? Warum wurde andererseits vor allem System- und Basissoftware übernommen? Darüber hinaus beschäftigt sich eine dritte Fragestellung mit den Aktivitäten der Leitungsinstanzen der DDR-Wirtschaft: Warum gab es nur wenig selbst entwickelte System- und Basissoftware? Die Entwicklungen innerhalb der RGW-Länder, in welche teilweise mehr als die Hälfte der jährlichen PC-Produktion exportiert wurde, werden in diesem Zusammenhang nicht betrachtet. [Kr90] [8]

---

4 Geipel fasst die Aspekte, die die Entscheidungen der politischen Führung der DDR auf dem Gebiet der Informationstechnologie beeinflussten, folgendermaßen zusammen: Imperativ der Technologie, Furcht vor Abhängigkeit, Isolation innerhalb der sozialistischen Gemeinschaft, Handel mit der Sowjetunion und Streben nach Legitimität.

5 Systemsoftware ist Software, die mit dem Einschalten des PC geladen wird und für das Zusammenspiel aller Komponenten des PC sorgt. Sie wird meist als Betriebssystem bezeichnet. Basissoftware ist Software, die Lösungen für ein bestimmtes Problem anbietet und vom Anwender ohne spezielle EDV-Kenntnisse für seine spezifischen Bedürfnisse konfiguriert werden kann. Sie wird auch als Werkzeug bezeichnet.

6 Nach Dietz stellt sich die Dominanz der US-amerikanischen Unternehmen wie folgt dar: Sie verbrauchen nach Wert 40% der weltweit produzierten Software und produzieren 78% davon. Europa konsumiert 41%, trägt aber nur mit 16% zur weltweiten Erzeugung bei.

7 Mowery bringt die Schwäche der (west-) europäischen Softwareindustrie mit der Schwäche der dortigen PC-Hardwareindustrie zusammen.

8 Krakat spricht davon, dass bis zu 70% der Jahresproduktion von Robotron in den Export gingen, davon 50% in die Sowjetunion.

2 Aus dem Leben von EDV-Dinosauriern

Im Jahr 1981 brachte die Firma International Business Machines (IBM) den „Urvater“ des heute bekannten PC auf den Markt, den später so genannten PC-XT mit einer Verarbeitungsbreite von 16 Bit. [Da94] Zu dieser Zeit existierte in den USA und Westeuropa bereits ein Markt für 8-Bit-PC, der von der Firma Apple beherrscht wurde, [Li99] [9] die damit im Jahr 1981 ca. 300 Mio. US$ Umsatz machte. [Fr99]

Der Marktführer in allen anderen Bereichen der EDV war jedoch die Firma IBM – sie konnte und wollte sich das Geschäft, auch auf dem neuen und anscheinend entwicklungsfähigen Gebiet der PC, nicht entgehen lassen. Mit alten Gewohnheiten der Produktentwicklung brechend, wurde der Prototyp des IBM-PC in einer relativ unabhängigen internen Entwicklungsabteilung unter Umgehung der hierarchischen Leitungsstrukturen und unter Verwendung externer – also nicht bei IBM produzierter – Hard- und Software in verhältnismäßig kurzer Zeit fertig gestellt. [Dr92] Entgegen allen Erwartungen und Prognosen begann mit diesem PC ein einzigartiger Siegeszug. Bereits im Jahr 1983 war IBM auch in diesem Marktsegment führend. [Ch01] [10]

Rückblickend können dafür folgende Faktoren ausgemacht werden: Der Computer basierte auf einer modernen Hardwarearchitektur [Fr99b] [11] und war transparent. So wurden sämtliche Spezifika der verwendeten Hardware veröffentlicht. [Fr99c] IBM erzielte durch die Produktion in einer neuen, hoch automatisierten Fabrik und durch den Zukauf von Komponenten von anderen Unternehmen niedrige Herstellungskosten. [Dr92b] Durch eine aggressive Preispolitik am Markt ergab sich für die Käufer eine laufende Reduzierung der Preise.

Die lange Zeit der Marktdominanz, die mit der Markteinführung der Großrechnerproduktlinie System/360 im Jahre 1965 begann, hatte das Selbstverständnis der Firma entscheidend geprägt. In dieser Zeit der klimatisierten Rechenzentren, in der EDV von einer kleinen Gruppe hoch bezahlter Spezialisten an Großrechnern betrieben wurde, konnte IBM den Anwendern vorschreiben, welche Hard- und Software sie einzusetzen hatten. [Ec94] [12] Diese Vorherrschaft wurde seit 1975 mehr und mehr zum Selbstzweck. [FM94b] IBM konzentrierte sich vorwiegend auf Hardwareverbesserungen, vervollkommnete technische Parameter und lockte die Anwender mit System-Kompatibilität. [Ec94b] Trotz vereinzelter Kritik und trotz Versuchen, die Marktbeherrschung von IBM durch das Angebot steckerkompatibler Rechner (Plug Compatible Machines: PCM) aufzuweichen [Ce99], schien der ungebrochene Verkaufserfolg der Firma Recht zu geben.

Es verwundert daher nicht, dass diese Vorgehensweise auch bei der Genese des IBM-PC beibehalten wurde. [Fr94] Wieder wurden technische Parameter in den Vordergrund gestellt. An der Erstellung von System- und Basissoftware für PC beteiligte sich IBM dagegen zunächst kaum. Hatte die Firma speziell für ihre Großrechnerfamilie zuverläs-

9 Im Jahr 1981 waren bereits ca. 300 000 Apple II ausgeliefert.

10 Die Einkünfte von IBM aus dem Geschäft mit PC stiegen von 50 Mio. US$ im Jahr 1982 auf 5,5 Mrd. US$ im Jahr 1984.

11 Zunächst wurde aus Kostengründen der Prozessor I 8088 genutzt, der nur intern mit einem 16-Bit-Datenbus arbeitet, mit dem zunehmenden Markterfolg konnte auf den teueren I 8086 gewechselt werden, der auch extern mit einem 16-Bit-Datenbus arbeitete.

12 Eckbauer berichtet von einem IBM-Vertreter, der vor EDV-Beratern sagte: „Haben Sie schon einmal erlebt, dass sich ein Kunde mit eigenen Vorstellungen rührte? Wir sagen ihm, was er kaufen soll, und er richtet sich danach.“

sige und effektive Betriebssysteme selbst entworfen und hergestellt, so überließ sie das Geschäft in der PC-Sparte zunächst der damals weithin unbekannten Firma Microsoft (MS). Dieser Firma, die sich gegenüber IBM in einer subalternen Situation befand, gelang es, ein nicht selbst entwickeltes, rudimentäres, dem damals führenden System CP/M nachempfundenes Betriebssystem (MS-DOS) für den IBM-PC zu lizenzieren. Dieses Geschäft sicherte Microsoft jahrelang kontinuierliche Einnahmen und stellte die finanzielle Grundlage für den kommerziellen Aufstieg der Firma dar. [De94] IBM verkannte zunächst die Bedeutung der PC-Software. Erst als der weltweite Verkaufserfolg und die offen gelegten Spezifikationen die ersten Nachahmer von PC – allen voran die US-Firma Compaq – auf den Plan riefen, begann IBM, sich auf das bis dahin stiefmütterlich behandelte Systemsoftwaregeschäft zu besinnen. Der Hauptgrund bestand darin, dass die „latecomer“ IBM sofort bedeutende Marktanteile abjagen konnten und somit die Erlöse bei IBM sanken. [Ce99b] Auch Microsoft ging eigene Wege und befreite sich aus der Abhängigkeit von IBM.

Der erste Versuch bei IBM, mit OS/2 ein eigenes leistungsfähigeres Betriebssystem für PC zu etablieren, endete in einer Serie von gerichtlichen Auseinandersetzungen mit Microsoft sowie einer Reihe von technischen [FM94c] und marktstrategischen [Ek94] Fehlern. IBM hatte mit der Konzentration auf Hardware die Zeichen der Zeit verkannt. Microsoft gelang es mit MS-DOS, MS-Office und später MS-Windows, nicht nur dem Trend zu folgen, sondern auch eigene Maßstäbe in der EDV-Branche zu setzen.

Obwohl in Größe und Umsatz nicht direkt vergleichbar, hatte das Kombinat Robotron Dresden (KRD) als der mit Abstand größte und wichtigste Hersteller von Hard- und Software in der DDR, zumindest was den Anspruch auf Gewicht und Bedeutung angeht, eine ähnliche Stellung im DDR-Wirtschaftssystem wie IBM auf dem US-Markt. Wie IBM war auch KRD in allen EDV-Bereichen mit eigenen Produktlinien vertreten, hier allerdings auf Grund planwirtschaftlicher Entscheidungen.

Zwar stellte KRD im Unterschied zu IBM die benötigten Schaltkreise und Prozessoren nicht selbst her, dafür war das Kombinat Mikroelektronik Erfurt zuständig. [Fa98] Auf dem Gebiet der Software für alle Produktlinien musste es jedoch selbst aktiv werden. Auch dies war ein Ergebnis zentraler planwirtschaftlicher Vorgaben. Unabhängige Anbieter von Software (wie Microsoft in den USA) gab es im Wirtschaftssystem der DDR nicht.

War den Forderungen der DDR-Wirtschaft an die ausreichende Bereitstellung von EDV-Anlagen schon schwer genug nachzukommen, so kamen spezielle Probleme, die sich aus dem Nachbau westlicher Produkte ergaben, noch hinzu. Diese vielfältigen Anforderungen sowie die fehlende Konkurrenz führten dazu, dass sich auch KRD auf die Bereitstellung von Hardware konzentrierte.

Das bekamen die Anwender der ersten PC (BC 5110, 5120, 5130) ab 1980 zu spüren. Hatte KRD im Gegensatz zu IBM ein eigenes Betriebssystem zustande gebracht (SIOS), so hörte die Unterstützung der Nutzer damit auf. Die Anwender konnten zwischen zwei verschiedenen Programmiersprachen wählen (MABS und Pascal). [Kö96] Basissoftware wurde nicht mitgeliefert, alle weiteren Wünsche nach allgemeiner oder spezieller Software waren entweder vom Anwender selbst zu befriedigen oder in Auftrag zu geben. Dieses Herangehen entsprach weitgehend der gewohnten Vorgehensweise im Umgang mit Großrechnern. Gerhard Merkel, Chefkonstrukteur von KRD, bemerkte dazu 1994, dass es in der DDR zunächst nur Rechner in einer Standardkonfiguration gegeben hat, die mit minimaler Software ausgestattet waren. [Me94]

Spätestens mit der Auslieferung der nächsten PC-Generation (PC 1715) seit 1985 war auch den Verantwortlichen im DDR-Wirtschaftssystem klar, dass nunmehr die Bereitstellung von Software den zentralen Engpass bilden würde. Die Reaktion darauf erfolgte nach erprobtem Muster: 1984 wurde der VEB Robotron-Projekt Dresden (RPD) als Softwarehaus des KRD gegründet. [Lo85] Die prinzipiell neuen Bedürfnisse, die sich aus der schnellen Verbreitung des PC in der Wirtschaft ergaben, konnte aber auch RPD nicht befriedigen. Gerhard Merkel beschrieb die Situation folgendermaßen: „Die Software ... übernahm man aus Kapazitätsgründen von westlichen Herstellern und vertrieb sie unter eigenem Namen wie Redabas (dBase) und Text30 (Wordstar)." [Me94b]

3 Kisten ohne Werkzeuge

Die ersten Prozessoren der Firma Intel (1971: I 4004, 1973: I 8008) wurden nicht mit dem Ziel entwickelt, als Basis für Computer zu dienen. Ihr Einsatzzweck war wesentlich spezieller: Sie sollten in Taschenrechnern bzw. in Terminals von Großrechnern eingesetzt werden. [Fr99d] Zwar gelang es Intel in den folgenden Jahren, immer mehr diskrete Bauelemente auf einem Schaltkreis zu kombinieren[13], die dabei stark zunehmende Komplexität der zu realisierenden Logik überstieg alsbald die Möglichkeiten der jungen Firma. Daher versuchte Intel, eine Art Universalprozessor (Central Processing Unit: CPU) zu implementieren, der über einen Mikrocode, den man variabel in einem Speicher ablegen konnte, gesteuert wurde. Dieser Ansatz barg das Potential für eine der wichtigsten technischen Entwicklungen des 20. Jahrhunderts – was aber damals weder die Firma noch ihre Kunden ahnten. Es waren Bastler und Hobby-Anwender, die mit der ersten CPU (1974: I 8080) experimentierten. [Fr99e] Weitere Hersteller wie Mostek (MOS 6502) und Motorola (MC 6800) sprangen auf den gerade abfahrenden Zug auf.
Entsprechend dem Anwenderkreis waren die ersten so genannten Tischcomputer hoch technisierte Spielzeuge von Bastlern. Aus dieser Szene für Hobby-Anwender entwickelte sich schließlich ein Markt für 8-Bit-Heimcomputer, in dem Unternehmen wie Tandy Radio Shack (TRS), Commodore und nicht zuletzt Apple dominierten. [Fr99f]
Ursprünglich allein auf den Heimanwendermarkt orientiert, musste die Firma Apple um 1980 erkennen, dass ihre Computer nicht wie Fernseher oder Telefon massenhaft in die Haushalte integriert wurden. Gleichzeitig stellte sie fest, dass die so genannten Heimcomputer immer häufiger auf den Schreibtischen in Unternehmen auftauchten. [Fr99g] Dafür lassen sich heute verschiedene Erklärungen finden: Im Laufe der 1970-er Jahre hatte die Erfahrung im Umgang mit computerisierten Geräten wie elektronischen Taschenrechnern, die seit 1970 auf dem Markt waren, und elektronischen Schreibmaschinen zugenommen. Die Anwender hatten eine gewisse Übung im Umgang erlangt. Es gab darüber hinaus Software für Heimcomputer, die auch im Büro genutzt werden konnte. Die Rolle der so genannten „Killerapplikation" spielte hier das Tabellenkalkulationsprogramm VisiCalc aus dem Jahr 1979. [Li99b] Der Zugang zu den in den Rechenzentren versteckten Großrechnern war mittlerweile über Terminals vom Arbeitsplatz aus möglich. Die Büronutzer ahnten somit die Vorteile, die die dezentrale EDV am Schreibtisch bringen konnte. Die beherrschende Rolle der EDV-Abteilungen wurde aufgeweicht.

13 Der Name Intel steht für Integrated Electronics.

Vor allem der letztgenannte Aspekt sorgte für zunehmenden Unmut in den Fachabteilungen der Unternehmen. [Fr94b] Der Einfluss des Marktführers IBM hatte sich offenbar auch auf das Personal der EDV-Abteilungen ausgewirkt. Diese waren damals für sämtliche Aspekte der Datenverarbeitung innerhalb der Unternehmen zuständig, insbesondere auch für die Erstellung und Wartung der von den Fachabteilungen benötigten Software. Das Bewusstsein, eine Schlüsselposition zu belegen, erzeugte ähnliche Reaktionen wie beim Marktführer IBM. Die Anwender klagten über Rechenzentren, die nur um ihrer selbst willen existierten und sich mehr schlecht als recht um die Anwender kümmerten. [Fr94c] Sie kritisierten die langen Rechenzeiten im Stapelbetrieb, wodurch die Ergebnisse meist erst am nächsten Tag vorlagen. Zudem waren die Antwortzeiten im Dialogbetrieb problematisch, da sich viele Anwender einen Rechner teilen mussten. [Ce99c] Die schleppende Aktualisierung der Software bedeutete für die Nutzer, dass sie auf kleine Änderungen (z. B. in Eingabemasken) oft tage- oder wochenlang warten mussten. [Ec94c] Neue Projekte waren mit langen Realisierungszeiten verbunden, da die zentralen EDV-Abteilungen auf die Kundenbedürfnisse nicht adäquat reagieren konnten.
Als IBM im Jahr 1981 den PC auf den Markt brachte, sahen viele Anwender in den Fachabteilungen endlich die Möglichkeit gekommen, das von den EDV-Abteilungen verschuldete Dienstleistungsdesaster zu beenden. [Kl94] Für die Verlagerung der EDV auf den eigenen Schreibtisch war aber auch ein Preis zu zahlen, denn es verlagerte sich nicht nur die Rechenleistung in die Fachabteilungen, sondern auch die Verantwortung für die Erstellung, Wartung und Nutzung der Software. Wollte man den oben aufgeführten Problemen im Zusammenhang mit den EDV-Abteilungen entgehen, konnte eine weiterhin zentral erstellte und gepflegte Software kaum die richtige Antwort sein. Außerdem ließen Breite und Vielfalt der Probleme, für deren Lösung der PC alsbald eingesetzt werden sollte, eine zentrale Verwaltung der Software nicht zu.
Die Antwort auf die sich anbahnende Softwarekrise bestand in der Entwicklung einer neuen Art von Software: einfach zu konfigurierende Basissoftware, die der Anwender direkt am Arbeitsplatz schnell und unkompliziert einsetzen konnte. [Ch01b] Die Grundlage für die effektive Nutzung der Basissoftware wurde durch die Einführung leicht zu handhabender Betriebssysteme gelegt, die eine effektive Nutzung aller Komponenten des PC ermöglichten.

Markt-Einführung	Produkt	Charakteristik
1977	CP/M	Betriebssystem
1979	WordStar	Textverarbeitung
1979	VisiCalc	Tabellenkalkulation
1981	MS-DOS	Betriebssystem
1981	dBase II	Datenbank
1982	WordPerfect	Textverarbeitung
1983	Lotus 1-2-3	Tabellenkalkulation

Tabelle 1 Erscheinungsjahr westlicher System- und Basissoftware

Auch KRD musste sich ähnlichen Anforderungen stellen. Die Einführung des PC erfolgte wegen des Nachbaus mit zeitlichem Abstand zum Westen. Zwar wurden schon im Jahr 1980 die ersten 8-Bit-PCs (BC 5100, BC 5120, BC 5130) ausgeliefert, von einer breitenwirksamen Einführung konnte jedoch erst mit Auslieferung der nächsten 8-Bit-

Generation (PC 1715) im Jahr 1985 die Rede sein. Zu diesem Zeitpunkt war in den USA bereits der PC-AT auf dem Markt – die zeitliche Verzögerung betrug also mindestens fünf Jahre. Trotzdem schien KRD von den Bedürfnissen der Anwender überrascht worden zu sein, jedenfalls stand zunächst weder eigene noch fremde Basissoftware für den PC 1715 zur Verfügung.
In den Artikeln der DDR-Fachzeitschriften *Neue Technik im Büro, rechentechnik-datenverarbeitung* und *edv-aspekte* dominierten bis ins Jahr 1985 Berichte über den PC-Einsatz in Landwirtschaft und Industrie unter Überschriften wie:

„Futterplanung mit dem Bürocomputer 5130“ [Ch83]
„Bürocomputer robotron A 5120/A 5130 als Technologenarbeitsplatz“ [Re84]
„Einsatz des Bürocomputers robotron A 5130 in der Materialwirtschaft“ [KRZ84]
„Kapazitätsbilanzierung mit Bürocomputer robotron A 5110“ [PT84]

Die dort verwendete Software war Projektsoftware, von qualifizierten Programmierern speziell für die jeweilige Aufgabe erstellt und meist nicht in einem anderen Betrieb des gleichen Kombinats nutzbar. In Industrie und Landwirtschaft wurden die Prinzipien der Großrechnerwelt direkt auf den PC-Bereich übertragen. Diese Herangehensweise erfüllte jedoch die Erwartungen der meisten PC-Anwender nicht. Der größte Teil der PC wurde nicht direkt in Landwirtschaft und Industrie eingesetzt, sondern in der Vor- und Nachbereitung der Produktion, zu statistischen Zwecken, zur Bilanzierung und Planung, kurz für alle Tätigkeiten, die man unter dem Stichwort Dienstleistung subsumieren kann.
Seit dem Jahr 1985 mehrten sich Stimmen, die eine neue Einbeziehung des Nutzers in den Prozess der Softwareerstellung forderten. Diese kamen aus akademischen Kreisen, aber auch von Seiten der Anwender. Als Standardaufgaben, die durch geeignete Werkzeuge unterstützt werden sollten, wurden folgende Bereiche genannt: Datenerfassung, Informationsrecherche, Reportgenerierung, Tabellenkalkulation, Textverarbeitung. [PT84b]
Die wachsenden Stückzahlen ausgelieferter PC und der Druck auf die Betriebe, mittels EDV Zeit, Material und Arbeitskraft zu sparen, wirkte auf KRD zurück. Ende 1985 stellte man dort auch für den PC 1715 Basissoftware und das dazu passende Betriebssystem zur Verfügung.

(Markt-) Einführung	**Produkt**	**Charakteristik**	**Vorbild**
1984	SCP	Betriebssystem	CP/M
1984	TP/Textprozessor/Text30	Textverarbeitung	WordStar
1985	KP/Kalkulationsprozessor	Tabellenkalkulation	VisiCalc (?)
1985	Redabas	Datenbank	dBase II

Tabelle 2 Erscheinungsjahr von System- und Basissoftware in der DDR

Alle angebotenen Produkte beruhten auf westlichen Entwicklungen. [Me94c] Das spiegelte sich seit 1986 auch in den DDR-Fachzeitschriften wieder. Die bis dahin favorisierten eigenen Betriebssysteme (SIOS und UDOS) fanden kaum noch Erwähnung, es dominierten Artikel über SCP und über Basissoftware. Mit der Auslieferung der ersten Generation von 16-Bit-PC (AC 7100, 7150) im Jahr 1986 versuchte KRD noch, das

Steuer herumzureißen und selbst entwickelte Basissoftware zu verkaufen. Bei der zweiten Generation von 16-Bit-PC (EC 1834) im Jahr 1988 verzichtete man schließlich auf eigene Angebote, nun wurde ganz offiziell mit hundertprozentiger Kompatibilität zum IBM-PC geworben. [Wi88]

4 Planwirtschaft im Kampf um weiche Ware

Die ersten PC-Jahre der DDR waren von einem rudimentären Softwareangebot gekennzeichnet. Die beiden Produzenten der Bürocomputer, VEB Robotron Büromaschinenwerk Sömmerda und VEB Robotron Buchungsmaschinenwerk Karl-Marx-Stadt, konzentrierten sich auf die Bereitstellung der Hardware, was sowohl ihrem überlieferten Selbstverständnis als Büromaschinenhersteller geschuldet war, [Wa99] als auch auf die traditionelle Hardware-Orientierung deutscher Ingenieure hinweist. [Ra99] Außer zwei selbst entwickelten Betriebssystemen (SIOS, UDOS) wurden verschiedene Entwicklungssysteme ausgeliefert (MABS, Basic, Pascal). Um die Erstellung der benötigten Anwendungssoftware musste sich der Nutzer im Rahmen von Projekten selbst kümmern. Erschwerend kam hinzu, dass die erprobten administrativen Mittel und Methoden der Planung und Leitung der DDR-Volkswirtschaft [Sc99] einem scheinbar immateriellen Gut wie der Software zunächst nicht gewachsen schienen. Die planerische Hauptmethode der Bilanzierung verfolgte das Ziel, ein Gleichgewicht zwischen den verfügbaren Ressourcen und dem geplanten Bedarf herzustellen. [DIW84] Dazu war es erforderlich, exakt den notwendigen Arbeitsaufwand für die Softwareerstellung zu ermitteln. Die in den Herstellungsaufwand für Software eingehenden schöpferischen Tätigkeiten, vor allem in den ersten Phasen des Lebenszyklus, erwiesen sich jedoch als schwer normier- und berechenbar. [Be89] Die Einordnung der Erstellung von Software in die Bilanzen und damit eine Zuweisung von notwendigen Mitteln durch die Planungsorgane konnte so nicht vorgenommen werden. Die Betriebe mussten sich mit finanziellen Mitteln behelfen, die in den Fonds für Forschung und Entwicklung festgelegt waren. [Mo02] [14]

Zur Verbesserung der unbefriedigenden Situation wurde im Juli 1984 der VEB Robotron-Projekt Dresden (RPD) als Softwarehaus des KRD gegründet. RPD bekam zwei Zielvorgaben: die Entwicklung von Software auf dem Gebiet von Datenbanken, Kommunikation und Compilern sowie die Realisierung komplexer Anwendungen im Rahmen der traditionellen Projektabwicklung. Friedrich Wokurka, Generaldirektor des KRD, sagte dazu im Herbst 1984: „Im VEB Robotron-Projekt werden die wesentlichen Kapazitäten konzentriert für die Entwicklung bzw. Weiterentwicklung von Systemsoftware, wie z. B. Betriebssysteme und Compiler, und von Standardsoftware, wie beispielsweise Datenbank-, Datenkommunikationssysteme und mathematische Verfahren.“ [Wo84]

Anscheinend war die wachsende Bedeutung von System- und Basissoftware (hier als Standardsoftware bezeichnet) nun auch auf der Herstellerseite bekannt. Diese Erkenntnis führte jedoch nicht zu Eigenentwicklungen. RPD machte sich vielmehr um die „Prototyprekonstruktion“ des westlichen Datenbankprogramms dBase verdient, welches alsbald unter dem Namen Redabas ausgeliefert wurde. Diese Basissoftware zur Datenerfassung, Informationsrecherche und Reportgenerierung entsprach den Anwenderbedürfnissen offenbar sehr gut.

14 Etablierte westliche Hardwareunternehmen standen vor ähnlichen Problemen. Sie rechneten Software als Aufwendungen für Forschung und Entwicklungen ab, nicht als Investitionen. Da der Aufwand für die Herstellung (d. h. für das Kopieren) vernachlässigbar klein war, tauchte Software auch nicht in der Kalkulation auf.

Die Bereitstellung von Basissoftware auf dem Gebiet der Textverarbeitung und Tabellenkalkulation wurde den Hardwareproduzenten in Chemnitz und Sömmerda überlassen. Diese hatten auf Druck der Anwender bereits Ende 1984 das Betriebssystem SCP (eine Adaption von CP/M) und die Textverarbeitung TP (entspricht Wordstar) für die Bürocomputer bereitgestellt. Kapazitäten für die Eigenentwicklung schien man nicht aufbauen zu können. Mit der weiten Verbreitung des PC 1715 verbreiteten sich auch diese Produkte.
Dabei gab es durchaus auch gegenteilige Entwicklungen. Mit den Planungen für die nächste PC-Generation, die erstmals auf der Basis von 16-Bit-Prozessoren arbeiten sollte, wurden durch RPD Pläne für die Entwicklung so genannter „Softwarebausteine" für den Arbeitsplatzcomputer aufgestellt. [HW86] Diese wurden auch pünktlich mit dem Erscheinen des ersten Arbeitsplatzcomputers ausgeliefert und beinhalteten bis auf die Datenbank Redabas ausschließlich selbst entwickelte Programme, die entsprechend einem abgestimmten Konzept um eine zentrale Dialogsteuerung gruppiert waren. Die Zeitschrift „edv-aspekte" sprach von dem Ziel, „etappenweise ein integriertes System untereinander passfähiger Softwarelösungen zu schaffen, die sowohl als Einzelkomponenten als auch im Verbund einsetzbar sind." [He87] Zwar entsprach diese Basissoftware ziemlich genau den Forderungen der Anwender, doch war sie mit den bereits verbreiteten Programmen auf dem PC 1715 nur eingeschränkt oder gar nicht kompatibel und zudem nur auf den neuen, nur wenig verbreiteten 16-Bit-PC einsetzbar. Das schuf wenig Anreiz zum Wechsel.
Die Situation auf dem Gebiet von System- und Basissoftware für PC blieb weiterhin unbefriedigend, was vor allem institutionellen Anwendern mit Kontakten ins westliche Ausland auffallen musste. Diese waren sowohl aus eigener Anschauung als auch über internationale Publikationen mit der aktuellen Entwicklung vertraut. [Kr96] [15] In diesem Zusammenhang schrieb Volker Kempe, Direktor des Zentralinstituts für Kybernetik und Informationsprozesse (ZKI), im Januar 1986 einen Brief an Erich Honecker. Seine „Gedanken und Vorschläge zur Schlüsseltechnologie Rechentechnik" analysierten den Leistungsstand der DDR-Softwaretechnologie: „Insgesamt ist das Niveau der programmtechnischen Ausstattung von DDR-Rechnern ... weit hinter den international üblichen zurückgeblieben. … Der Übergang zur Breitenanwendung der Rechentechnik erfordert ein industriemäßiges Herangehen an die Softwareproduktion. Es müssen Betriebe und Institute für Softwareproduktion, -pflege und -vertrieb geschaffen werden, die zumindest in der ersten Zeit auf die Versorgung mit Basissoftware auszurichten sind. … Unabdingbare Voraussetzung ist jedoch die qualitätsgerechte Versorgung mit Basissoftware durch Rechnerhersteller und eine aufzubauende Softwareindustrie." [SAPMO1]
In einem weiteren Brief vom Februar 1986 unterbreitete er konkrete Vorschläge zur Aufteilung der Zuständigkeiten: „Für die Entwicklung der problemorientierten Anwendersoftware sind ... die Kombinate verantwortlich. … Betriebssystem- und Programmiersystemsoftware ist durch den Rechnerhersteller termin- und qualitätsgerecht bereitzustellen. … Anwendungsorientierte Querschnittssoftware ist schwerpunktmäßig in spezialisierten Softwareeinrichtungen zu entwickeln. Als solche ist das Kombinat Datenverarbeitung zu profilieren." [SAPMO2] Offenbar waren die Einflussmöglichkeiten

15 Von den vier Säulen der DDR-Forschungslandschaft (Akademie der Wissenschaften, Universitäten und Hochschulen, Industrie und Parteiinstitute) waren vor allem die Mitarbeiter der Akademien in der Lage, intensive Kontakte zu Kollegen und Institutionen im Westen zu pflegen.

von Kempe auf die obersten Entscheidungsebenen ausgezeichnet, denn sein Vorschlag wurde nicht zurückgewiesen, sondern sowohl durch den Minister als auch die zuständige ZK-Abteilung befürwortet. Gerhard Tautenhahn, von 1986 bis 1989 Minister für Allgemeinen Maschinen-, Landmaschinen- und Fahrzeugbau, schrieb in einer Stellungnahme zum Vorschlag von Kempe am 28. Januar 1986: „Unbestritten ist die zunehmende Bedeutung der Software. Die Grundlinie zur Bereitstellung ist durch Beschlüsse des Politbüros (CAD/CAM) und des Ministerrates festgelegt. Danach ist zentral mit Kapazitäten des Kombinat Robotron und Kombinat Datenverarbeitung die Grundsoftware bis zur Erarbeitung von Typenlösungen zu schaffen. Notwendige Entscheidungen zur Schaffung der materiellen Bedingungen sind getroffen. Zunehmende Verantwortung tragen die Anwender der Rechentechnik selbst für die Erarbeitung der anwendungsspezifischen bzw. branchenorientierten Software und nachnutzungsfähiger Typenlösungen.“ [SAPMO3] Auf besondere Gegenliebe beim zuständigen Minister traf offensichtlich der Vorschlag von Kempe, eine weitere zentrale Institution, diesmal im Rahmen des Kombinats Datenverarbeitung (KDV), zu etablieren. Als solche wurde noch im Jahr 1986 das Leitzentrum für Anwendungsforschung (LfA) als Softwarehaus des KDV gegründet. [Wa88] Aber auch die zuständige ZK-Abteilung Maschinenbau und Metallurgie äußerte am 17. Februar 1986 ihre Zustimmung zu Kempes Verteilung der Zuständigkeiten. [SAPMO4]
Das LfA erhielt genaue Zielvorgaben: So sollte es branchenorientierte Querschnittssoftware erarbeiten und an der Entwicklung von Basissoftware mitwirken. Es war die Entwicklung vielseitig nutzbarer Softwarepakete und eine Koordinierung von Nutzergemeinschaften vorgesehen. Nicht zuletzt sollte das Leitzentrum Schulungen und Beratungen durchführen. [Wi88]
Doch auch diese späte Entscheidung ebnete den Weg zur Entwicklung eigener System- und Basissoftware nicht. Die Entscheidungen zur Weiterführung des Kurses der „Prototyprekonstruktion“ waren längst getroffen worden. Karl Nendel, Staatssekretär im Ministerium für Elektrotechnik und Elektronik und Koordinator der Beschaffungslinien zum Import von Embargowaren, schrieb in einem Brief an Günter Mittag vom 6. März 1986: „Dem Vorschlag von Genossen Professor Kempe, für diese Aufgabe im VE Kombinat Datenverarbeitung eine zentralisierte Kapazität zu profilieren, entspricht die Entscheidung, in diesem Kombinat mit der Hauptaufgabe der Adaption aus dem NSW zu beschaffender Pakete einen Softwarebetrieb noch 1986 aufzubauen (ca. 80–100 Softwareentwickler).“ [SAPMO5]
Im Sommer 1987 merkte Nendel in einer Stellungnahme zur Entwicklung der Rechentechnik bis zum Jahr 2000 an, „…dass die Erhöhung des Niveaus der Software durch weiteren zielgerichteten Import von bewährter Basis- und Anwendersoftware zu gewährleisten ist.“ [SAPMO6] In seinem Eröffnungsbeitrag auf dem 4. Kongress der Informatiker der DDR im Februar 1988 in Dresden berichtete er über den erreichten Stand bei der Verteilung der Verantwortung für die Entwicklung von Software: „Wir gehen in der DDR bei der Führung der Prozesse zur Herstellung von Software von der Grundposition aus, dass für die Bereitstellung von spezieller Branchen-Software der Anwender verantwortlich ist, der zur Wahrnehmung dieser Verantwortung durch den Hersteller der Gerätetechnik, (der die Basis-Software liefert), durch zentralisierte Einrichtungen der Kombinate und Bereiche, durch Möglichkeiten zur Qualifizierung und der gegenseitigen Information zu unterstützen ist.“ [Ne88]
Das Problem der Bewertung und Einordnung von Software in den wirtschaftlichen Reproduktionsprozess schien nach seinen Worten gelöst. Software wurde in der DDR nun

als Ware mit Wert und Gebrauchswert betrachtet, die in einem Produktionsprozess hergestellt werden und daher plan- und abrechenbar sein sollte. [Ne88b] Angesichts der zentral getroffenen Entscheidungen blieb RPD Ende 1988 nur noch die Möglichkeit, den status quo zu akzeptieren: Als Betriebssysteme für Büro- und Personalcomputer wurden die CP/M-Klone SCP und DCP festgeschrieben, die Weiterentwicklung der als originäre DDR-Softwareprodukte entstandenen Betriebssysteme SIOS und UDOS wurde abgebrochen.

5 Zusammenfassung

Zusammenfassend lassen sich in Beantwortung der einleitend gestellten Fragen drei Thesen ableiten. Fragt man nach den Gründen, warum es auf dem Gebiet der PC überhaupt zur Übernahme und Anpassung westlicher Softwareprodukte durch EDV-Betriebe der DDR kam, kann man davon ausgehen, dass neben Kapazitätsengpässen und dem Prinzip des Nachbaus vor allem die Unterschätzung der Bedeutung der Software eine Rolle spielte.

Erstens lässt sich zeigen, dass sich weltweit große EDV-Unternehmen, wie z. B. IBM oder Robotron, zunächst ausschließlich auf die Entwicklung der Hardware konzentrierten. Sie hielten an tradierten Konzepten aus der Ära der Großrechner fest, als der Kreis der Anbieter und der Anwender noch klein und exklusiv war. Zu diesem Zeitpunkt war es möglich, den Kunden vorzuschreiben, welche Hard- und Software sie zu beschaffen hatten. Außer den Herstellern selbst und den EDV-Abteilungen war niemand in der Lage, derartige Entscheidungen zu beurteilen. Mit dem Aufkommen der PC versagte jedoch das Konzept der totalen Kontrolle. Die Dezentralisierung der EDV und die Öffnung der Systeme hatten einen mündigeren Anwender zur Folge. Damit konnten die Großunternehmen nicht umgehen. Sie konzentrierten sich wie gewohnt auf die Verbesserung der Hardware, die Anwender wurden mit ihren Softwareanforderungen allein gelassen. Konnten in der Marktwirtschaft kleinere Unternehmen die entstandene Lücke füllen, indem sie passende Software entwickelten und auf den Markt brachten, wurde in der Planwirtschaft mit Übernahme und Anpassung westlicher Software reagiert.

Zweitens stellten die Anwender im Zuge der Tertiärisierung der Wirtschaft neuartige Anforderungen an die Software. Hatten die Anwender einerseits mit ihrer Entscheidung für Heim- und später für Personalcomputer die Dezentralisierung der EDV erzwungen und damit die Verantwortung für ihren effektiven Einsatz übernommen, mussten sie andererseits auch auf die sich immer schneller verändernden wirtschaftlichen Zusammenhänge reagieren. Fehlende Kenntnisse und Kapazitäten erlaubten es ihnen nicht, Anwendungen selbst zu programmieren und zu pflegen. In eine Abhängigkeit von den zentralen EDV-Abteilungen wollten sie jedoch auch nicht mehr geraten. Sie benötigten daher allgemeine, für definierte Zwecke erstellte, mit geringem Aufwand anpassbare und nicht an eine bestimmte Hardware gebundene Software. Diese Anforderungen wurden durch die sich etablierende System- und Basissoftware befriedigt.

Drittens konnten die Steuerungsinstanzen der Planwirtschaft auf dem Gebiet der Softwaretechnologie nur reagieren, nicht agieren. Wurde das Softwareproblem zu Beginn der PC-Zeit fast völlig negiert und kam es in zentralen Entscheidungen kaum vor, so erkannte und reagierte man später mit der Gründung von Softwarehäusern durchaus auf die entstandenen Defizite. Dabei orientierte man sich sowohl an tradierten Konzepten aus der Ära der Großrechner als auch an internationalen Entwicklungen. Der Gestaltungs-

rahmen der Wirtschaftsordnung wurde nicht verlassen, eine eigenständige Einordnung in die Planung und Leitung konnte auf Grund prinzipieller theoretischer Probleme nicht erfolgen. Damit war eine vorausschauende und angemessene Erstellung eigener System- und Basissoftware nicht möglich. Man reagierte mit der Übernahme und Anpassung westlicher Software. Den Anwendern war es egal – sie benötigten vor allem geeignete Software für die Lösung ihrer Probleme. Späte Ansätze auf Seiten der Hersteller konnten sich nicht mehr durchsetzen.

Die Literaturliste kann beim Verfasser angefordert werden.

Von den Anfängen der objektorientierten Softwareentwicklung an der TU Ilmenau bis zu einem Ansatz für ein komponentenorientiertes Softwaresystem

Reinhold Schönefeld

Schulung & Beratung für Software
Naumannstr. 9a
98693 Ilmenau
schenski@t-online.de

Abstract: In der Arbeit wird kurz die Geschichte der objektorientierten Softwareentwicklung an der TH Ilmenau aufgezeigt, um dann auf die durch Kompositionsabstraktion erreichten Eigenschaften eines Objektes einzugehen. Der Hauptteil beschäftigt sich damit, wie durch systemtheoretische Konzepte ein Ansatz für die Entwicklung von komponentenorientierter Software aus den in der DDR begonnenen Untersuchungen zur Objektorientierung entstand.

1 Kurzer Abriss zur Geschichte der objektorientierten Softwareentwicklung an der TH Ilmenau in den 80-er Jahren

Anfang der 80-er Jahre wurde im Rechenzentrum der TH Ilmenau in Zusammenarbeit mit dem Kombinat Robotron Dresden am Betriebssystem MUTOS (Multi User Time Sharing Operating System) für die Klein- und Mikrorechner der DDR gearbeitet. Das MUTOS war kompatibel zum UNIX. Im Rahmen dieser Forschungs- und Entwicklungsarbeiten erschien eine erste Veröffentlichung „Objektorientierte Betriebssysteme – ihre Implementierung auf Kleinrechnern“ [SCHE84] aus dem Rechenzentrum, in der der Grundgedanke der Objektorientierung erstmalig eine Rolle spielte.

Kurze Zeit später konnten wir der internationalen Literatur entnehmen, dass die Wurzeln des objektorientierten Paradigmas bereits 1967 in der Programmiersprache SIMULA 67 zu finden waren und dass verschiedene Arbeiten zu Abstrakten Datentypen von Anfang der 70-er Jahre und auch die Module von D. L. Parnas aus dieser Zeit zum Themenkreis der Objektorientierung von Software gehörten. Die objektorientierte Programmiersprache Smalltalk 80 aus den USA war ab 1986 an der TH Ilmenau auf Kleinrechnern verfügbar und bis zum Herbst 1989 wurden an der TH Ilmenau bereits Vorlesungen zur Programmierung in C, C++ und Smalltalk gehalten. Eine „Studie zum objektorientierten Softwareentwurf“ als Forschungsauftrag für VEB Robotron-Projekt Dresden durch das Rechenzentrum der TH Ilmenau spiegelte den damaligen Wissensstand in Ilmenau im Jahre 1989 wider. Das Literaturverzeichnis enthält weitere Arbeiten aus dieser Zeit.

2 Das objektorientierte Paradigma aus systemtheoretischer Sicht

Ein Objekt bzw. ein Objekttyp (Klasse) ist bekanntlich die Zusammenfassung (Komposition) von Funktionen und Daten (Datenobjekte) zu einer neuen integrierten Softwareeinheit (Abb.1). In einem Objekt (Klasse) hängen die Funktionen und Daten von einander ab. Sie erlangen beide bestimmte Eigenschaften, die erst durch ihre Komposition zum Objekt erzeugt werden und das Objekt selbst hat bestimmte Eigenschaften, die nicht unmittelbar in den Funktionen und Daten zu finden sind. In der Systemtheorie wird von Emergenz neuer Eigenschaften gesprochen, die durch die Komposition verursacht werden.

Welche neuen Eigenschaften erlangen die Funktionen und Daten durch die Komposition zu einem Objekt? Die Funktionen dürfen direkt auf die Daten zugreifen und alle Funktionen zusammen bilden mit ihren Signaturen die Schnittstelle des Objekts zu seiner Umgebung. Andere Funktionen, außerhalb des Objekts, haben diese Eigenschaften nicht. Die Daten wiederum werden im Objekt zu Trägern des Objektzustandes. Sie spannen einen Zustandsraum des Objekts auf. Für sich betrachtet haben die Daten diese Eigenschaft nicht.

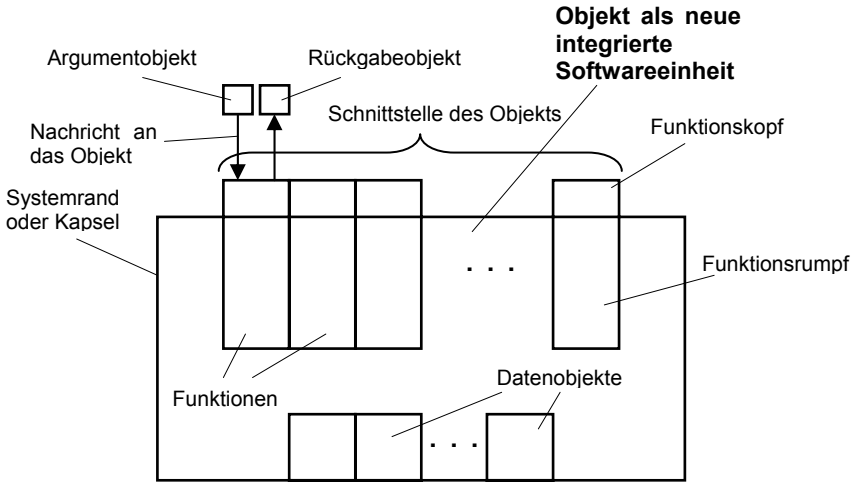


Abbildung 1: Sinnbild eines Objekts als Komposition von Funktionen und Datenobjekten

Welche neuen Eigenschaften bekommt ein Objekt durch die Komposition? Das Objekt als neues Ganzes besitzt eine Identität und zeigt ein äußeres Eingabe-Ausgabe-Verhalten an seiner Schnittstelle zur Umgebung. Dieses Verhalten ist nicht allein durch das Verhalten der einzelnen Funktionen zu erklären, sondern hängt auch von der Lage des Zustandsvektors im Zustandsraum des Objekts ab. Eine Funktion für sich hat für denselben Eingabewert den gleichen Ausgabewert. Das Verhalten des Objekts ist eine neue Qualität, es wird durch die Menge aller Eingabefolgen (Nachrichtenfolgen) an das Objekt und die dazugehörigen Ausgaben (Nachrichten an die Umgebung) während des Lebenszyklus eines Objekts gebildet. Dabei müssen auch wiederholt auftretende

Nachrichten in den Folgen berücksichtigt werden. Für dieselbe Nachricht hat ein Objekt nicht zwangsläufig das gleiche Rückgabeobjekt. Sein Verhalten zeigt ein Objekt an seiner Schnittstelle, die lediglich eine Passfähigkeit für die Nachrichten darstellt, die das Objekt empfangen kann, aber nicht das Verhalten selbst beschreibt. Die Ansicht, dass die statische Schnittstelle das dynamische Verhalten des Objekts ausdrückt, ist in der Literatur leider öfter zu beobachten.

Das Verhalten eines Objekts ist nicht einfach zu ermitteln. Für ein bereits existierendes Objekt entsteht es durch Überlagerung in den Funktionen des Objekts und wird durch die fundamentalen Steuerkonstrukte (Sequenz, Alternative, Iteration) und die aktuellen Werte der Zustandsvariablen verursacht. Das Verhalten ist durch die so genannte Programmfunktion mathematisch beschreibbar. Sie kann aus dem Programmcode oder dem graphisch dargestellten Programmablaufplan extrahiert werden (s. Kap. 4.2). Existiert ein Objekt noch nicht, kann sein gefordertes Verhalten durch die so genannte Sequenzbasierte Spezifikation ermittelt werden, indem die Nachrichtenfolgen oder Sequenzen systematisch durch Kombination erzeugt und den Ausgabewerten zugeordnet werden [Sch07, Kap. 6].

Systemtheoretisch betrachtet ist ein Objekt Ausdruck des Holismus, der Ganzheitslehre (Konzept erstmals bei J. Smuts 1926), indem ein System in seiner Wechselwirkung mit der Umgebung als Ganzes betrachtet wird und dabei die Teile, aus denen das System besteht, aus Sicht der Umgebung keine Rolle spielen. Der Holismus resultiert aus der Kompositionsabstraktion. In der ganzheitlichen Betrachtungsweise gehen die Eigenschaften des Systems nicht unmittelbar aus den Eigenschaften der Teile hervor, sondern „das Ganze ist mehr als die Summe seiner Teile“ (Aristoteles 384-322 v. Chr.). Durch Komposition von Teilen treten, wie schon erwähnt, vor allem bei komplexen Systemen emergente Eigenschaften auf, die nicht in den Teilen für sich betrachtet zu finden sind. Das Konzept der Abstrakten Datentypen, auch das der Softwaremodule und das der Endlichen Automaten entsprechen dem Konzept des Holismus der Systemtheorie.

3 Die Bedeutung der Objekte für die Softwareentwicklung der 90-er Jahre

Das Interesse der praktischen Softwareentwicklung an Objekten und Klassen war Ende der 80-er Jahre deshalb geweckt, weil viele reale Dinge oder Objekte der Wirtschaft und anderer Problemdomänen genau solche Objekte mit Identität, Verhalten und Zustand zu verwalten hatten. Denken wir dabei an die bekannten Businessobjekte ein Konto, einVertrag, einePerson usw. Der Entwurf eines Softwaresystems wurde durch das Objektkonzept vereinfacht, da ein Softwareentwickler die realen Objekte möglichst in entsprechende Softwareobjekte übertrug. Ein objektorientiertes Programm war durch die angewandte Kompositionsabstraktion bei der Objekt- bzw. Klassenbildung weitaus weniger komplex als ein vergleichbares strukturiertes oder gar unstrukturiertes Programm. Die Objekte (Objekttypen) wurden, ähnlich wie die Zellen durch die Natur, als elementare Bausteine in der Softwareentwicklung eingesetzt.

Neben der Reduktion der Komplexität objektorientierter Programme interessierte die Praxis vor allem die Wiederverwendung einmal entworfener Objekttypen (Klassen) in weiteren Programmen. Dazu dienten Klassenbibliotheken und das Konzept der Vererbung, das auf der Basis der bekannten Generalisierungsabstraktion zum Aufbau von Klassenhierarchien führte. Der damit einhergehende Polymorphismus war ein weiteres Konzept, was Flexibilität und mehr Möglichkeiten zur Wiederverwendung bot. Die 90-er Jahre zeigten jedoch, dass die Wiederverwendung von Objekten nicht den erhofften Erfolg bei der Steigerung der Arbeitsproduktivität und der besseren Beherrschbarkeit bei der Entwicklung komplexer objektorientierter Programme brachte. Woran lag das? Was kann uns die Geschichte hierzu lehren?

4 Ansatz für eine komponentenorientierte Softwareentwicklung

Aus heutiger Sicht hätte das systemtheoretische Konzept der Kompositionsabstraktion damals wiederholt angewandt werden müssen, um nicht nur Daten und Funktionen, sondern unterschiedliche Objekte zu einer weiteren neuen integrierten Softwareeinheit zusammen zu fassen. Diese Einheit wollen wir als Verbundkomponente bezeichnen (Abb. 2).

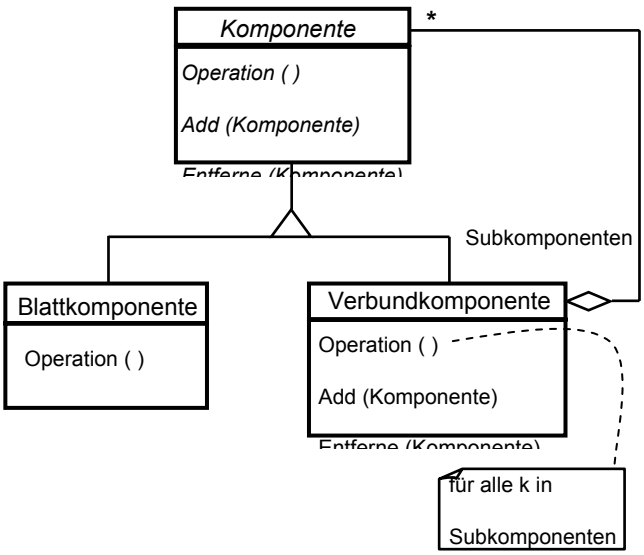


Abbildung 2: Kompositionsmuster zur Darstellung von Komponentenhierarchien

Wiederholt angewandt heißt, es können auch Verbundkomponenten und Objekte oder nur Verbundkomponenten zu einer neuen Verbundkomponente zusammengefasst werden. Eine Verbundkomponente setzt sich somit aus weiteren Verbundkomponenten und Objekten in einer hierarchischen Struktur zusammen, einer Komponentenhierarchie (Abb. 3). An der Spitze (Wurzel) befindet sich eine i. d. R. komplexe Verbundkomponente, die das gesamte zu bauende Softwaresystem darstellt und auf den jeweils niedrigsten Ebenen befinden sich ausschließlich Blattkomponenten, die einfache Objekte sind. Dazwischen können weitere Verbundkomponenten angeordnet sein. Wichtig ist, dass erkannt wird, dass die Kompositionsabstraktion nicht nur auf Funktionen und Daten im Objekt angewandt werden kann, sondern jetzt auf alle Teilsysteme in der Komponentenhierarchie in einheitlicher Weise anwendbar ist. Dabei wirkt die Kompositionsabstraktion, mit ihren durch Emergenz verursachten zusätzlichen Eigenschaften, bei jeder Verbundkomponente und ihren Teilkomponenten in gleicher Weise, wie bei den Objekten (s. Kap. 2) beschrieben. Jede so gebildete Verbundkomponente hat eine eigene Identität, zeigt ein äußeres Verhalten an ihrer Schnittstelle und besitzt einen inneren Zustand in ihrem Zustandsraum.

Zur Darstellung der hierarchischen Verhältnisse von Komponenten eignet sich das Kompositionsmuster aus der Sammlung der Entwurfsmuster für Softwaresysteme [GAM95] in bester Weise. Abbildung 2 zeigt den rekursiven Zusammenhang zwischen den Klassen der Komponenten. Die abstrakte Klasse *Komponente* vererbt ihre Funktionen (Methoden) und Daten (Attribute) an die konkreten Klassen Blattkomponente und Verbundkomponente. Eine Verbundkomponente setzt sich aus beliebig vielen (*) weiteren Subkomponenten zusammen, die entweder wieder Verbund- oder Blattkomponenten sind. Die Rekursion bricht ab, wenn nur noch Blattkomponenten (Objekte) auftreten. Eine aus dem Kompositionsmuster abgeleitete beispielhafte Komponentenhierarchie ist in Abbildung 3 dargestellt.

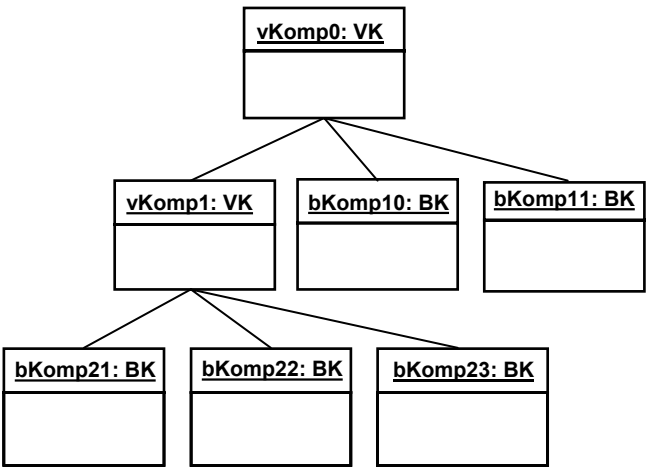


Abbildung 3: Komponentenhierarchie aus Verbund- und Blattkomponenten

4.1 Die Vorgehensweise beim Aufbau komponentenorientierter Software

Für den Aufbau eines komponentenorientierten Softwaresystems können zwei unterschiedliche Ausgangsituationen vorliegen:

1. Für ein solches Softwaresystem sind nur die informalen Anforderungen bekannt, jedoch keine vorbereiteten Komponenten vorhanden.

2. Das Softwaresystem existiert ebenfalls nicht, soll aber aus vorhandenen Komponenten zusammengebaut werden.

In beiden Fällen kann für die Entwicklung die so genannte Box-Struktur-Methode (BSM) zum Einsatz kommen [SCH07, Kap.5]. Die BSM ist die Anwendung der Systemtheorie auf ein Softwaresystem, indem es durch drei Sichten beschrieben wird. Die Black-Box-Sicht (BB-Sicht) oder äußere Verhaltenssicht, die State-Box-Sicht (SB-Sicht) oder Datensicht und die Clear-Box-Sicht (SB-Sicht) oder Architektursicht. Durch Anwendung der BSM wird ein Softwaresystem zunächst als Ganzes betrachtet und dann durch eine systematisch betriebene Dekomposition in Komponenten zerlegt. Die BSM geht auf Harlan D. Mills (1919-1996) zurück und ist Teil des Cleanroom Software Engineering (s. dazu [SCH07]).

Die BB-Sicht beschreibt in formaler Weise das geforderte funktionale äußere Verhalten des gesamten Softwaresystems in Form einer tabellarisch repräsentierten Funktion und ist die mathematisch formulierte Spezifikation des Softwaresystems. Die BB-Funktion beschreibt in abstrakter Weise das Eingabe-Ausgabe-Verhalten (die Dienste), die das Softwaresystem als zusammengesetztes Ganzes ausführen soll und nach seiner korrekten Implementierung auch ausführen kann.

Die SB-Sicht beschreibt ebenfalls in formaler Weise das äußere Verhalten (die Dienste) des Softwaresystems, jedoch unter Einführung von inneren Zustandsgrößen. Die SB-Sicht ist die bekannte Darstellung eines endlichen Automaten (auch Zustandsmaschine) in Form einer tabellarisch notierten Funktion. Die BB- als auch die SB-Sicht werden durch die Methode der Sequenzbasierten Spezifikation [SCH07, Kap. 6] aus den auf dem Rand des Softwaresystems eingehenden Stimuli (Eingaben) und den dort verlangten Reaktionen (Ausgaben) durch eine systematisch betriebene Auflistung von Stimulussequenzen (Stimulusgeschichten) und durch ihre Zuordnung zu den geforderten Reaktionen gefunden.

Aus der Auflistung gehen für das Softwaresystem charakteristische Sequenzen hervor, die so genannten kanonischen Sequenzen. Aus diesen lassen sich die Zustandsgrößen des Softwaresystems herleiten, die die Daten für das Softwaresystem, also der Verbundkomponente auf der obersten Ebene der Komponentenhierarchie, sind. Die Zustandsgrößen akkumulieren die auf dem Rand des Softwaresystems eintreffenden Stimuli und spannen je nach Anzahl der Stimuli einen mehrdimensionalen Zustandsraum auf. Die Menge der kanonischen Sequenzen und damit der unterscheidbaren Zustandsgrößen und ihrer Werte ist ein Maß für die Komplexität des Softwaresystems.

Sind die Zustände bekannt, kann die SB-Funktion des zu bauenden Softwaresystems als Zustandsmaschine ermittelt werden, die alle geforderten Transitionen enthält, die das zu bauende Softwaresystem ausführen können muss. Eine Transition ist ein Abbildungspaar in der SB-Funktion, das einem aktuellen Stimulus und dem existierenden Vorzustand eine Reaktion und einen Nachzustand zuordnet. Die SB-Funktion kann auch als geforderte Übertragungsfunktion des Softwaresystems angesehen werden, die angibt, wie unter Beachtung des Zustandes alle eintreffenden Stimuli in Reaktionen zu übertragen sind. Die SB-Funktion gibt somit vor, wie die Architektur und die daraus resultierende Implementierung aufgebaut sein müssen, um das gleiche Übertragungsverhalten wie die SB-Funktion zu haben. Die SB-Funktion wird deshalb auch als Vorgabefunktion bezeichnet.

Die CB-Sicht des Softwaresystems ist seine Architektur in Form von Komponenten (Blatt- und Verbundkomponenten), die über die fundamentalen Steuerstrukturen Sequenz, Alternative und Iteration verknüpft sind. Die Architektur ist die Verbundkomponente auf der obersten Ebene („das neue Ganze“), die sich aus den Komponenten auf der nächst niederen Ebene zusammensetzt, und wie sie in der obersten Ebene über die Steuerstrukturen verknüpft sind. Das Entwurfsproblem dabei ist bekanntlich, welche Komponenten mit welchem Verhalten benutze ich und mit welchen Steuerstrukturen verknüpfe ich sie, um mit der daraus resultierenden Architektur genau das geforderte Übertragungsverhalten zu erzeugen, das durch die Vorgabefunktion gefordert wird. Es geht also darum, eine Funktion für das Übertragungsverhalten der Architektur aus einem ersten Ansatz für die Architektur zu finden. Die so gewählte Definition der Architektur eines Softwaresystems besitzt den Vorteil der Verifizierbarkeit und Zerlegbarkeit. Verifizierbar, indem sie auf Korrektheit gegenüber der Vorgabefunktion überprüft werden kann. Zerlegbar, da die Komponenten auch über die gleichen fundamentalen Steuerkonstrukte verknüpft sind wie die elementaren Anweisungen innerhalb des Codes der Komponenten, die letztendlich durch fortgesetzte Zerlegung (schrittweise Verfeinerung) aus den Komponenten entstehen.

In der o. g. ersten Ausgangsituation, wo keine vorbereiteten Komponenten verfügbar sind, müssen diese erst entworfen werden. Dabei bieten die Transitionen der SB-Funktion einschließlich der gefundenen Zustandsgrößen wichtige Informationen für den ersten Ansatz solcher Komponenten. Das hauptsächlich eingesetzte Entwurfsprinzip ist das der „Kapselung der Zustandsgrößen“. Das bedeutet, bilde für eine oder für mehrere Zustandsgrößen zunächst eine Blattkomponente, in der als Daten die Zustandsgrößen eingehen. Als Funktionen werden solche eingeführt, die einen Teil der geforderten Transitionen der Vorgabefunktion ausführen können. Dabei ist die Abgeschlossenheit der Transitionen für die entworfene Komponente ein hilfreiches Kriterium. Die mit den Funktionen ausführbaren Transitionen müssen notwendig und hinreichend sein für die Bildung und Modifikation der gekapselten Zustandsgrößen und die Erzeugung der Reaktionen. Die Zustandsgrößen ihrerseits müssen notwendig und hinreichend sein für die Ausführung aller Transitionen der Blattkomponente.

Sind alle Blattkomponenten der Architektur entworfen worden, müssen sie geeignet zur Architektur über Steuerkonstrukte verknüpft werden. Die so erzeugte Ausgangsarchitektur ist entweder in einer Entwurfssprache (z. B. Pseudocode) oder als Box-Struktur-Diagramm notiert, das einem Ablaufplan für ein komponentenorientiertes Programm entspricht. Aus beiden lässt sich die Übertragungsfunktion der Architektur extrahieren, die auch Architekturfunktion genannt werden soll, und die in Form einer mit der Vorgabefunktion vergleichbaren Tabelle notiert ist. Die Architekturfunktion ist für die Ausgangsarchitektur erwartungsgemäß nicht in Übereinstimmung mit der Vorgabefunktion. Wie es möglich ist, die Architekturfunktion zu extrahieren und wie sie an die Vorgabefunktion angepasst werden kann, wird in Kap. 4.2 gezeigt.

Zunächst soll noch ein anderer Weg aufgezeigt werden, wie die Komponenten für den ersten Ansatz der Architektur gewonnen werden können. Das ist der Weg über die Spezifikation des geforderten Verhaltens für jede Komponente aus den informalen Anforderungen heraus. Es werden informale Anforderungen für die beabsichtigten Komponenten aufgestellt und für diese die BB- und SB-Funktion über eine Auflistung für jede Komponente ermittelt. Damit ist das Verhalten jeder Komponente zunächst nur in abstrakter Weise als BB- oder SB-Funktion gegeben. Das genügt aber, um die Architekturfunktion aus der Verknüpfung der Komponenten extrahieren zu können.

In der oben genannten zweiten Ausgangssituation, wo schon fertige Komponenten vorhanden sind, müssen diese mit einer exakten Verhaltensbeschreibung als BB- oder SB-Funktion geliefert werden. Denn nur so können sie zu einem ersten Ansatz für eine Architektur verknüpft werden. Ist eine explizite Verhaltensbeschreibung nicht vorhanden, muss sie zunächst aus der Architektur der jeweiligen Komponente oder aus ihrem Code extrahiert werden.

4.2 Extraktion der Übertragungsfunktion eines Softwaresystems

Jedes Programm (Softwaresystem) in Form seines Programmablaufs ist mathematisch betrachtet eine graphische oder prozedurale Regel, die einer Funktion oder Relation äquivalent ist. Das Ziel der Extraktion ist es diese Übertragungsfunktion, die das Eingabe-Ausgabeverhalten beschreibt, frei von graphischen oder prozeduralen Ausdrucksmitteln, rein als mathematische Funktion darzustellen. Diese Funktion wird als Programmfunktion bezeichnet. Die Architekturfunktion im speziellen ist die Programmfunktion für eine Verbundkomponente auf der obersten Hierarchieebene eines komponentenorientierten Softwaresystems. Die entscheidende Idee zur erfolgreichen Extraktion der Programmfunktion ist die Tatsache, dass die fundamentalen Steuerstrukturen selbst mathematischen Funktionen entsprechen, d. h. eine Abbildung ihrer Eingabewerte in ihre Ausgabewerte vornehmen. Die Architekturfunktion ist somit eine zusammengesetzte Funktion aus den Übertragungsfunktionen der Steuerstrukturen und den Übertragungsfunktionen der Komponenten. Sie hängt davon ab, welche Komponenten mit welchen Steuerstrukturen in verschachtelter und sequenzierter Weise verknüpft sind.

Zunächst ist es sinnvoll, die Programmfunktionen der bekannten Steuerstrukturen Sequenz, Alternative und Iteration mit ihren minimal notwendigen Komponenten in SB-Sicht aufzuzeigen. Die Programmfunktion ist die Abbildung, die zwischen den Daten am Eintrittspunkt (E) und den Daten am Austrittspunkt (A) jeder Steuerstruktur definiert ist.

Für die Sequenz ist in (Abb. 4) die Programmfunktion [P] durch die Funktionskomposition der beiden Übertragungsfunktionen ***g*** und ***h*** der einzelnen Komponenten gegeben:

$$[P] = [\boldsymbol{g} \,;\, \boldsymbol{h}] = [\boldsymbol{h}] \circ [\boldsymbol{g}] \qquad (1)$$

Eckige Klammern bezeichnen Programmfunktionen und „o“ ist der Kompositionsoperator, der auch als „;“ (fettes Semikolon), mit Vertauschung der Operanden, geschrieben werden kann.

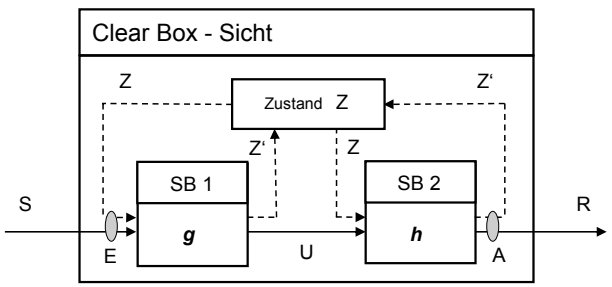


Abbildung 4: Zwei durch Sequenz verknüpfte Komponenten

Die Programmfunktion einer Alternative (Abb. 5) ist wie folgt definiert:

$$[P] = [\text{if } \boldsymbol{p} \text{ then } \boldsymbol{g} \text{ else } \boldsymbol{h} \text{ fi}] = ([\boldsymbol{p}] = \text{true} =: [\boldsymbol{g}] \vee [\boldsymbol{p}] = \text{false} =: [\boldsymbol{h}]) \quad (2)$$

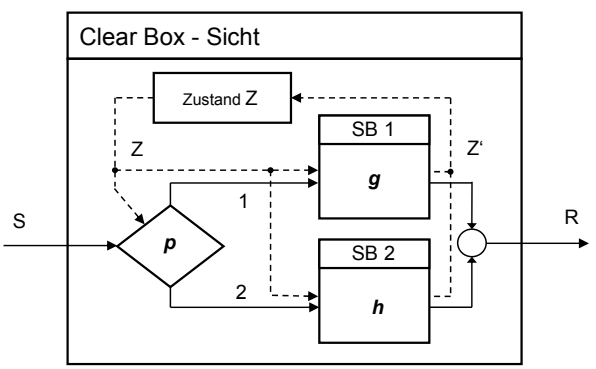


Abbildung 5: Zwei durch Alternative verknüpfte Komponenten

Für die Semi-Alternative gilt:

$$[P] = [\text{if } \boldsymbol{p} \text{ then } \boldsymbol{g} \text{ fi}] = ([\boldsymbol{p}] = \text{true} =: [\boldsymbol{g}] \vee [\boldsymbol{p}] = \text{false} =: I) \quad (3)$$

“I” ist die Identitätsfunktion. Die Programmfunktion der Alternative ist je nach Wahrheitswert des Prädikats ***p*** entweder die Programmfunktion der Komponente ***g*** oder die der Komponente ***h***. Das sind aber die tabellarisch gegebenen Übertragungsfunktionen der Komponenten. Entsprechendes gilt für die Semi-Alternative.

Die Programmfunktion der Iteration (Abb. 6) ist nur definierbar, wenn der Abbruch der Iteration gesichert ist. Unter dieser Voraussetzung lautet sie:

[P] = [while ***p*** do ***g*** od] = [if ***p*** then ***g*** **;** while ***p*** do ***g*** od fi] (4)

Wird die Schleife einmal durchlaufen, wird zunächst der Prädikatknoten mit true verlassen und ***g*** einmal ausgeführt und danach wird der Programmablauf wieder in die Schleife eintreten. Das ist aber funktionsäquivalent einer Semi-Alternative mit ***p*** im Prädikatknoten und ***g*** in Sequenz mit der Iterationsschleife im then-Zweig.

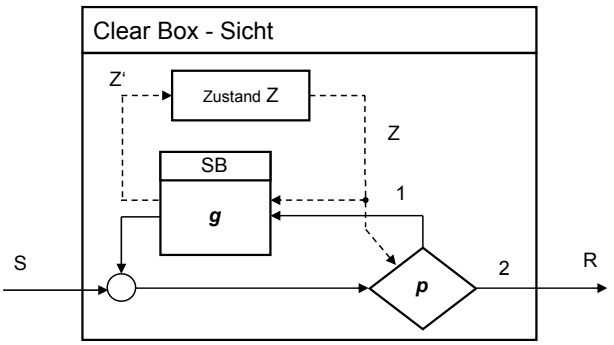


Abbildung 6: Eine Komponente in einer Iteration

Das bedeutet, die Programmfunktion einer Iteration ist nur rekursiv zu bestimmen. Sie ist, solange ***p*** true ist, durch wiederholte Funktionskomposition der Übertragungsfunktion ***g*** mit sich selbst zu ermitteln. Für ***p*** = false ist sie gleich der Identitätsfunktion I.

[P] = [if ***p*** then ***g*** **;** [P] fi] = ([***p***] = true =: [P] o [***g***] ∨ [***p***] = false =: I) (5)

Eine Architektur ist aus einer endlichen Anzahl von ineinander geschachtelten und sequenzierten Komponenten aufgebaut. Ihre Architekturfunktion kann extrahiert werden, indem mit den innen liegenden Steuerstrukturen begonnen wird und sukzessive nach außen hin die einzelnen Programmfunktionen der Steuerstrukturen ermittelt und zusammengesetzt werden. Diese Komposition der Steuerstrukturen ist, wie jede Komposition, hierarchisch geordnet. Da die Regeln dieser hierarchischen Ordnung definiert sind, ist der Extraktionsprozess prinzipiell algorithmisierbar unter den Vorrausetzungen, die bei der Iteration erfüllt sein müssen. Gleiches gilt für ein Programm in einer höheren Programmiersprache, auch hier ist die Programmfunktion extrahierbar. Statt der Übertragungsfunktionen der Komponenten gehen hier Anweisungen in die Steuerstrukturen ein.

Die bisherigen Betrachtungen zeigen, es besteht eine Äquivalenz zwischen einer mathematischen Funktion, sagen wir ***f*** und graphischen bzw. prozeduralen Regeln. Das kann in den folgenden Funktionsgleichungen formuliert werden:

f = ***g*** **;** ***h*** (6)
f = if ***p*** then ***g*** else ***h*** fi (7)
f = while ***p*** do ***g*** od (8)

Wenn ***p***, ***g***, ***h*** als gegebene Funktionen in der jeweiligen Regel betrachtet werden und ***f*** als gesuchte Funktion, drücken diese Gleichungen die Extraktions- bzw. Verifikationsaufgabe aus. Sie besteht darin, aus einer zusammengesetzten graphischen oder prozeduralen Regel (Architektur oder Programm) schrittweise für die jeweils innerste Regel die gesuchte Funktion ***f*** (Architektur- oder Programmfunktion) zu ermitteln und im Fall der Verifikation mit der Vorgabefunktion zu vergleichen.

Die entgegengesetzte Aufgabe ist die Entwurfs- bzw. Programmieraufgabe. Hier ist eine Spezifikationsfunktion ***f*** (als BB- oder SB-Spezifikation) gegeben und die Regel, wie diese Funktion graphisch oder prozedural durch ***p***, ***g***, ***h*** ausgedrückt werden kann, ist gesucht. Bekannterweise ist der erste Architekturentwurf nicht sofort geeignet, dass seine Architekturfunktion die Vorgabefunktion identisch erfüllt. Die Architektur muss schrittweise verbessert werden, bis Übereinstimmung mit der Vorgabefunktion erreicht ist. Dieses Vorgehen entspricht der iterativen Lösung der Entwurfsaufgabe, indem die Regel für den Entwurf in ihrer Zusammensetzung (Steuerkonstrukte) und in ihren dazugehörigen Konstituenten ***p***, ***g*** und ***h*** modifiziert werden. Die in iterativer Weise gefundene Architekturfunktion beschreibt vollständig und eindeutig das äußere Verhalten des Softwaresystems und erfüllt korrekt seine funktionale Spezifikation. Das Softwaresystem kann daher im Ganzen als Komponente in einem weiteren Softwaresystem genutzt werden.

Die theoretische Untersuchung des Lösungsraumes für die Gleichungen (6)-(8) wäre nützlich, um Anhaltspunkte für den Ausgangsentwurf und für den Einfluss der Konstituenten, d. h. ihre Empfindlichkeit auf die Architekturfunktion, zu erhalten.

5. Schlussbemerkungen

Die Architektur eines Softwaresystems besteht aus einer einzigen Hierarchieebene. Das ist die oberste oder nullte Ebene der Clear-Box-Sicht, in der die Struktur der Verbundkomponente durch die benutzten Steuerstrukturen festgelegt wird. Die Einbindung der Komponenten erfolgt durch Nachrichten an bestimmte Dienste der Komponenten, die selbst mit Rückgabeobjekten die Architektur bedienen. Die bekannte Ausnahme, wo keine Reaktion durch die Komponenten erfolgt, ist wenn bestimmte Datenobjekte in ihr direkt gesetzt werden. Die Komponenten einer Architektur sind ausschließlich Blattkomponenten, die aber empfehlenswerter Weise untereinander nicht durch Nachrichten verknüpft werden sollten, sondern nur mit der Architekturebene in vertikaler Richtung über Nachrichten kommunizieren. Andernfalls gehen die Vorzüge einer strengen Hierarchie verloren. Die Blattkomponenten selbst bilden somit keine eigene Ebene. Sie sind zunächst die elementaren Teilsysteme, die noch nicht weiter zerlegt wurden. Der Vorzug einer strengen Hierarchie besteht hier darin, dass jede Verfeinerung der Blattkomponenten arbeitsteilig durch unterschiedliche Entwickler simultan erfolgen kann, sobald eine korrekte Architektur vorliegt.

Das Bedürfnis nach Wiederverwendung und damit in Zusammenhang stehend, der Unempfindlichkeit gegen Änderungsauswirkungen fordert eine feiner granulierte Zerlegung der Blattkomponenten in weitere Komponenten, u. U. über mehrere

Hierarchieebenen verteilt. Dabei sollte nie, wie schon gesagt, das Prinzip der referenziellen Transparenz der Verfeinerung in streng hierarchischen Strukturen verletzt werden, da jede horizontale Verknüpfung von Komponenten die Komplexität des Softwaresystems erheblich steigert. Außerdem wird auch unerwartetes Verhalten bei der Komposition solcher horizontal und vertikal verknüpfter Komponenten verursacht. Damit eine Komponente zur Wiederverwendung geeignet ist, muss sie in den drei Sichten der Box-Struktur-Methode beschrieben sein. Diese Methode ist als Ansatz für eine komponentenorientierte Softwareentwicklung geeignet. Es ist für den Einsatz der Methode in der Praxis noch etliches zu erforschen, vor allem müssen die Sequenzbasierte Spezifikation, wie auch die Funktionsextraktion algorithmisiert werden. Nur der Einsatz des Rechners selbst kann nach diesem Ansatz zu einer Verbesserung der Qualität von komponentenorientierten Softwaresystemen führen.

Literaturverzeichnis

[SCHE84] Scheffel, H.; Schönefeld, R.: Objektorientierte Betriebssysteme – ihre Implementierung auf Kleinrechnern. Vortrag, Tagungsbericht INFO84, Dresden, 6.2.-10.2.1984.

[SCH86] Schönefeld, R.; Schrewe, I.: Objektorientierte Programmierung in C. edv-Aspekte, 5 (1986) 4, S. 34 – 37.

[SCH87] Schönefeld, R.: Datenabstraktion – ein wichtiges Prinzip der Softwaretechnik. Plenarvortrag, Kolloquium z. 25. Jahrestag des ORZ der TH Ilmenau, 11.9.1986 u. Wiss. Zeitschrift der TH Ilmenau 33 (1987) 5.

[SCH88] Schönefeld, R.: Objektorientierte Softwareentwicklung. Tagungsunterlagen, XXXIII. Intern. Wiss. Kolloquium der TH Ilmenau, Reihe B, 1988.

[SCH89] Schönefeld, R.: Warum ist C++ als Sprache für die Softwareentwicklung so bedeutsam? Wiss. Zeitschrift der TH Ilmenau 35 (1989) 2, S. 43 – 56.

[SCH90a] Schönefeld, R.: Eine objektorientierte Architektur für eine C++-Programmierumgebung (Teil 1 und 2). Wiss. Zeitschrift der TH Ilmenau, 1990, H1 und H2.

[SCH90b] Schönefeld, R.: Einfluß der objektorientierten Programmierung auf Softwaresysteme. edv-Aspekte, 9 (1990) 3, S. 57 – 60.

[KLO91] Klotz, St.; Schmidt, S.; Schönefeld, R.: Ein ECD-Dienst für die objektorientierte Programmentwicklung. Workshop “2. Dresdner Informatik-Tage“, Teil 1: Informatik und Umwelt, TU Dresden, Fak. Informatik, 4.2. – 6.2.1991.

[SCH91] Schönefeld, R.: Erfahrungen mit dem objektorientierten Paradigma,. Tagungsbericht XXXVI. Intern. Wiss. Kolloquium der TH Ilmenau, 21.10. – 24.10.1991.

[GAM95] Gamma, E. et.al: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, 1995.

[SCH07] Schönefeld, R.: Theoretische Aspekte der Softwaretechnik – Softwaresysteme und Softwarekomponenten. Gastvorlesungen an der TU Ilmenau, SS 2007, http://www.tu-ilmenau.de/fakia/Theoretische-Aspekte.6107.0.html

Die Entwicklung von Datenbanksystemen für die Verwaltung strukturierter Daten im Kombinat Robotron

Jürgen Bittner

SQL GmbH Dresden
01069 Dresden
juergen.bittner@sql-gmbh.de

Abstract: Der Beitrag behandelt die Entwicklung von Datenbanksystemen für die Verwaltung strukturierter Daten, die im Kombinat Robotron mit dessen Gründung 1969 ihre Anfänge fand und im Verlauf der folgenden 20 Jahre zur fachlich und wirtschaftlich bedeutsamsten Eigenentwicklung im Ostblock wurde. Die Hauptlinie bildete das Datenbankbetriebssystem/Robotron (DBS/R) mit mehr als 200 Anwendern, dem ab 1985 die Entwicklung des Systems INTERBAS folgte.

1 Einführung

Die Entwicklung von Datenbanksystemen zur Verwaltung strukturierter Daten im Kombinat Robotron wurde durch zwei Vorhaben initialisiert, die zunächst unabhängig voneinander konzeptionelle Quellen waren und andererseits umfangreiche Anwendungserfahrungen lieferten.

Als Ergebnis der Analyse der Anforderungen für das effiziente Speichern und Wiederauffinden von Daten der Fertigungsvorbereitung in Industriebetrieben wurde 1969 das System BOMP (Bill of Material Processor) der IBM gefunden und detailliert untersucht. Es bildete dann die Grundlage für die erfolgreiche Entwicklung des Systems **BASTEI** (Bankspeicherung Technischer Informationen), mit dem die Mehrheit der großen Fertigungsbetriebe der DDR ihre Stücklisten und Arbeitspläne verwalteten. Maßgebend hierfür war die Möglichkeit, komplexe Beziehungsstrukturen in großen Datenmengen zu realisieren und damit eine völlig neue Qualität und Performance der DV-Prozesse zu erreichen.

Etwa gleichzeitig wurde das System **SAWI** (Speichern und Automatisches Wiederauffinden) entwickelt, das eine Verwaltung fest formatierter Dateien mit einer Definitions- und Anfragesprache für einfach strukturierte Anwendungen ermöglichte und bereits eine komfortable Metadatenverwaltung besaß, die auch eine Darstellung externer Eingabedaten einschloss. Damit war ein bedeutender Produktivitätszuwachs für die Entwicklung einfacher bzw. begrenzt komplexer Datenmanipulations- und Anfrage-Programme erreichbar.

Bereits 1971 wurde die folgerichtige Vereinigung der Funktionalitäten beider Systeme vorbereitet und erweitert um neue Zielstellungen ab 1973 unter der Produktbezeichnung **Datenbankbetriebsystem/Robotron (DBS/R)** konsequent entwickelt.

Mit dieser Entwicklung wurde die begrenzte Datenstruktur des Systems BASTEI verallgemeinert und erstmals ein Datenmodell realisiert, das eine flexible Abbildung beliebiger Anwender-Datenstrukturen ermöglichte. Dazu wurden auch die Datenstrukturen der Systeme IMS, IDS und Total untersucht. Das DBS/R-Datenmodell ist ein Netzwerkmodell, dessen Abbildungsmächtigkeit mit der des CODASYL-Standards vergleichbar ist. Der CODASYL-Standard fand allerdings erst später Verbreitung und konnte für die DBS/R-Entwicklung nicht mehr berücksichtigt werden. DBS/R verfügte über eine DDL und DML, die ausgeprägt komfortabel und erweiterungsfähig war. Sie wurde als eigenständige Sprache, aber auch einbettungsfähig realisiert. Auf der Grundlage des DBS/R-Datenmodells, das theoretisch beliebig umfangreiche Strukturen ermöglichte, wurde auch die Vision einer Unternehmens-Datenbank verfolgt. Ein erster Schritt sollte die Anwendung von DBS/R als Basis der Sachgebietsorientierten Programmiersysteme (SOPS) sein, was nur in Ansätzen erreicht wurde.

Nach erfolgreicher Industrieerprobung 1978 wurde DBS/R im Verlauf von 10 Jahren bis zur Version 6 stark anwendungsgetrieben weiter entwickelt. Das ursprünglich typische Single User System erhielt mit Hilfe des vom Kooperationspartner LfA entwickelten Transaktionsmonitors ESY eine Multi User Version, und wurde später zu einem transaktions-basierten Datenbanksystem.

Auf der Grundlage der DBS/R-Erfahrungen, der parallel durchgeführten Forschungsarbeiten und einiger Erfahrungen aus der Entwicklung des SQL-Systems DABA 1600 begann dann 1986 die Entwicklung des DBMS **INTERBAS** gemeinsam mit dem sowjetischen Softwarehaus Zentrprogrammsistem in Kalinin (heute Twer). Die umfangreiche Zielstellung dieses Vorhabens wurde bis Anfang 1990 mit großer Anstrengung verfolgt. Dazu gehörten solche Merkmale wie Mehr-Ebenen-Architektur, koexistierende Datenmodelle und darauf aufbauende modulare Baustein-Struktur des Systems bei einem völlig neuen Anforderungsmaß hinsichtlich der Plattform-Portabilität.

Die drei Entwicklungsetappen BASTEI und SAWI ab 1969, DBS/R ab 1973 und INTERBAS ab 1985 bilden die Gliederung für diesen Beitrag. Während in der ersten Etappe teilweise bearbeitete Programmcodes von Vorbildsystemen einbezogen wurden, beruhten die hier beschriebenen Folgeetappen vollständig auf Eigenentwicklung.

2 Das Zeitalter der Datenbanken beginnt

Mit der angekündigten Herstellung eigener Direktzugriffs-Speicher im ESER und der vorläufigen Bereitstellung der entsprechenden IBM-Typen (7,25 MB) ergaben sich völlig neue Möglichkeiten für DV-Prozesse, die bisher ohne umfangreiche Sortier-, Misch- und Suchvorgänge mit sequentiellen Dateien nicht möglich waren.

Aber der einfache Direktzugriff mit Plattenadresse oder auch mit Schlüssel über einen Index genügten noch nicht um die aufwandsintensive Verwaltung von Stücklisten und Arbeitsplänen in den Unternehmen der Fertigungsindustrie sowie die notwendigen Bedarfs- und Planungsrechnungen ausreichend zu beschleunigen. Dazu bedurfte es spezieller Konzepte zur gespeicherten Darstellung von Beziehungen, wie sie im System BOMP (Bill Of Materials Processor) der IBM gefunden wurden. Grundlage war die Verwendung von physischen Adressen zur Realisierung von Verweisen und zur Darstellung von Datensatz-Mengen durch Verkettung.

Es war ein richtungweisender Vorschlag von J. ADAM und R. HEINEMANN dieses System, das heute weltweit als erstes kommerzielles Datenbanksystem anerkannt ist, als Grundlage für die Entwicklung des Robotron-Systems BASTEI (Bankspeicherung Technischer Informationen) [BAS72] zu benutzen. Die damals Beteiligten wurden sich der technologischen Bedeutung dieses Qualitätssprungs für die künftige DV sehr schnell bewusst. Davon zeugen die ausgiebigen Bemühungen um eine Definition zum Begriff Datenbank, die schließlich das redundanzfreie Speichern der Daten einschließlich der gespeicherten Beziehungen und den Direktzugriff betonte.

2.1 BASTEI – Start mit großem Erfolg

Die Entwicklung von BASTEI umfasste ein Neuschreiben des Programmcodes, die Anpassung an die verfügbaren ESER-Betriebssysteme sowie eine Reihe funktionaler Erweiterungen zur Verbesserung der Dateneingabe sowie der Auswertbarkeit. Die Prinzip-Struktur wurde vom Vorbild übernommen. Im Programmcode war die Verwaltung von maximal vier Dateien für jeweils einen fest formatierten Satztyp einschließlich der Beziehungsstrukturen vorgesehen (Abb. 1).

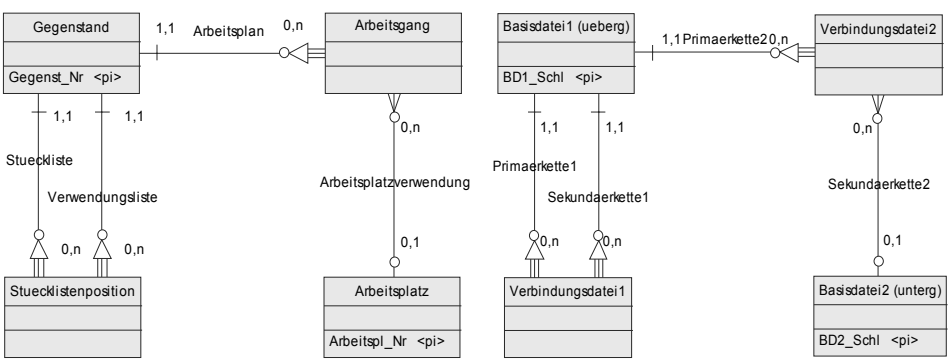


Abbildung 1: ER-Diagramm von BASTEI (ursprünglich und verallgemeinert)

Neben der programmtechnischen Realisierung und reinen Übersetzung der anwendungsbezogenen Terminologie ins Deutsche entstand auch eine verallgemeinerte, konzepte-bezogene Terminologie, die dem Verständnis des gegebenen Strukturtemplates für andere Anwendungsfelder dienen sollte. So wurden die Begriffe Basisdatei (übergeordnete, untergeordnete), Verbindungsdatei und weitere geprägt.

Das Strukturtemplate des jeweiligen Satztyps enthielt bereits die Adressfelder (Verweisadressen, Kettadressen), die gemäß der strukturellen Bedeutung des jeweiligen Satztyps erforderlich waren. Eine Metadatenverwaltung und Datenbeschreibungssprache existierte noch nicht. Stattdessen gab es eine parameter-orientierte Anpassung des Assembler-Quellcodes. Diese enthielt die Spezifizierung der Datenbanksätze einschließlich bestimmter Pflichtfelder, der Annahme oder Einschränkung der jeweiligen Beziehungsstrukturen mit einer Reihe spezifischer Merkmale. Weiterhin war alles zu spezifizieren, was zur Generierung eines Programms zur Datenverwaltung erforderlich war, also z.B. auch Eingabedaten, Kontrolllisten und Fehlerprotokolle.

Die Auswertung der Daten erfolgte mit Hilfe von Assemblermakros unterschiedlicher Komplexität für den satzweisen Zugriff mit Schlüssel oder Adresse, das Navigieren durch Kettenverfolgung, sowie auch rekursive Verfahren zur Auflösung von Strukturen.

Die beabsichtigte Zielstellung der BASTEI-Entwicklung wurde mit großem wirtschaftlichem Erfolg erreicht, indem mehr als 100 Unternehmen in der DDR und im östlichen Ausland ihre Fertigungsvorbereitung mit BASTEI realisierten. Auch erste Anwendungen im Sinne der angestrebten Verallgemeinerung kamen bereits zu Stande [BIT74] [RAS80]. Die Projekte hatten eine Lebensdauer, die teilweise bis in die 80er Jahre reichte.

2.2 SAWI – Wichtige Erfahrung für später

Die besondere Bedarfssituation, die zur Entstehung von BOMP und BASTEI geführt hatte, bedeutete in diesen Jahren noch nicht, dass ein allgemeiner Trend zu beziehungs-orientierten Datenverwaltungssystemen, also Datenbanken eingesetzt hätte. Im Gegenteil, überall wo der Nutzen nicht sofort offensichtlich war, wurde wegen der befürchteten Komplizierung, z.B. in Verbindung mit der Handhabung von Adressketten auf einfachere Werkzeuge orientiert.

Auf diesen Bedarf war das System **SAWI** (Speichern und Automatisches Wiederauffinden) gerichtet, dessen Entwicklung I. BLUDAU leitete. Festformatierte Dateien ohne Berücksichtigung irgendwelcher Beziehungen konnten mit einer leicht handhabbaren Sprache definiert werden, ebenso Eingabedateien, die der Pflege der Bestandsdateien dienten. Diese Datenbeschreibungssprache ermöglichte eine Metadatenverwaltung, die die Voraussetzung für Sprachanweisungen zum Laden und Ändern der Daten bildete. Anweisungen zum Lesen, Suchen und weiteren Manipulationen bewirkten, dass SAWI zu einem effizienten Entwicklungswerkzeug für einfache bis durchschnittliche DV-Anwendungen wurde, die dem Anwender für diese Aufgaben das Benutzen des Assemblers oder anderer Programmiersprachen ersparte.

Auch SAWI erreichte eine Anzahl mehrjähriger Anwender, die die Wirtschaftlichkeit der Entwicklung wahrscheinlich sicherstellte. Sie blieb aber deutlich unter der von BASTEI, was vielleicht damit zusammenhängt, dass die gemeinsame Nachfolgeentwicklung schon sehr frühzeitig in Aussicht gestellt wurde. Die Entwicklung einer Metadatenverwaltung und einer komfortablen DDL und DML gehören zu den Erfahrungen, die für die weitere Entwicklung sehr wichtig waren.

3 DBS/R – 15 Jahre Entwicklung und Anwendung

Beide Entwicklungslinien ließen für eine längerfristige Sicht schnell ihre Grenzen erkennen, die einerseits die Vereinigung in einem System nahe legten, andererseits aber auch wesentliche Erweiterungen der vorhandenen Konzepte erforderten. Es war z.B. klar, dass die Dateienanzahl einer Datenbank nicht spürbar begrenzt bleiben durfte und dazu die Unabhängigkeit des Programmcodes von der Datenbeschreibung auf der Grundlage einer Metadatenverwaltung unerlässlich war. Wenn schon die Struktur der Datenbank für die Zwecke der Produktionsplanung zusätzliche Objekt- und Beziehungstypen wie Kunden, Kundenaufträge, Fertigungsaufträge, Lagerbestände erforderte, dann sollte die Perspektive einer unternehmensweiten Datenintegration möglichst nur durch Hardware-Ressourcen begrenzt sein.

Außerdem gab es spezielle Erfahrungen, die eine stärkere Differenzierung der bisherigen Strukturformen erforderten, um die in der Realität vorhandenen Beziehungsstrukturen adäquat abbilden zu können und insgesamt Performanceverbesserungen zu erreichen. Es ging um die Vision, dass künftige Software weitgehend auf der Grundlage von Datenbanken arbeitet, was zunächst noch jahrelang sehr umstritten war.

Die Initiative zur DBS/R-Entwicklung ging wiederum von R. HEINEMANN aus. Er beauftragte Ende 1971 J.BITTNER mit ersten konzeptionellen Überlegungen für ein neues Datenbankbetriebssystem [BIT73] und später mit der Leitung der Entwicklung. Das System wurde für die Rechenanlagen des ESER geplant und war auch für IBM/360/370 anwendbar. Es begann ein mehrjähriger Entwicklungsprozess, der ab 1975 auch die Kooperation mit der Firma Leitstelle für Anwendungsforschung Berlin einschloss. Diese Kooperation führte zur Entwicklung des Exekutivsystems ESY/R unter Leitung von V.LANGE, das als online-System und erster Transaktionsmanager für DBS/R diente. Alle konzeptionellen Überlegungen erforderten die ausgiebige Überprüfung in der Praxis, was durch die planmäßige Einordnung der Industrieerprobung in die Entwicklungspläne einer jeden Ausgabe erreicht wurde. Besonders die Erprobung der ersten Entwicklungsstufe führte zu umfangreichen konzeptionellen Präzisierungen, die vor allem der Performance-Verbesserung dienten. DBS/R wurde bis 1988 entwickelt und kam bei mehr als 200 Anwendern zum Einsatz.

3.1 Datenmodell und Performance

Ausgehend von den bisher gesammelten Erfahrungen war zu klären, welche Daten- und Beziehungsstrukturen unterstützt werden sollten. War das zunächst die Frage der guten Abbildbarkeit komplexer Sachverhalte, so musste im Weiteren das Zeitverhalten der jeweiligen Strukturform, aber auch die Konsequenzen hinsichtlich des benötigten Plattenspeicherplatzes berücksichtigt werden. Die wichtigsten Entscheidungen dazu wurden nicht ohne Auswertung der verfügbaren Informationen über einige Systeme getroffen, die bereits vor und nach 1970 entstanden und noch in Entwicklung waren. Das waren das von C.BACHMAN bei Honeywell geprägte Integrated Data Store (IDS), das bei Cincom Systems entwickelte TOTAL und das bei IBM neben BOMP ab 1969 entwickelte IMS.

Während im IDS bereits das Master-Detail-Modell existierte, das später ein grundlegendes CODASYL-Konzept wurde, schien dessen Realisierung vorrangig mit Hilfe von Pointer Arrays oder Master-Detail Recordsequenzen kaum die Effizienz der uns vertrauten Adresskettentechnik zu erreichen. Pointer Arrays bewirkten damals variabel lange Datensätze, die mit der damaligen Plattentechnik schwer zu behandeln waren. Recordsequenzen sind änderungsunfreundlich und erschweren die Suche aus anderen Kontexten, für die die Sequenz nicht gedacht war. Ähnliche Skepsis bewirkte auch die in dieser Hinsicht konsequente hierarchische Datenstruktur des IMS. Dagegen schien die Möglichkeit der Verkettung aller Satztypen miteinander, wie es bei TOTAL der Fall war, viel mehr „datenbanktypisch“ zu sein, weil hiermit ein freizügiges Navigieren, Ändern und die effiziente Bereitstellung relevanter Daten erreicht wurde. Es war auch bekannt, dass IBM die BOMP-Technologie mit MRP und PICS fortsetzte.

Die schon bewährte Verkettungstechnik mit Hilfe von Adresspointern wurde deshalb weiter verallgemeinert und erweitert [BIT73][DBS76][KB77][AB78], sodass ein leistungsfähiges Netzwerkdatenmodell entstand. Charakteristisch dafür ist das Zusammenwirken von Basis- und Verbindungsdateien. Jeder Satz einer Basisdatei ist unabhängig und besitzt einen eindeutigen Schlüssel. Er enthält nur solche Daten, die nur in Verbindung mit dem Schlüssel, aber nicht in Verbindung mit den Schlüsseln anderer Dateien vorkommen. Jeder Satz einer Verbindungsdatei stellt eine Beziehung zu einer festen Anzahl von Sätzen anderer Basis- oder Verbindungsdateien her, die auch die Identifizierung der Verbindungssätze übernehmen können. Er enthält die Daten, die nur in Verbindung mit den Schlüsseln der verbundenen Dateien vorkommen und ist von einem Satz mindestens einer Datei abhängig. Abb.2 zeigt zwei Strukturbeispiele.

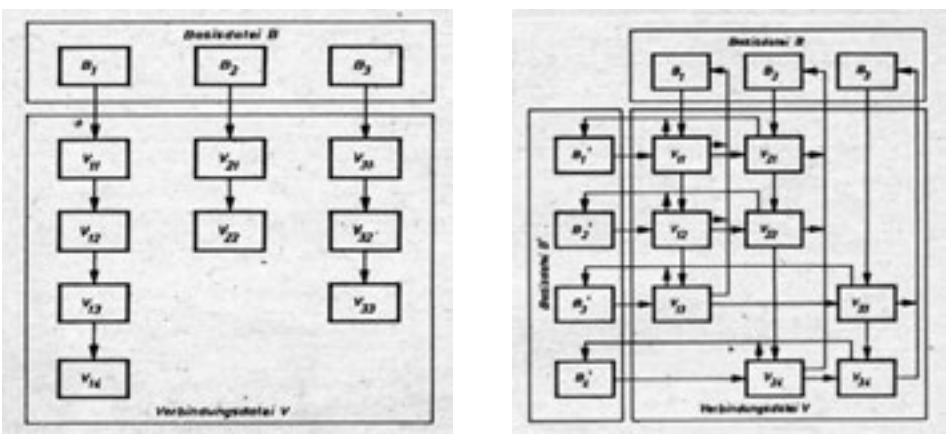


Abbildung 2: Verkettungs-Strukturen mit Basis- und Verbindungsdateien [KB77]

Der Dateibegriff steht bei DBS/R immer noch auch für Satztyp. Das war damals ein Vorteil, da bei begrenzter Laufwerksanzahl Datenbankteile leicht demontiert werden konnten.

Die Verallgemeinerung (Abb.3) erlaubte nun eine beliebige Anzahl von Dateien, wobei die Anzahl je Verbindungsdatei nicht mehr auf maximal 2 begrenzt war, unter denen jetzt auch Verbindungsdateien erlaubt waren. Eine Reihe von Spezifizierungs-Details war erforderlich, um eine optimale Performance für das Navigieren und Ändern zu erreichen. Weiterhin wurde auch die Notwendigkeit der jeweiligen Beziehung im Sinne der Referenzintegrität festgelegt. Für den Spezialfall, der 1:1 oder 0:1-Beziehung zwischen zwei Basisdateien wurde die „Kopplung“ eingeführt, ein- oder wechselseitiger Adressverweis. Neben dem notwendigen Primärindex für jede Basisdatei konnten die für eine Suche relevanten Felder oder Gruppen von Feldern einen eindeutigen oder mehrdeutigen Sekundärindex erhalten.

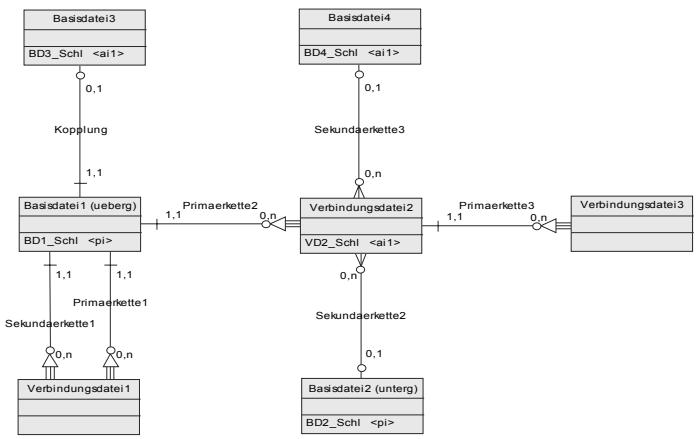


Abbildung 3: Möglichkeiten der Struktur von DBS/R-Datenbanken

Die vielfältigen Möglichkeiten der Darstellung auch komplexer Sachverhalte wurden umfassend genutzt (Abb.4). Schon in den ersten Jahren nach Beginn der kommerziellen Verfügbarkeit wurde DBS/R bei mehr als 100 Unternehmen in allen Wirtschaftszweigen und Verwaltungseinrichtungen angewendet. Diese Anzahl war weniger durch die Anwendbarkeit, sondern durch die verfügbaren Rechner begrenzt.

Rückblickend stellt sich die Frage, ob der komplizierte Weg der Pointer-basierten Strukturverwaltung nicht vermeidbar gewesen wäre, vor allem mit Hilfe einer stärker Index-orientierten Strukturrealisierung, was mit Sicherheit den Entwicklungsaufwand reduziert hätte. Dieser Weg war auch zu Beginn der DBS/R-Entwicklung ausgiebig diskutiert worden, konnte aber nicht benutzt werden, weil der Performanceverlust bei der Selektion größerer bestimmter Datenmengen und bei Strukturverfolgungsprozessen einfach zu groß war. Diese Einschätzung hat sich über viele Jahre hinweg bestätigt, am stärksten durch die Tatsache, dass die relationalen DBMS in den ersten 10 Jahren ihrer Existenz erhebliche Performance-Probleme hatten und besonders eine Stücklisten-verwaltung mit RDBMS erst seit etwa 10 Jahren praktikabel ist. Jedoch waren die Performance-Wirkungen einiger Eigenschaften von Verbindungsdateien auch für die DBS/R-Anwender nicht immer überschaubar.

Interessanter ist die Frage, welche Position DBS/R gegenüber dem CODASYL-Standard einnahm. Zunächst ist festzustellen, dass die Entwicklung bei CODASYL zwar bereits etwa 1969 begann, sich aber erst Ende der 70er Jahre festigte, so dass für DBS/R ein später Kurswechsel in Richtung des Standards nach mehreren Entwicklungsjahren erheblichen Mehraufwand gebracht hätte, außerdem Kompatibilitätsprobleme bei den Anwendern, die in erster Linie Performanceverbesserungen forderten. Der Vergleich beider Netzwerkmodelle zeigte, dass für jedes CODASYL-Konzept eine Entsprechung in DBS/R vorhanden ist. Bemerkenswert war auch die Tatsache, dass die später erfolgreichsten CODASYL-Systeme, IDMS von Cullinane Corp. und UDS von Siemens als SET MODUS auch CHAIN benutzten. Sieht man ab vom unbestrittenen Wert der Standardisierung, so bliebe die Beziehungssicht beider Seiten zu vergleichen und zu bewerten. CODASYL löst alle Beziehungen in einfache, als SET bezeichnete OWNER-MEMBER Konstrukte auf und liefert damit eine gut überschaubare logische Sicht, als Grundlage einer 2-Ebenen-Architektur. Allerdings ist die physische Realisierung der SETs zwangsläufig nicht unabhängig voneinander. DBS/R realisiert die Beziehung eher gemäß ihrer realen Komplexität als Verbindungsdatei, die man aus diesem Blickwinkel als Realisierung eines „N-dimensionalen Set“ verstehen kann. Für das DBS/R-Datenmodell kann das ursprüngliche Entity-Relationship-Modell von CHEN als am meisten adäquat gelten.

3.2 DBS/R-Anwendungsentwicklung und Betriebsarten

Seinem Entwicklungszeitraum entsprechend war die DBS/R-Anwendung zunächst vom Stapelbetrieb geprägt. Dabei kam es in erster Linie darauf an, wie die Vorteile der Datenbank mit Hilfe einer leistungsfähigen Programmierung dem Anwender verfügbar gemacht werden konnten. Dazu wurde eine leicht erlernbare Datenmanipulationssprache entwickelt [DBS77], die hauptsächlich von P. KRAH und W. PAUKSCH entworfen wurde.

Die DBS/R-Sprache umfasst Anweisungen für den Zugriff zu einzelnen Sätzen und Satzmengen, zum Navigieren und zur Ausführung komplexer Strukturauflösungs-verfahren. Weiterhin unterstützt die Sprache die Ein- und Ausgabe sequentieller Dateien, die Gestaltung von Listen, das Vergleichen sowie arithmetische Operationen. Besonders komfortabel war die Verarbeitung von Eingabedateien für das Laden und Ändern der Datenbank. Die Sprache folgte den Regeln der strukturierten Programmierung und ermöglichte die Entwicklung vollständiger Anwendungsprogramme, die schnell fertig zu stellen waren und nur geringen Testaufwand erforderten. Die Anweisungsfolge wird primär über die Belegung von Austrittspunkten mit den Namen der Folge-Anweisungen gesteuert. Die Sprache, auch als Englisch-Version verfügbar, wurde zu einem wichtigen Produktivitätsfaktor der DBS/R-Anwendung, zumal ihre Einfachheit auch ad hoc-Anfragen zuließ.

Alternativ zur eigenständigen Programmierweise wurde die Einbettung der DBS/R-Sprache im CODASYL-Stil in Assembler, PL/1 und COBOL realisiert. Möglicherweise gab es in der Mitte der 70er Jahre keine Datenbanksprache, die dem Anwender ein höheres Unterstützungsniveau bieten konnte.

```
LEITANW     WIEDERAUFFINDEN UEBERSETZEN UND
            AUSFUEHREN;
            SUCHEN IN DBPERS MIT SPRACHEN=
            RUSSISCH UND SPRKENNT=DOLMETSCH
            FUER SUCHAUS=AUSG;
AUSG        AUSGEBEN;
            KOPF 'MITARBEITER MIT DOLMET-
            SCHERKENNTNISSEN IN RUSSISCH' 20;
            VORSCHUB 3;
            KOPF;        LEERZEILE
            ZEILE VON DBPERS  PERSNR 2 NAME 12 VORNAME 24;
```

Abbildung 4: Beispiel eines einfachen DBS/R-Anwendungsprogramms [KB77]

Der Mehrbenutzer- und online-Betrieb des DBS/R wurde zunächst mit Hilfe eines zusätzlichen Systems, des Exekutivsystems ESY/R ermöglicht. ESY/R realisierte einen nachrichtenbasierten Teilhaberbetrieb für maximal 250 Datenstationen, in dem DBS/R in Form von mehreren Unteraufgaben laufen konnte. Das entsprach der auch bei anderen DBMS-Herstellern in den 70er Jahren üblichen Arbeitsweise wie z.B. IMS/DC für DL/1 der IBM.

Der wichtigste Nachrichtentyp diente dem Rufen von Transaktionen in Verbindung mit den dazu erforderlichen Parametern. Für jede Transaktion musste im ESY/R ein Anwendungsprogramm und dessen Beschreibung vorliegen, die u. a. Sperranforderungen der Transaktion enthält und die maximale Laufzeit. Drei Zugriffstypen bzw. Sperrniveaus wurden unterschieden: exklusiver Dateizugriff, geteilter Dateizugriff für Lesen, Aktualisieren von Datensätzen. Damit konnte ESY/R eine für den Parallelbetrieb erforderliche einfache Sperrverwaltung organisieren. Alle DB-Zugriffe wurden von ESY/R kontrolliert, so dass Rollback und Recovery möglich wurden.

Nachteilig war, dass die auf Parameter beschränkte Eingabe keine Dialoge innerhalb von Transaktionen zuließ. Bei IBM wurde aus diesem Grund CICS entwickelt, das später bei Robotron unter dem Namen DAKS zur Verfügung stand. Das vollständige Erschließen der diesbezüglichen Vorteile für DBS/R hätte jedoch ein spezielles auf die CICS-Architektur bezogenes Redesign mit hohem Aufwand erfordert.

Ab 1980 zeigten die Forschungen von T. HÄRDER und A. REUTER [HR80] [REU81], dass ein anspruchsvoller, performanter und integritätssichernder Mehrbenutzer-Betrieb durch Integration des Transaktionssystems in das DBMS wesentlich besser möglich ist. Dazu wurde zwischen 1984 und 1987 ein umfassendes Redesign realisiert. Ausgabe 6 brachte einen Transaktionsmanager, der Parallelität einer DDL-Aufgabe mit beliebigen Compile- und Leseaufgaben zuließ, aber Änderungs-Aufgaben je Datenbank serialisierte. Dieser Kompromiss war dem Umstand geschuldet, dass die Entwicklung der nächsten Produktgeneration bereits begonnen hatte und der mehr als 10 Jahre andauernde Anwenderdruck auf immer neuem Niveau die Laufzeit der Einzelaufgaben priorisierte. Bemerkenswert ist, dass mit Ausgabe 6 die Gesamtheit der Recovery-Arten R1–R5 vermutlich erstmalig kommerziell realisiert wurde.

3 INTERBAS – Herausforderung mit offenem Ende

Umfangreiche Erfahrungen aus der DBS/R-Anwendung, Forschungsergebnisse und erste relationale DBMS lieferten dringende Anstöße für eine Neuentwicklung, die aber auch prinzipielle Konflikte der damaligen Datenbanktechnologie zu behandeln hatte. Die eigene Konzeption wurde wegen ihres Umfangs dem Softwarehaus Zentrprogramm-sistem in Twer zur gemeinsamen Realisierung vorgeschlagen. Ende 1986 begann die Entwicklung mit der Bezeichnung INTERBAS, geleitet von J.BITTNER und W.S.RIPINIZER, die sich auf ein Führungsteam von etwa 15 Mitarbeitern beider Seiten stützen konnten. Bis Ende 1989 wurde ein für mehrere Betriebsysteme lauffähiges Teilsystem der Produkte in Abb. 5 entwickelt.

Schon die Betrachtung der Architektur, heute als Client-Server-Architektur zu bezeichnen, lässt Entscheidungen erkennen, die aus Problemen der damaligen Zeit resultieren, die aber teilweise vielleicht noch heute diskussionswürdig sind.

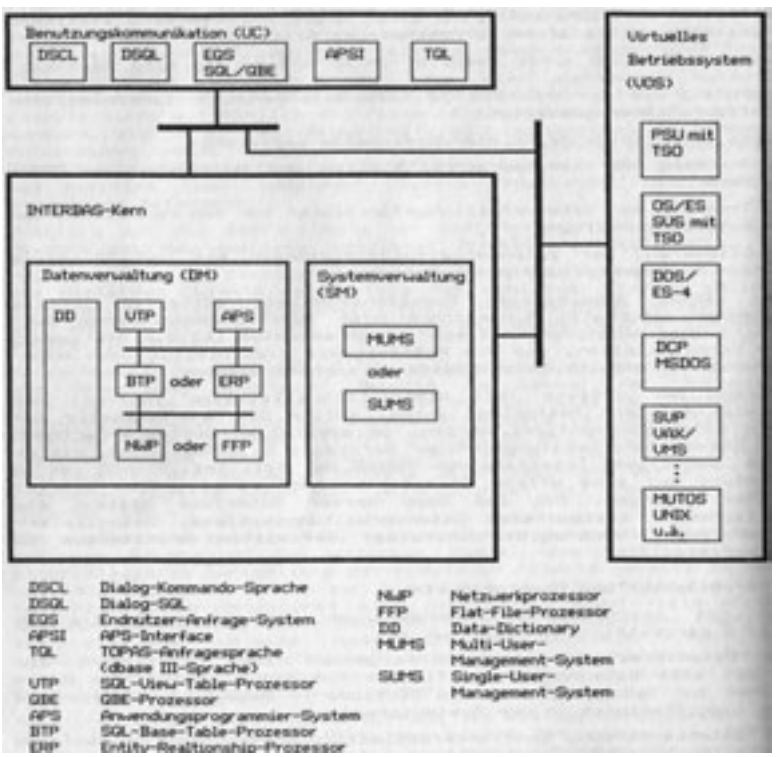


Abbildung 4: Architektur und Hauptprodukte von INTERBAS-1[BIT91]

Ein besonderer Aspekt war der Konflikt zwischen der absehbaren Zukunft der Sprache SQL, der Zielstellung objektorientierter Datenbanken und der Realität einer performance-kritischen Praxis. Über 50% der DBS/R-Anwendungen waren Fertigungs-Datenbanken mit Stücklisten, die mit dem damaligen Stand der RDBMS, vor allem ihrer verflachten Speicherungsstruktur kaum ablösbar waren, was bis weit in die 90er so blieb.

Eine mehrfache Datenhaltung derselben Daten in verschiedenen DBMS (SQL und Netzwerk-DB) sollte vermieden werden. Deshalb war eine Koexistenz der Datenmodelle zu realisieren, die einerseits ein performance-orientiertes Navigieren in einer Netzwerk-DB zulässt (CODASYL) und auf den höheren Ebenen die Anwendung von SQL für Basistabellen und Views ermöglicht. Neben diesem Vorteil war zu erwarten, dass die Abbildung von SQL-Operationen auf Zugriffspfadstrukturen des CODASYL-Netzwerks Performanceverbesserungen für SQL bewirkt, z.B. durch eine Join-Methode „set implemented join". Der Bedarf zur Verbesserung der Join-Performance besteht bis heute und hat z.B. zu einem Join Cluster oder Join Index geführt.

Die scheinbar naheliegende Zukunft der Objektorientierung bzw. der Nicht-Standard-DB sollte in der INTERBAS-1-Architektur mindestens eine Voraussetzung finden, indem die Basis für die interne Realisierung in Form von Netzwerk-Datenmodell-Stukturen bereitgestellt wurde und die Anforderungen einer Kern-Architektur gemäß [HR85] zu berücksichtigen waren. Zur Entscheidung für irgendein objektorientiertes Datenmodell fehlte allerdings ein hinreichender Motivationsdruck.

Hohe Priorität hatte die Zielstellung einer Verbesserung der Datenunabhängigkeit. Änderte man bei DBS/R die Speicherungsstruktur z.B. aus Performancegründen, so ergaben sich häufig Auswirkungen auf die Anwendungsprogramme. Deshalb wurde die konsequente Anwendung der 3-Ebenen-Architektur nach ANSI/X3/SPARC festgelegt, was sich auch innerhalb des SQL-Systems positiv auswirken sollte. Wird ein normalisiertes Tabellen-Schema als Ausprägung eines Konzeptuellen Schemas verwendet, so könnte man, eine geeignete Abbildungssprache voraussetzend, die interne Realisierung der Tabellen z.B. vertikal partitionieren, verteilen, denormalisieren ohne Auswirkungen auf die Verwendung der Tabellen in Queries, View-Definitionen, Triggern und gespeicherten Prozeduren.

Die 3-Ebenen-Architektur sollte außerdem die methodische Grundlage für den Datenbank-Entwurf bilden. Die Dringlichkeit eines kompatiblen Werkzeugs war offensichtlich und innovative Überlegungen zur Vorgehensweise [BIT84] waren in Arbeit. Hierbei sollte eine Schnittstelle für Anwendungsprogrammierer (APS) das Beschreiben von Benutzersichten (externe Schemas) gestatten, deren interaktive Integration den Entwurf eines Konzeptuellen Schemas unterstützt. Mit Hilfe von Kardinalitäten und einer Benutzungsstatistik sollten Voraussetzungen für den Entwurf Interner Schemas geschaffen werden

Die Architektur mit ihren Datenmodell-Optionen und -Alternativen der einzelnen Ebenen bildete gleichzeitig einen Rahmen für die Produktgliederung des Daten-verwaltungssystems. Während damals die bekannten Hersteller große monolithische Produkte auf den Markt brachten, neben denen einige kleinere Lösungen anderer Hersteller existierten, sollte INTERBAS eine einheitliche Produktfamilie werden, aus der gemäß des jeweiligen Bedarfs, oft auch durch Hardwareverfügbarkeit begrenzt, die geeignete Installation konfiguriert werden konnte. So war im einfachsten Fall ein Single-User-System(SUMS) mit dem Flat-File-Processor(FFP) oder Netzwerk-Processor der internen Ebene konfigurierbar.

Zeigt schon die angedeutete makroskopische Sicht auf INTERBAS mindestens für seine Zeit den hohen Anspruch des Projekts, so darf nicht unerwähnt bleiben, dass auch auf der Ebene der Systemkomponenten sehr gute Ergebnisse vorlagen. Ein Beispiel ist das EA-System (M.LENK), das den damaligen Stand der Forschung realisierte und u. a. eine Pufferverwaltung mit Merkmalen enthielt, die führende kommerzielle RDBMS erst seit wenigen Jahren anbieten.

Als die Arbeit an INTERBAS eingestellt wurde, gab es natürlich noch offene Fragen. Wie die Entwicklung weiter verlaufen wäre, ist nur spekulativ zu beantworten, Realität ist aber, dass viele der Beteiligten wichtige Erfahrungen für ihre spätere Tätigkeit mitgenommen haben.

Danksagung

Anlässlich dieses kurzen, nicht gerade systematischen Einblicks in einen 20jährigen Entwicklungsverlauf möchte ich mich bei allen Kollegen und Freunden bedanken, mit denen ich in dieser Zeit zusammen arbeiten und von denen ich lernen konnte. Danken möchte ich auch denen, die nach Einstellung der Entwicklung unserer Arbeit ihre Aufmerksamkeit und Anerkennung zuwendeten.

Literaturverzeichnis

[AB78] Adam, J.; Bittner, J; Grunert, B.; Heinemann, R.; Krah, P.: DBS/R Datenbankbetriebssystem/Robotron. Schriftenreihe Informationsverarbeitung, Verlag Die Wirtschaft, Berlin 1978.

[BAS72] Bittner, J.; Heinemann, R.; Krah, P. u.a.: BASTEI – Beschreibung eines Datenbankbetriebssystems. VEB Kombinat Robotron, ZFT, 1972.

[BIT73] Bittner, J.: Konzeptionelle Übersicht über die Leistungsfähigkeit und Anwendungsmöglichkeiten von DBS/R. VEB Kombinat Robotron, 1973.

[BIT74] Bittner, J.: BASTEI-Anwendung für die Planung, Lenkung und Abrechnung der Produktion in der chemischen Industrie. Rechentechnik Datenverarbeitung, Verlag Die Wirtschaft, Berlin 1974; Heft 3, S. 19-24, Heft 4, S. 17-20.

[BIT84] Bittner, J.: Design of the Conceptual Scheme on the Base of External Schemes. Proc. Of the Seventh International Seminar on Data Base Management Systems, Varna, 1984

[BIT91] Bittner, J.: Die Architekturkonzeption eines DBMS aus pragmatischer Sicht. Proc. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft, Kaiserslautern 1991, IFB 270, Springer Verlag, S. 1-15.

[DBS76] DBS/R – Anwendungsbeschreibung. VEB Kombinat Robotron-ZFT 1976.

[DBS77] DBS/R – Sprachbeschreibung. VEB Kombinat Robotron-ZFT 1977.

[HR80] Härder, T.; Reuter, A.: Abhängigkeiten von Systemkomponenten in Datenbanksystemen. Informatik-Fachberichte, 33, Springer 1980, S. 243-257.

[HR85] Härder, T.; Reuter, A.: Architektur von Datenbanksystemen für Non-Standard-Anwendungen. Proc. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft, Karlsruhe 1985, IFB 94, Springer Verlag

[KB77] Krah, P.: Bittner, J.: Speicherstruktur und Sprachkonzept des DBS/R. rechentechnik datenverarbeitung, Verlag Die Wirtschaft Berlin 1977; Beiheft 4, S.16-29.

[RA80] Rauchhaus, K.-H.: BASTEI in prozeduralen Auskunftssystemen. Rechentechnik Datenverarbeitung, Verlag Die Wirtschaft, Berlin 1980; Heft 2, S. 16.

[REU81] Reuter, A.: Fehlerbehandlung in Datenbanksystemen. Carl Hanser, München 1981.

Technologietransfer am Beispiel von DABA1600

Birgit Demuth Karl-Heinz Wiggert

Abstract: DABA 1600 war die DDR-Eigenentwicklung eines relationalen Datenbankmanagementsystems (DBMS) für PDP-11-kompatible Systeme wie dem Kleinrechner robotron K1630. Es wurde in den achtziger Jahren in Kooperation vom Zentrum für Forschung und Technik (ZFT) des Kombinates Robotron, später dem Betrieb Robotron-Projekt Dresden (RPD) und der Sektion Informationsverarbeitung der Technischen Universität Dresden entwickelt. Ab 1988 wurde es verkauft und fand bis zur Wende vorwiegend im Inland etwa 100 Anwendungen. Aus heutiger Sicht ist die Entwicklung eines kommerziellen Datenbankmanagementsystems in Deutschland und insbesondere gemeinsam mit einer Universität etwas Undenkbares. In der DDR war es ein Beispiel für die unmittelbare Überführung von Forschungsergebnissen in die Praxis. Diese Form von Technologietransfer wird von dem ehemaligen Robotron-Themenleiter und einer ehemaligen Forschungsstudentin der TU Dresden mit dem Abstand von zwei Jahrzehnten kritisch betrachtet.

1 DABA1600 im internationalen Vergleich

DABA 1600 war das erste und einzige vorbildfreie relationale Datenbankbetriebssystem (DBBS)[1] in der DDR, das die später standardisierte Datenbanksprache SQL unterstützte. Die erste Version von DABA 1600 wurde vom ZFT Robotron in Kooperation mit der Sektion Informationsverarbeitung der TU Dresden zu Beginn der achtziger Jahre für PDP-11-kompatible Systeme entwickelt[2]. Eine Datenbank auf einem Kleinrechner betreiben zu wollen, wie er z.B. damals mit den Anlagen der Familie Robotron K 1600 zur Verfügung stand, das war Ende der siebziger Jahre zunächst ein durchaus ungewöhnliches Vorhaben. Zumindest wurde der Erfolg eines solchen Vorhabens von den Entwicklern des für Großrechner (R 300 bzw. die nachfolgenden ESER-Rechner) ausgelegten Datenbankbetriebssystems DBSR im damaligen Zentrum für Forschung und Technik (ZFT) des Kombinats Robotron stark angezweifelt.

Das Forschungskollektiv „Daten- und Programmbanken“ von Prof. D. Schubert hatte Ende der siebziger Jahre umfangreiche Forschungsarbeiten zur Implementierung relationaler Systeme geleistet. Ausgangspunkt dieser neuen Forschungsrichtung war die Veröffentlichung des CODDschen Relationmodells 1970 [Cod70]. Die Bedeutung dessen erkennen wir daran, dass Informationssysteme heute weltweit allgegenwärtig und ohne relationale Systeme nicht mehr denkbar sind. Prof. D. Schubert gebührt der Verdienst, das Potential relationaler Datenbanken bereits damals erkannt und entsprechende Forschung begonnen

[1] Zu DDR-Zeiten haben wir den Begriff „Datenbankbetriebssystem“ statt dem heute üblichen Begriff „Datenbankmanagementsystem“ (DBMS) verwendet.

[2] Im folgenden verwenden wir als Abkürzung für das ZFT Robotron „ZFT“ und für die Sektion Informationsverarbeitung der TU Dresden „TUD“.

zu haben.

Was waren die DBBS bzw. SQL-Implementierungen, mit denen wir DABA 1600 vergleichen mußten? Einen guten Rückblick auf die Entwicklung der internationalen relationalen Systeme vermittelt der Bericht zum Treffen ehemaliger SQL-Pioniere [M+97].

Das **System R** war der bekannteste Prototyp für relationale Datenbankmanagementsysteme mit der ersten SEQUEL- (später SQL-)Implementierung [C+81]. Das IBM Research Laboratory San Jose in Kalifornien, in dem das System R entstand, war praktisch das Experimentallabor für alle späteren relationalen Systeme. Im deutschsprachigen Raum hat Theo Härder seine Erfahrungen bei experimentellen Untersuchungen im System R-Projekt in einem hervorragenden Buch [Hä78] niedergeschrieben, welches für die DABA 1600-Entwickler die „Bibel“ für die eigene Implementierung wurde. Wichtige Themen waren zum Beispiel die Optimierung von SQL-Anfragen, Implementierung von Zugriffspfaden, Zugriffssicherung und die Transaktionsverarbeitung. Aus dem System R gingen etwa zeitgleich zu DABA 1600 die großen bekannten kommerziellen SQL-Systeme von IBM hervor (1982 SQL/DS auf VM/VSE und 1983 DB2 auf MVS V1).

1977 begannen Larry Ellison, Bob Miner und Ed Oates die Entwicklung von **Oracle**, eines der bekanntesten relationalen Systeme. 1979 wurde das erste Produkt an Kunden ausgeliefert: Oracle RDBMS Version 2 lief wie das spätere DABA 1600 auf einer PDP-11 mit dem Betriebssystem RSX-11. Diese Version wurde jedoch schnell auf das leistungsfähigere VAX-System portiert [Wha96].

Etwa parallel zum System R wurde an der University of California, Berkeley **Ingres** entwickelt, das ebenfalls auf PDP-11-Maschinen lief, jedoch eine andere Nutzerschnittstelle (QUEL anstelle von SQL) unterstützte [SWKH76]. Als Nachfolger des als Open Source-Systems bereitgestellten Ingres ist heute noch PostgreSQL als frei verfügbares objektrelationales System im Einsatz.

Aus TUD-Sicht ([TD83a]) entsprach 1983 DABA 1600 dem Entwicklungsstand vom System R und repräsentierte

> „... in einigen Komponenten eigenständige Weiterentwicklungen. Dies betrifft die Zielsprache des SQL-Compilers (DABA-Code) sowie eine die optimale Dekomposition von SQL-Anfragen unterstützende „innere Zwischensprache“ ASQL, in der syntaktisch und semantisch korrekte SQL-Anfragen als Ergebnis der entsprechenden Analyseprozesse bereitgestellt werden.“

In der verwendeten DDR-Terminologie wurde dieser Stand als „BES(2)“ (Mitbestimmung des Weltstandes zum Zeitpunkt der Einführung des Ergebnisses) deklariert. Es ist aus heutiger Sicht nicht ganz klar, ob diese Einschätzung in den einzelnen Punkten, die in den A4-Verteidigungsunterlagen [TD83b] im Detail nachzulesen sind, bei detaillierter Kenntnis des westlichen Forschungsstandes standgehalten hätte.

Aufgrund der beschränkten personellen und Rechnerressourcen wurde eine aus Anwendersicht relevante Teilmenge der vorgeschlagenen „Ursprache“ SEQUEL(2) implementiert. Diese Sprache nannten wir SQL 1630. Wir waren uns der Spracheinschränkungen wohl bewußt. Schaut man jedoch heute auf die Entwicklung des SQL-Sprachstandards und die Konformität der kommerziellen SQL-Systeme, so sind die getroffenen Einschränkungen

im Verhältnis zum Entwicklungszeitpunkt üblich gewesen. Von Bedeutung war, dass SQL 1630 dem ANSI/ISO-Standardisierungsprozess gefolgt ist und damit stets SQL-kompatible Anwendungslösungen möglich waren.

Betrachtet man allerdings das Gesamtsystem und nicht nur die oberen Schichten von DABA 1600 (Datensystem mit dem SQL 1630-Compiler), so ergibt sich ein anderes Bild. Beispielsweise hat Oracle 1983 bereits seine Version 3 ausgeliefert, die sich durch eine hohe Portabilität auszeichnete und auf verschiedenen Rechnerklassen (Mainframes, Kleinrechner und PC) lief[3]. In der weiteren Entwicklung haben sich weltweit nur wenig relationale Systeme auf dem Markt etabliert, darunter kein deutsches Produkt.

Während das System/R vorwiegend in PL/1 und Assembler und Oracles Version 2 noch in Assembler (Version 3 dann in C) geschrieben wurde, haben wir für die Implementierung des SQL 1630-Compilers die aus softwaretechnischer Sicht wesentlich fortschrittlichere Sprache CDL verwendet. CDL (Compiler Description Language) zeichnete sich durch gute Lesbarkeit, Wartbarkeit, Effizienz, Erweiterbarkeit und eine hohe Portabilität aus. Wir konnten in unserer Arbeit auf das an der TUD entwickelte CDL-Programmiersystem und die vorhandenen methodischen Erfahrungen der Kollegen zurückgreifen [Hei85]. Heute ist diese Compilerentwicklungstechnologie durch moderne, Java-basierte Parsergeneratoren und eine Vielzahl weiterer Werkzeuge, insbesondere zur Unterstützung des sogenannten „Language Engineerings“, überholt. Bezogen auf den Stand der Technik zu Beginn der achtziger Jahre konnte aber, im Wesentlichen im Rahmen von Diplomarbeiten, ein effizient arbeitender und portabler SQL-Compiler entwickelt werden, der weit über den Status eines universitären Prototypen hinausging. Die Verwendung von CDL war dabei eine wichtige Voraussetzung für die spätere Wartbarkeit der TUD-Implementierung als Teil der kommerziellen Software.

Nach der Auslieferung von DABA 1600 an Kunden, Erfahrungen aus Tests und späteren Anwendungen von DABA 1600 resümieren führende ZFT-Entwickler 1987 [MS87c]:

> „Das System arbeitet auf der vorgegebenen Basismaschine mit einem anspruchsvollen Funktionsumfang bei akzeptabler Effizienz im Dialog- und Verarbeitungsbetrieb. ... Die konsequente Einhaltung der Softwaretechnologie im Entwurf und während der Implementierung hat sich bewährt. Sie war eine Voraussetzung für das Sichern einer hohen Softwarequalität. ... Fehlerbearbeitungen und Anpassungsarbeiten konnten bisher auf Grund des modularen Systemaufbaus ohne konzeptionelle Probleme gelöst werden.“

2 DABA1600 als Kooperationsprojekt zwischen Wirtschaft und Universität

Im Jahr 1979 wurde innerhalb von Robotron durch den Direktor für Forschung und Entwicklung Prof. G. Merkel eingeschätzt, dass die „Technologie“ des Forschens, Entwi-

[3] Unter http://www.oracle.com/timeline/ ist die Geschichte von Oracle in der Zeit von 1977-2007 sehr anschaulich dargestellt.

ckelns, Konstruierens und Überleitens innerhalb von Robotron entschieden verbessert werden muss [dTD89]. Es ist anzunehmen, dass eine Maßnahme zur Verbesserung dieser Situation die enge Zusammenarbeit mit wissenschaftlichen Einrichtungen war. Zugleich sollte seit der Hochschulreform von 1968 generell der Technologietransfer von Universitäten zur volkseigenen Industrie durch formelle Forschungsverträge beschleunigt werden [Sch03]. Im Folgenden wird über das Kooperationsprojekt DABA 1600 als eines der Beispiele für die Zusammenarbeit und den damit verbundenen Technologietransfer Ende der siebziger und in den achtziger Jahren berichtet.

2.1 Wie war die Zusammenarbeit zwischen der TU Dresden und dem VEB Kombinat Robotron geregelt?

Im Jahr 1977 wurde ein „Komplexvertrag zwischen der TU Dresden und dem ZFT Robotron“ unterzeichnet, der durch Jahres- oder Zweijahresvereinbarungen konkretisiert wurde. Während dieser Komplexvertrag im Universitätsarchiv auffindbar war [TD77], haben die Autoren einen konkreten Vertrag über das DABA 1600-Projekt nicht finden können.

Im Universitätsarchiv sind verschiedene Briefe, Aktennotizen und Vorlagen enthalten, die regelmäßige Beratungen des Rektors der TU Dresden mit dem Generaldirektor des Kombinates Robotron, das heißt Treffen auf höchster Leitungsebene dokumentieren [dTD89]. Teilnehmer waren neben dem Rektor und dem Generaldirektor auf TU-Seite die Prorektoren, Sektionsdirektoren der Sektionen Informationsverarbeitung, Informationstechnik und Elektrotechnik sowie auf Robotron-Seite die Direktoren und der Beauftragte für die Komplexvereinbarung des Kombinates Robotron mit der TU Dresden (Prof. Jordan). Zur politischen Sicherstellung der Zusammenarbeit waren zusätzlich die Parteibeauftragten beider Seiten dabei. Themen waren neben der eigentlichen wissenschaftlich-technischen Zusammenarbeit wichtige Fragen wie die Versorgung der TU Dresden mit Rechentechnik, Export von Produkten in das NSW (nichtsozialistische Wirtschaftsgebiet), der Kaderaustausch und die Anforderungen an die Ingenieurausbildung und Weiterbildung. Komplexverträge bzw. -vereinbarungen gab es mit vielen namhaften Kombinaten und wissenschaftlichen Einrichtungen, wie z.B. der Akademie der Wissenschaften. Es gab Muster zur Gestaltung der Verträge (Komplexvertrag, Wirtschaftsvertrag, Export- und Inlandvertrag, Koordinierungsvertrag, Lizenzvertrag) ebenso wie für einen „Massnahmeplan zur Umsetzung eines Wirtschaftsvertrages“. Die Sektionsdirektoren mußten dem Rektor der TU Dresden zu „überführungsgefährdeten Forschungsleistungen und zu diesbezüglichen Problemen“ Bericht erstatten (siehe „Signalberichterstattung“ vom 24.6.1983 in [fNuT90]). Interessant darin ist die Aussage, dass die Sektion Informationsverarbeitung die Überführung von Forschungsergebnissen durch die Übernahme in die vom Kombinat Robotron vertriebenen Systemunterlagen als die wirkungsvollste Art ansah. Allerdings ergaben sich oftmals Schwierigkeiten durch fehlende Kapazitäten und Qualifikation beim Robotron-Vertrieb.

2.2 Wie lief das mit DABA1600?

Im Verlaufe der Arbeiten wurden auf vertraglicher Basis neben der damaligen Sektion Informationsverarbeitung der Technischen Universität Dresden (TUD) auch Mitarbeiter von Robotron-Vertrieb (RVD) Cottbus in die Arbeiten zu DABA 1600 einbezogen. Die Koordinierung aller Zuarbeiten lag bis zur Beendigung beim Themenkollektiv im ZFT bzw. später RPD (Robotron-Projekt Dresden). Die Zusammenarbeit mit der TUD wurde von den Mitarbeitern des ZFT bzw. RPD als sehr positiv und als dringend notwendige personelle Verstärkung empfunden, zumal sich daraus auch der Zugang zu ausländischer Fachliteratur leichter realisieren ließ, als das damals von ZFT bzw. RPD aus möglich war. Die räumliche Nähe und die dadurch gegebenen besseren Kommunikationsmöglichkeiten (im Vergleich zu den am Thema mitarbeitenden Kollegen des RVD Cottbus) erwiesen sich als vorteilhaft.

Das DABA 1600-Kooperationsprojekt hat eine Vorgeschichte, die bereits 1977 mit dem Komplexvertrages zwischen dem ZFT und der TU Dresden begann. Darin ist das Thema „Daten und Programmbanken“ in Verantwortung von Prof. D. Schubert mit 4,1 VBE Fachpersonal und „ohne Einnahmen“ verankert. Hier kann sich die TUD-Autorin an ihr eigenes Ingenieurpraktikum 1978 während des Studiums Informationsverarbeitung erinnern. Wir saßen ein Semester lang unter „Robotronern“, um die neuesten internationalen Veröffentlichungen auf dem Gebiet der relationalen Datenbanken aufzubereiten und diese mit den Robotronern zu diskutieren. Parallel dazu hatten wissenschaftliche Mitarbeiter (V. Wenk, B. Keller, M. Starruß, H. Hartmann) an der TUD mit experimentellen Untersuchungen zur Implementierung des Relationenkonzeptes für verschiedene Plattformen begonnen. Diese vorlauforientierten Forschungsarbeiten mündeten im Herbst 1979 in die Vorbereitung des DABA1600-Projektes (zugleich Staatsplanaufgabe). In den „Aufgabenblättern für Forschung“ [TD] wurde nicht nur eine wissenschaftliche/technische Zielstellung, sondern auch eine volkswirtschaftliche/gesellschaftliche Zielstellung formuliert. So war z.B. die Aufnahme des SQL1630-Compilers in die Standardsoftware für K1630 (Einführungsjahr 1984) geplant.

Ein kritischer Punkt waren, nicht nur auf Robotron-Seite, die Ressourcen. Neben Problemen mit verfügbarer Rechentechnik stand auch die Frage nach Entwicklungskapazität. Ein Compiler konnte damals (wie oben bereits beschrieben) nicht so effizient, wie z.B. heute im Rahmen einer einzelnen Diplomarbeit, implementiert werden. Die universitäre DDR-Lösung war die Nutzung umfangreicher studentischer Ressourcen. Erinnert sei an das Jugendobjekt „Datenbanken“, 10 Diplomarbeiten und etliche Ingenieurpraktikums- und Großen Belege[4]. Softwaredokumentationen wurden zum Bestandteil (Anlagen) von Diplomarbeiten. Vergleicht man diese Praxis zur Erhöhung der universitären Entwicklungskapazität mit heutigen Projekten erkennt man als Unterschiede, dass die Entwicklung in politisch geführten Strukturen straffer organisiert wurde und die entstandene Software kommerziellen Charakter hatte. Die Mitarbeit von Studierenden ist allerdings heute wie damals unerläßlich für den Erfolg von Projekten und zugleich für die Studierenden Motivation und Schule für die Vorbereitung auf das Berufsleben.

[4] Genannt seien die Arbeiten von W. Reime, E. Reime, F. Schönefeld, A. Fischer, H.-J. Fischer, U. Zickert, H.-U. Sauermilch, A. Urban, A. Scheifhacken, A. Schneegaß, H.-U. Mäser, K. Bakardiew, K. Herrmann, I. Rohner

3 DABA1600 und die Alltagsprobleme aus der Sicht der Entwickler

Der Alltag der Entwickler war, gemessen an heutigen Arbeitsbedingungen, sehr mühsam. Das hatte nicht gerade einen fördernden Einfluß auf den Fortschritt im Projekt. Hans-Jürgen Brosch hat bereits 2006 ausführlich über die Bedingungen der Forschungsarbeit von Robotronern berichtet [Bro06]. Wir möchten diese Darstellung um ein paar ganz persönliche Erfahrungen sowohl aus ZFT- als auch aus TUD-Sicht ergänzen.

Damals war nicht daran zu denken, dass jeder Programmierer einen PC an seinem Arbeitsplatz hatte. Die Programme mussten mit Papier und Bleistift entworfen werden. Dann hieß es, Testzeit im Rechenzentrum anzumelden. Die wurde rund um die Uhr vergeben. Im Kollektiv waren auch Frauen, die im „Nebenberuf“ eine Familie zu versorgen hatten. Nicht ungewöhnlich war, dass diese Mitarbeiterinnen, wie auch ihre männlichen Kollegen, z.B. Rechenzeit von 23 bis 3 Uhr nachts zugeteilt bekamen. Einzige Vergünstigung: Für die nächtliche Heimfahrt wurde ihnen das Taxi bezahlt.

Noch erinnert sich der ZFT-Autor an eine mehrtägige sehr starke Regenperiode, der in den 80er Jahren das Dach der Rechenstation nicht stand hielt. Die hier untergebrachten Importrechner mussten mit Kunststoffplanen abgedeckt werden, schließlich fielen sie trotzdem durch Wasserschaden aus. Der Testbetrieb war langfristig gestört, die Plantermine ernstlich gefährdet.

Unvorhersehbare Ausfälle von Mitarbeitern bereiteten dem Themenleiter große Schwierigkeiten, waren sie doch in einem kleinen Mitarbeiterkollektiv, wie das bei DABA 1600 der Fall war, kaum zu kompensieren. Dazu ein Beispiel: Eine als Programmiererin eingesetzte Diplom-Mathematikerin leistete sehr gute Arbeit, war ideenreich, zuverlässig und pünktlich. Ihr einziger „Nachteil“: Sie hatte eine Familie mit zwei oft kranken Kleinkindern zu versorgen.

Neben solchen in der Privatsphäre der Mitarbeiter begründeten Ausfällen gab es auch genügend von oben verordnete Ausfälle der normalen Arbeitszeit. Der ZFT-Autor denkt an die „Jubel-Kommandos“, wenn z.B. beim Besuch ausländischer Staatsgäste oder zu bestimmten Veranstaltungen in Dresden der ganze Betrieb, einzelne Struktureinheiten oder eine vorgegebene Zahl von Mitarbeitern „delegiert“ wurden. So geschehen z.B., als dem damaligen afghanischen Staatschef Nadjibulla bei einer Kundgebung auf dem Theaterplatz, dem rumänischen Potentaten Ceausescu oder dem Kubaner Fidel Castro bei ihren Rundgängen durch die damalige Leningrader Straße zuzujubeln war. Auch bei der Einweihung des Lenin-Denkmals vor dem Hauptbahnhof (unserem „Bahnhofsvorsteher“ wie der Volksmund sagte) hatten Mitarbeiter während der Arbeitszeit zum Jubeln und Schwenken von Fahnen und „Winkelementen“ präsent zu sein.

Noch wesentlich schwerer waren vom Themenkollektiv langfristige Abstellungen zu verkraften. Konnte man das in Katastrophensituationen noch verstehen, wenn z.B. Klassenfeind Nummer 4 (das war nach Frühling, Sommer und Herbst der Winter) wieder tückisch zugeschlagen hatte und Dresdens Straßen durch Eis und Schnee unpassierbar waren. Da fanden sich dann hochqualifizierte diplomierte und promovierte Fachkräfte statt am Rechner, am Schreibtisch oder im Labor, mit der Schneeschippe auf der Straße wieder. Gab es in Kombinatsbetrieben oder in anderen volkswirtschaftlich wichtigen Einrichtungen

Planrückstände, war dort „sozialistische Hilfe“ zu leisten. Zur Erntezeit erfolgten Delegierungen in die LPG, im Sommer zur Betreuung des betrieblichen Kinderferienlagers, bei Engpässen in der Energieversorgung galt es in der Braunkohleindustrie zu helfen. Als Höhepunkt solcher die fachliche Arbeit stark behindernden Aktionen muss die Abstellung von in der Entwicklung tätigen hochqualifizierten Mitarbeitern zur städtischen Müllabfuhr gesehen werden.

In der persönlichen Erinnerung des ZFT-Autors haften aus dieser Zeit auch noch die heute lächerlich wirkenden Anstrengungen zur verbalen Abgrenzung in der internationalen (westlichen) Fachterminologie. Im ZFT gab es damals zwei Struktureinheiten, in denen „Maschinenorientierte Systemunterlagen“ (MOS) und „Problemorientierte Systemunterlagen“ (POS) entwickelt wurden. Was war unter dem Begriff „Systemunterlagen“ zu verstehen? Außerhalb der DDR verstand man darunter schlicht und einfach „Software“. Doch dieses Wort musste für uns tabu bleiben, kam es doch aus der westlichen Welt. Wie an der Basis über diese „Abgrenzungsmanie“ auch bei anderen international üblichen Fachbegriffen gedacht wurde, lässt ein damals in Mitarbeiterkreisen kursierender Spruch erkennen: „Kaum möcht einer es mal wagen, leise „IBM“ zu sagen, da schreit schon die Prominenz: Freunde, das ist Konvergenz!“ Die Konvergenztheorie besagte, dass sich im Verlauf der technischen Entwicklung die Gesellschaftssysteme Kapitalismus und Sozialismus allmählich immer mehr annähern und schließlich miteinander verschmelzen würden. Das war natürlich keinesfalls im Sinne der damaligen Obrigkeit in der DDR und wurde schärfstens verdammt.

Die DABA 1600-Mitarbeiter und -Forschungsstudenten an der TUD haben ähnliche Erfahrungen gemacht. Für sie war es allerdings um die Verfügbarkeit der notwendigen Kleinrechentechnik (K1630 mit für Datenbankarbeit ausreichender Peripherie) und eigener Datenträger (Kassettenplatten) noch schlechter bestellt. Es gab keinen eigenen K1630 und der Zustand der Datenerfassungstechnik an der TUD war generell schlecht. Der Ausweg war die Nutzung von vorwiegend Nacht- und Wochenendrechenzeiten im Testrechenzentrum des VEB Robotron-Projekt Dresden.

Ein weiteres Hemmnis waren die geringen personellen Entwicklungskapazitäten. Wird heute die Frage gestellt, welcher Aufwand zur Realisierung von DABA 1600 erforderlich gewesen ist, so kann vom damaligen Themenleiter nur eine sehr grobe Abschätzung gegeben werden, da beweiskräftige Unterlagen nicht mehr verfügbar sind. An der TUD wurden 1980-1984 pro Jahr Vertragslaufzeit max. 7 VBE bereitgestellt, wobei die Studierenden mit 4,4 VBE den größten Anteil darstellten (1983, siehe [TD]). Beim ZFT Robotron waren das in der Erinnerung des Themenleiters etwa 10 durchgehend Beschäftigte über eine Gesamtlaufzeit von 9 Jahren. Die abgeschätzte Gesamtkapazität von maximal 120 Mannjahren entspricht in etwa den durchschnittlich im Zeitraum 1974 - 1981 eingesetzten 15 Entwicklern vom experimentellen System R, von denen in [M$^+$97] berichtet wird. Aussagen über eingesetzte Kapazitäten bei der Entwicklung kommerzieller Systeme, wie es letztendlich DABA 1600 war, wurden nicht veröffentlicht. Trotzdem „sickerte“ durch, dass wesentlich mehr Entwickler an Systemen wie DB2 oder Oracle beteiligt waren und nach wie vor sind.

Die planmäßige Überführung von Forschungsleistungen war auch dadurch beeinträchtigt, dass jederzeit Kollektive beim ZFT „umorientiert“ werden konnten, was im Fall von DA-

BA 1600 1983 auch passierte. Eine wichtige softwaretechnische Erfahrung, die insbesondere die TUD-Entwickler machen mußten war der außerordentlich hohe Abstimmungsaufwand mit dem Partnerkollektiv auf Grund eines sehr breiten Interfaces zwischen den beiden Teilsystemen (Datensystem und Zugriffssystem[5]). Insbesondere ergaben sich daraus Probleme, weil die Partner zu unterschiedlichen Zeitpunkten an einer Interfacespezifikation interessiert waren.

Und nicht zuletzt sei auf das Problem hingewiesen, dass die TUD-Mitarbeiter und Forschungsstudenten in ihrer wissenschaftlichen Arbeit durch die notwendige aufwendige Entwicklungs- und Projektmanagementtätigkeit eingeschränkt wurden. Dazu kamen die Schwierigkeiten bei der Beschaffung von Forschungsliteratur. Wir haben z.B. Dutzende von durch „offizielle“ Stellen geprüfte und vorgedruckte Postkarten an Forscher im westlichen Ausland mit der Bitte um Bereitstellung von wissenschaftlichen Artikeln geschickt, von denen nur einzelne erfolgreich waren. Eine weitere Ressource, der Zugriff auf Open Source-Software wie Ingres, blieb uns ebenfalls verwehrt.

Zur Leistungsstimulierung/Motivierung der wissenschaftlichen Mitarbeiter war es möglich, Prämienvereinbarungen abzuschließen. Die finanziellen Mittel wurden im Rahmen der TU Dresden aus Erlösbeteiligungen von Lizenzvergaben bereitgestellt. Studenten konnten auch in Zielprämienvereinbarungen einbezogen werden. Diese Dinge waren allerdings für die Mitarbeiter wenig transparent.

4 Der DABA1600-Lebenszyklus

Die Entwicklung von DABA 1600 begann 1979 in der Struktureinheit Anwendung der Rechentechnik im ZFT. Im Rahmen eines wenig später aus innerbetrieblichen Gründen notwendigen Führungswechsels wurde der ZFT-Autor ca. 1980/81 als Gruppen- und Themenleiter für dieses Vorhaben eingesetzt. Er hat das Entwicklerteam bis zur 1988 erfolgten Überleitung in die Produktion und bis zum Sommer 1989 bei der sich anschließenden Produktionsbetreuung geleitet.

Die TU Dresden hat in den Jahren 1981-1986 auf inhaltlicher Basis eines Anfang 1981 vom ZFT bestätigtes DABA 1600-Pflichtenheftes an der Softwareentwicklung von DABA 1600 mitgearbeitet und neben dem SQL 1630-Compiler den SORT-Operator für das Zugriffssystem für DABA 1600 implementiert. Zum Zeitpunkt der A4-Verteidigung am 29.3.1983 [TD83b] lagen gleichzeitig die Arbeitsergebnisse für die E4-Stufe (TKO-Test) vor. Danach verpflichtete sich die TU Dresden zur Mitarbeit an der Kopplung des Compilers mit dem Zugriffssystem und an der Auswertung der Erprobung des Gesamtsystems sowie zur schrittweisen Beseitigung von Einschränkungen der vorliegenden Compilerversion. Die Kooperation ZFT-TUD wurde im November 1984 mit dem Ende der Industrieerprobung und der Freigabe von DABA 1600 an die Anwender offiziell abgeschlossen ([Sch85]). Das endgültige Ende der Mitwirkung der TUD an der Wartung von DABA 1600 ist allerdings erst im Frühjahr 1986 gewesen. Bis zu diesem Zeitpunkt hat die TUD-Autorin nach dem Ende ihres Forschungsstudium 1983 an der Fehlerbehebung in DABA

[5] Wer sich heute noch für die Architektur von DABA 1600 interessiert, sei [MS87d] empfohlen.

1600 mitgearbeitet.

Denkt man an die dem Vorhaben DABA 1600 anfangs im ZFT entgegen gebrachte Skepsis, so mutet es beinahe kurios an, dass der erfolgreiche Abschluss dieser Entwicklung gerade der ehemaligen „Konkurrenz“ von Großrechnerseite zu verdanken war. In der Struktureinheit, in der die DABA-Entwicklung begonnen hatte, ergaben sich ca. 1987 durch neue Aufgabenstellungen große Kapazitätsschwierigkeiten. Deswegen sollte DABA 1600 abgebrochen und die hier tätigen Mitarbeiter für andere Aufgaben eingesetzt werden. Inzwischen hatte sich im internationalen Maßstab die Stellung des Kleinrechners entscheidend zu dessen Gunsten verändert. Das erkannte die Leitung im Bereich Großrechner-Datenbanken im RPD rechtzeitig und bot den DABA 1600-Entwicklern die Übernahme an. Dieser Strukturwechsel wurde realisiert und damit neben DBSR auch der erfolgreiche Abschluss der Entwicklung von DABA 1600 bei RPD gesichert. Praktisch hieß das damals, dass nach Verteidigung der Entwicklungsstufe ÜK 11 den Käufern des Systems jetzt Magnetbänder und gedruckte Anwenderdokumentationen sowie eine vertraglich zu vereinbarende befristete Unterstützung bei der Anwendung angeboten bzw. übergeben werden konnten. Die Überwachung der nachfolgenden praktischen Nutzung des Kleinrechner-Datenbankbetriebssystems in etwa 100 Betrieben überwiegend in der DDR, in denen DABA 1600 zum Einsatz gekommen ist[6], wurde unter der Bezeichnung Produktionsbetreuung von dem jetzt zahlenmäßig verkleinerten Team bei RPD durchgeführt und hat die Richtigkeit dieser Entscheidung voll bestätigt.

In der einschlägigen Fachpresse ist darüber mehrfach berichtet worden, wie auch Veröffentlichungen schon während der Entwicklungszeit dazu beigetragen haben, DABA 1600 mit seinem Leistungsumfang in der DDR bekannt zu machen [BEW82], [Sch85], [Wig86], [MS87c], [MS87a], [MS87d], [MS87b]. Die Ausbeute an wissenschaftlichen Schriften beschränkt sich auf zwei Dissertationen [Sta83], [Tem84] und eine Reihe von Beiträgen in Zeitschriften (z.B. [SW85]), Schriftenreihen und auf Konferenzen innerhalb des RGW-Bereiches. Im Juni 1988 wurde getragen von VEB Robotron-Projekt in Dresden die Nutzergemeinschaft DABA 1600 gegründet. Anliegen war der Austausch von Erfahrungen, die bei der Arbeit mit DABA 1600 gesammelt worden sind. In Diskussionsbeiträgen berichteten die Anwender über eine stabile Arbeitsweise des Datenbankbetriebssystems für ihre spezifischen Aufgaben [Wig88].

Als mit dem Ende des Kombinats Robotron auch keine K 1630-Rechner mehr produziert wurden, war der Einsatz von DABA 1600 gegenstandslos geworden. Ob bei ehemaligen Anwenderbetrieben, soweit sie die Wende überstanden haben, DABA 1600 noch eine Zeit lang genutzt wurde, ist dem Autor, der im Sommer 1989 in Rente gegangen und damit aus dem RPD ausgeschieden ist, nicht bekannt. Geblieben sind bei den DABA 1600-Entwicklern wertvolle technologische Erfahrungen in der Softwareentwicklung und ein tiefes Verständnis der Datenbanktechnologie. Die TU Dresden/die TUD-Autorin hat an der SQL-Standardisierung (zunächst in Form einer TGL in der DDR, nach 1990 in den zuständigen DIN- und ANSI/ISO-Gremien) mitgearbeitet. Nach der Wende hat die TUD-Autorin außerdem Vorlesungen über die Architektur und Implementierung von Datenbanksystemen an der TU Dresden gehalten.

[6] Darunter waren der VEB Chemieanlagenbau Germania, Karl-Marx-Stadt, der VEB Waggonbau Ammendorf, der VEB DKK Scharfenstein und das Moskwitsch-Automobilwerk, Moskau/UdSSR [MJ+06]

5 Was haben wir gelernt?

5.1 Sicht des ZFT-Kooperationspartners

Die Arbeit an DABA 1600 hat gezeigt, dass im Kombinat Robotron zu Anfang der siebziger Jahre erarbeitete prognostische Einschätzungen wenig zutreffend waren. Es wurden zwei (extreme) Alternativen diskutiert: Liegt die Zukunft der Datenverarbeitung bei einem für die DDR oder den gesamten Ostblock in Berlin bzw. Moskau arbeitenden Zentralrechner - für den allerdings wegen der fehlenden Datenfernübertragungsmöglichkeiten keinerlei Chancen bestanden, oder bei künftigen Kleinstanlagen für jeden Arbeitsplatz - wozu nun wieder die Bauelementeindustrie nicht leistungsfähig genug war. Die DDR-Entscheidung führte zu dezentralen Rechenzentren für jedes Kombinat bzw. jeden Betrieb, ausgestattet mit mittelgroßen Anlagen (R 300- und nachfolgend ESER-Rechner).

Dass unter diesen Voraussetzungen die Arbeit an DABA 1600 überhaupt aufgenommen und durchgeführt werden konnte, bis sich im internationalen Maßstab die künftige Bedeutung des Kleinrechners abzeichnete (als deren Fortsetzung die Autoren heute den Einsatz des PC am Arbeitsplatz sehen), muss wohl als Glücksfall angesehen werden. Inzwischen ist heute eine Datenbankarbeit mit den von PC oder Notebooks gebotenen Möglichkeiten selbstverständlich.

Die Zusammenarbeit des Entwicklerteams von DABA 1600 mit der TUD stellte eine nicht nur quantitativ sondern auch qualitativ wirkende neuartige Verstärkungskomponente dar. Dazu haben sicherlich auch die von zwei Mitarbeitern während der Themenlaufzeit vorgenommenen Arbeiten für ihre Promotion beigetragen.

5.2 Sicht des TUD-Kooperationspartners

Für die TUD-Beteiligten war das DABA1600-Projekt zeitweise ein schmerzlicher Prozess. Im Jahresforschungsbericht 1983 der Sektion 08 wurden von Prof. Schubert die Probleme vorsichtig formuliert und die fehlende wissenschaftspolitische Unterstützung kritisiert [TD83a]. Diese Probleme waren 1983 nicht neu, wurden nur leider von „zentraler Seite“ nicht bzw. „ohne erkennbare Reaktionen“ zur Kenntnis genommen.

So war die „Orientierung auf den Weltstand“ Augenwischerei. Dieser wurde in den Berichten zu Beginn und nicht zum Abschluß der Arbeiten als Maßstab genommen (siehe der Vergleich von DABA 1600 zu Oracle in Abschnitt 1). Ein Grund für den größer werdenden Abstand zur Weltspitze waren viele der Probleme, die im Abschnitt 3 beschrieben sind. Insbesondere die geringe Verfügbarkeit fortgeschrittener und in der weltweiten Entwicklung nicht mithaltender Rechentechnik ist bei der Entwicklung von Datenbankbetriebssystemen ein Problem.

Während die Forschungskooperation zu wissenschaftlichen Einrichtungen aus ZFT-Sicht generell als wesentliche Verstärkung der kombinatseigenen Forschungskapazitäten eingeschätzt wurde [Jor06], fühlten sich die wissenschaftlichen Mitarbeiter oftmals als reine

Programmierer mißbraucht. Robotron wollte 1983 für die TU Dresden nur noch Entwicklungsarbeiten und keine Vorlaufforschung finanzieren [TD83a]. Es konnte jedoch trotz des Wunsches nach Technologietransfer nicht Ziel einer Universität sein, als verlängerter Arm eines Softwarehauses umfangreiche Programmierarbeiten zu leisten und damit die wissenschaftliche Arbeit zu vernachlässigen. Damit war die, wie in Abschnitt 2 beschrieben, verfolgte Strategie der Universität, Forschungsergebnisse als Software-Artefakte unmittelbar in die Standardsoftware von Robotron aufzunehmen mehr als zweifelhaft. Dieses Spannungsfeld zwischen Universität und Softwareindustrie beobachtet die TUD-Autorin heute noch in vielen Kooperationsbeziehungen mit Firmen, insbesondere bei Diplomprojekten von Studenten. Einen effizienten Technologietransfer zu realisieren ist damit nach wie vor eine große Herausforderung für beide Seiten.

Literatur

[BEW82] G. Bulla, J. Ernst und K.-H. Wiggert. Problemorientierte Systemunterlagen für das Mikrorechnergerätesystem robotron K 1630 - Datenbankbetriebssystems DABA 1600. *Neue Technik im Büro*, 26(6):168 ff., 1982.

[Bro06] Hans-Jürgen Brosch. Informatik-Grundlagenforschung im Kombinat Robotron - ein Erfahrungsbericht. In Friedrich Naumann und Gabriele Schade, Hrsg., *Informatik in der DDR - eine Bilanz*. GI-Edition, Lecture Notes in Informatics, 2006.

[C+81] Donald D. Chamberlin et al. A History and Evaluation of System R. *Communications of the ACM*, 24(10):632–646, 1981.

[Cod70] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[dTD89] Rektor der TU Dresden. Korrespondenz mit Sektion 08, 1974-1989. Universitätsarchiv der TU Dresden, Akte Prorektor für Naturwissenschaft und Technik Nr. 4490/112.

[fNuT90] Prorektor für Naturwissenschaft und Technik. Kooperationsbeziehungen mit Kombinat Robotron , 1968-1990. Universitätsarchiv der TU Dresden, Akte Rektor Nr. 250.

[Hä78] Theo Härder. *Implementierung von Datenbanksystemen*. Hanser, 1978.

[Hei85] Monika Heiner. Echtzeit-CDL: Eine Fachsprachenrealisierung mittels CDL. *rechentechnik/datenverarbeitung*, 22(2):21–24, 1985.

[Jor06] Enno Jordan. Überblick über die Wissenschaftskooperation des VEB Kombinat Robotron. Arbeitsgruppe Rechentechnik in den Technischen Sammlungen Dresden, Januar 2006. http://robotron.foerderverein-tsd.de.

[M+97] Paul McJones et al. The 1995 SQL Reunion: People, Projects, and Politics. SRC Technical Note 1997-018, August 1997. http://www.research.digital.com/SRC/.

[MJ+06] Gerhard Merkel, Siegfried Junge et al. Sammlung von Beiträgen zur Geschichte der Zentralen Forschungs- und Entwicklungseinrichtung des Kombinates Robotron. Arbeitsgruppe Rechentechnik in den Technischen Sammlungen Dresden, Februar 2006. http://robotron.foerderverein-tsd.de.

[MS87a] Franz Menhart und Jürgen Stumpf. Das Datenbankbetriebssystem DABA 1600 Version 2.0. *rechentechnik/datenverarbeitung*, 24(s):16–18, 1987.

[MS87b] Franz Menhart und Jürgen Stumpf. SQL 1630 - Datenbanksprache von DABA 1600. *rechentechnik/datenverarbeitung*, 24(9):15–18, 1987.

[MS87c] Franz Menhart und Jürgen Stumpf. Wertung des relationalen Datenbankbetriebssystems DABA 1600. *rechentechnik/datenverarbeitung*, 24(4):5–7, 1987.

[MS87d] Franz Menhart und Jürgen Stumpf. Zum Entwurf des Datenbankbetriebssystems DABA 1600. *rechentechnik/datenverarbeitung*, 24(7):9–11, 1987.

[Sch85] D. Schubert. Das relationale Datenbankbetriebssystem DABA 1600 - ein weiteres Ergebnis der Kooperation von Robotron und TU Dresden. *Neue Technik im Büro*, 29(4):110 ff., 1985.

[Sch03] Manuel Schramm. Universitäts-Industrie-Beziehungen in zwei deutschen Innovationssystemen: die Beispiele Biotechnologie und Werkzeugmaschinenbau, ca. 1960-90. In *Technisierungen 86. Jahrestagung der Deutschen Gesellschaft für Geschichte der Medizin, Naturwissenschaft und Technik*, Freiberg, 2003.

[Sta83] Birgit Starke. Übersetzung und Optimierung relationaler Anfragen am Beispiel der Datenbanksprache SQL. Dissertation A, TU Dresden, 1983.

[SW85] Birgit Starke und Vera Wenk. Zur Optimierung in relationalen Datenbanksystemen. *Wissenschaftliche Zeitschrift der TU Dresden*, 34(4):103–108, 1985.

[SWKH76] Michael Stonebraker, Eugene Wong, Peter Kreps und Gerald Held. The Design and Implementation of INGRES. *ACM Transactions on Datbase Systems*, 1(3):189–222, 1976.

[TD] Sektion Informationsverarbeitung TU Dresden. 5-Jahr-Planung für die Forschung in der Sektion 08 1981 - 1985. Universitätsarchiv der TU Dresden, Akte Nr. 4163.

[TD77] WB Automatisierte Informationssysteme TU Dresden, Sektion Informationsverarbeitung. Komplexvereinbarung über wissenschaftlich-technische Zusammenarbeit zwischen dem VEB Kombinat Robotron und der Technischen Universität Dresden, 1977. Universitätsarchiv der TU Dresden, Direktor für Forschung Nr. 4658/I/10, Komplexvertrag Robotron 1977 - 1988.

[TD83a] Sektion Informationsverarbeitung TU Dresden. Jahresforschungsbericht der Sektion 08 1983 , 1983. Universitätsarchiv der TU Dresden, Akte Nr. 3968.

[TD83b] WB Automatisierte Informationssysteme TU Dresden, Sektion Informationsverarbeitung. Implementierung der relationalen Datenbanksprache SQL für Kleinrechner des Systems K1630 im Rahmen des Datenbankbetriebssystems DABA 1600. A4-Verteidigungsunterlagen, März 1983. Universitätsarchiv der TU Dresden, Prorektor für Naturwissenschaft und Technik Nr. 4490/112.

[Tem84] Ingo Temmler. Zur Implementierung der Zugriffssicherung für relationale Datenbankbetriebssysteme. Dissertation A, TU Dresden, 1984.

[Wha96] Edward Whalen. *Oracle Performance Tuning and Optimization*. Macmillan Publishing, 1996.

[Wig86] K.-H. Wiggert. Die Ausbaustufe 2.0 des Datenbankbetriebssystems DABA 1600. *Neue Technik im Büro*, 30(2):58 ff., 1986.

[Wig88] Karl-Heinz Wiggert. Nutzergemeinschaft DABA 1600 gegründet. *rechentechnik/datenverarbeitung*, 25(10):2, 1988.

Das Robotron Informationsrecherchesystem AIDOS: Software-Technologie von gestern?

Hanns-Dieter Hartmann Peter Naumann
Rolf-Dieter Wenzel Frank Zänker

Abstract: Der Vortrag ist Aspekten des Funktionsumfanges, der Systemarchitektur und der Software-Entwicklungstechnologie gewidmet, auf denen das IRS AIDOS basierte. Nach einer Einordnung des AIDOS in die Palette der Anwendungssoftware des Kombinats Robotron werden Besonderheiten des Systementwurfes betrachtet, die es von traditionellen und relationalen Datenbanksystemen unterscheidet. Nach einer Erörterung von im AIDOS unterstützten Technologien zur Inhaltsverarbeitung, die heute im Bereich des Content Managements eine breite Anwendung finden, wird das erreichte Niveau der Plattformunabhängigkeit des AIDOS vorgestellt, das mit einer Reihe von technologischen Maßnahmen angestrebt wurde. Zum Abschluss wird gezeigt, welchen Einfluss die dargestellten Entwurfs- und Technologie-Entscheidungen auf die sich unter dem Namen AIDOS in den 90-er Jahren anschließenden Softwareentwicklungen hatten.

1 Einleitung

Im Symposiumsbeitrag von H. Löhr (Die Verarbeitung unformatierter Daten am Beispiel der Beschreibung von wissenschaftlich-technischen Dokumenten) wurde auf das R300-Typenprojekt für die Information und Dokumentation eingegangen, das die Grundlage für die Speicherung und Wiederauffindung von wissenschaftlich-technischen Dokumenten im Informationssystem für Wissenschaft und Technik der DDR bildete. Mit diesem Typenprojekt wurde die Basis für die Einführung von IT-gestützten Verfahren in vielen Bibliotheken und Dokumentationsstellen, vor allem in der Industrie, geschaffen.

Mit der Bereitstellung von leistungsfähiger ESER-Technik durch das Kombinat Robotron wuchsen auch die Möglichkeiten zur Vervollkommnung der Programmunterstützung im Bereich der Information und Dokumentation.

Im Rahmen des Konzepts der sachgebietsorientierten Systemunterlagen (SOPS) wurde diese Entwicklung seitens des Kombinats Robotron in enger Zusammenarbeit und Abstimmung mit dem Zentralinstitut für Information und Dokumentation (ZIID) in Berlin vorangetrieben. Das **A**utomatisierte **I**nformations- und **Do**kumentations**s**ystem AIDOS stellte das Paradigma für die Entwicklung einer ganzen Reihe von Programmsystemen dar, die im Laufe der Zeit für die Informationsstellen der Kombinate und die öffentlichen und betrieblichen Bibliotheken in der DDR erschienen. Durch die

Zusammenarbeit mit dem **I**nternationalem **Z**entrum für **w**issenschaftlich-**t**echnische **I**nformation in Moskau (IZWTI) wurden auch der Einsatz des AIDOS über die Ländergrenzen hinaus vorbereitet und Voraussetzungen für den Internationalen Datenaustausch zwischen den verschiedenen Informationssystemen der RGW-Länder geschaffen.

Bei der Planung der verschiedenen AIDOS-Systeme wurde von den rechentechnischen Möglichkeiten der ESER-Technik ausgegangen. Angelehnt an die ESER-Betriebssystemversionen DOS, OS und VS wurden die Programmssysteme AIDOS/DOS (vgl. [We78]), AIDOS/OS (siehe [St78]) und AIDOS/VS ([Na85]) entwickelt und bereitgestellt. Dieser informationstechnologisch motivierte Ausbau von AIDOS berücksichtigte spezifische Möglichkeiten der Betriebssystemversionen, wie etwa Einzelnutzer- oder Mehrnutzerbetrieb, Band- oder Plattenspeicherorientierung, Stapel- oder Dialogbetrieb u. a. Vgl. hierzu auch [Ha85].

Zugleich erfolgte jedoch auch der Ausbau der Funktionalität, getrieben durch Anwenderforderungen, die Nutzergemeinschaften und vor allem das ZIID stellten. Dabei spielte der Übergang von der Stapelrecherche zur Dialogrecherche die augenscheinlichste Rolle, konnte doch ein Nutzer selbst in Echtzeit die Suche von relevanten Informationen vornehmen, ohne auf die Vermittlung der Kollegen der Informationsstelle angewiesen zu sein.

AIDOS/FORTE als Datenaustauschsystem mit komfortablen Mitteln zur Beschreibung unterschiedlichster Datenformate und den für den Austausch erforderlichen Umwandlungsregeln (siehe hierzu [St83]) sowie AIDOS/AID als Programmpaket zur automatischen Indexierung von Dokumenten ergänzten die Leistungsfähigkeit von AIDOS für spezielle Aufgaben.

Mit Bereitstellung des ersten DDR-Bürocomputers durch Robotron erweiterte AIDOS/M das Spektrum der AIDOS-Systeme. Damit wurde nicht nur die Möglichkeit geschaffen, auf dezentralisierten Arbeitsplätzen Daten für das zentrale AIDOS-Informationssystem zu erfassen oder für den Endnutzer aufzubereiten, sondern es gelang mit dem Ausbau der technischen Möglichkeiten, die Hauptfunktionalitäten der ESER-Rechner-AIDOS-Systeme auf dem Personalcomputer zur Verfügung zu stellen und damit die Voraussetzungen für die verteilte Verarbeitung von Informationen zu schaffen.

Die AIDOS/M-Systeme gewannen in Laufe der Zeit immer mehr an Bedeutung. Die Vorverlagerung der Erfassung und Plausibilisierung der Daten für Großrechner auf PC-Systeme brachte hohe Einsparpotentiale mit sich und stellte zum Zeitpunkt der Einführung eine bemerkenswerte Innovation dar. Uns ist kein Recherchesystem bekannt, das zum damaligen Zeitpunkt etwas Ähnliches zu bieten hatte.

2 Das Konzept des IRS AIDOS

Die grundlegenden Konzepte des IRS AIDOS sind im Beitrag von Harald Löhr dargestellt. Verweise auf verschiedenartige Informationsquellen (Bücher, Zeitschriftenartikel, Forschungsberichte, Patente, Gesetze, Prospekte aber auch Produktbeschreibungen in Text- und Tabellenform u. ä.), Nachweise genannt, können in einem Recherchesystem wie AIDOS, gesteuert von Erfassungsvorschriften, erfasst, auf der Basis verschiedenster Ordnungsmittel (Thesauri, Klassifikationen, Rubrikatoren, Faktenprofilen) geordnet so gespeichert werden, dass bei einer Recherche durch die Angabe von Suchmerkmalen (Deskriptoren, Indizes, Fakten) unter Nutzung logischer Operatoren und der semantischen Möglichkeiten der Ordnungsmittel relevante Mengen von Nachweisen gefunden und nutzergerecht aufbereitet werden können. Die Qualität eines Recherchesystems wird dabei unter anderem dadurch bestimmt, wie genau und vollständig eine Anfrage beantwortet wird, wie komfortabel mit dem System gearbeitet werden kann und wie operativ es auf die sich im iterativen Rechercheprozess ändernden Anfragen reagieren kann.

Das Konzept von AIDOS wird hier am Beispiel von AIDOS/M dargestellt ([Ha87]).

2.1 Anforderungen an die AIDOS-Datenbank

AIDOS-Systeme waren ursprünglich für die wissenschaftliche Informationstätigkeit konzipiert worden. Daraus leiteten sich auch die Anforderungen an die Datenbasis ab:

- thematische Trennung der Inhalte,
- stark schwankende Datenbelegung,
- variable Längen der Feldinhalte,
- unterschiedliche Anzahl von Werten in den Feldern,
- leistungsfähige Datenprüfung während der Erfassung,
- Ablagemöglichkeit von Texten,
- strukturierte Wörterbücher (Thesauri) zur Zugriffsunterstützung,
- Klassifikationen zur Zugriffsunterstützung,
- operative Suche nach Informationen.

Unformatierte Daten, Texte und mehrere Werte in einem Datenfeld, wie von dem Anwendungsgebiet gefordert, wurden damals weder durch relationale Datenbanken, wie etwa dBase, noch von den verfügbaren klassischen Datenbanksysteme (z. B. nach dem CODASYL-Konzept) unterstützt. Weiterhin gab es keine direkte Unterstützung von Wörterbüchern zur Dateneingabe. Zum Wiederauffinden, d. h. zum Navigieren durch die Baum- bzw. Netzstruktur, musste der Benutzer eine spezielle Abfragesprache verwenden. Die Formulierung solcher Anfragen erforderte Spezialkenntnisse, die in dem Anwendungsgebiet von AIDOS nicht vorausgesetzt werden können. Aus diesen Gründen wurde für die AIDOS–Datenbank eine eigene Datenstruktur entwickelt, die einen Kompromiss aus den dargestellten Anforderungen und den verfügbaren Speichermedien darstellte.

2.2 Architektur der AIDOS-Datenbank

Eine AIDOS-Datenbasis besteht aus folgenden Grundbestandteilen:

- Datenbeschreibung
  In ihr wird der Aufbau der Informationen mit ihren Merkmalen und Eigenschaften festgelegt und vorgegeben, mit welchen Wörterbüchern (Thesauri) und Klassifikationen in einer Datenbasis gearbeitet werden soll.

- Wörterbücher und Klassifikationen
  Sie bilden zusammen mit der inversen Liste einen Index über die gespeicherten Informationen. Dieser ermöglicht kurze Zugriffszeiten bei einer operativen Suche nach den gewünschten Informationen.

- Nachweise- und Textspeicher
  In ihnen sind die eigentlichen Informationen abgelegt.

- Rechercheergebnisspeicher
  Darin werden die Suchergebnisse abgelegt und können dort weiter selektiert bzw. ausgegeben werden. So kann eine Recherche in mehreren Stufen durchgeführt werden (stepwise refinement). AIDOS unterstützt auch das Durchsuchen der abgelegten Texte, (z. B. Abstracts). Die Vorselektion über Thesaurus- bzw. Klassifikationsmerkmale und die anschließende Präzisierung des Ergebnisses mittels Textbrowsing bringt einen wesentlichen Effektivitätsgewinn gegenüber der Textsuche durch den gesamten Datenbestand.

2.3 Datenbeschreibung

In der Datenbeschreibung sind die zu speichernden Nachweisearten (unterschiedlicher Datensatzaufbau) mit ihren Merkmalen (Feldern) und wiederum deren Eigenschaften beschrieben. Eigenschaften eines Merkmales sind z.B.:

- Datentyp (Zeichenkette, ganze Zahl, reelle Zahl oder Datum),
- Sprachtyp (Thesaurus-, Klassifikationsmerkmal, Fakt oder Text),
- Gültigkeitsbereich (Angaben zur Datenprüfung),
- Verarbeitungstyp (automatische Aufnahme in den Thesaurus),
- Wertanzahl (minimale und maximale Anzahl der Werte).

2.4 Klassifikation

Klassifikationen sind ein wichtiges Werkzeug zum schnellen und gezielten Wiederauffinden. Sie bildeten bei AIDOS eine wichtige Voraussetzung zur Anwendung in den wissenschaftlich/technische Informationsstellen der Betriebe. Eine Klassifikation besteht aus hierarchisch angeordneten Notationen und Sachgebietsbezeichnungen.

2.5 Wörterbuch (Thesaurus)

Aufbau und Funktion in AIDOS

Wir unterscheiden generell zwei Arten von Wörterbüchern:

- Wörterbücher mit vordefiniert Beziehungen zwischen den normierten Stichworten (Thesaurus) und
- einfache Wortsammlungen.

Wörterbücher (bzw. Thesauri) waren eines der bedeutendsten Merkmale von AIDOS. Mehrere Wörterbücher konnten in einer Datenbasis angelegt werden. Wörterbücher in Form einfacher Wortsammlungen konnten auch beim Laden von Nachweisen automatisch gefüllt werden. Derartige Wortsammlungen sind heute in vielen WEB 2.0-Systemen zu finden. Für einen Thesaurus standen mehrere Möglichkeiten zur Verfügung, um die inhaltliche Beziehungen zwischen den normierten Stichworten darzustellen:

- begriffliche Unterordnung (asymmetrische Beziehung),
- Synonymität (asymmetrisch),
- Teil-Ganzes-Beziehung (symmetrische Beziehung),
- assoziative Beziehung (asymmetrische Beziehung).

Beispiel für eine Struktur zum Stichwort „Computer“ :

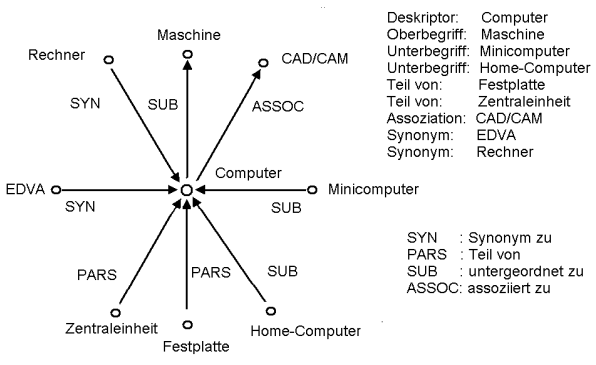


Abbildung 1: AIDOS-Thesaurus-Struktur

Implementierung des Thesaurus in AIDOS

Bei der Implementierung des Thesaurus von AIDOS waren die folgenden Anforderungen zu beachten:

- geringe Kapazität der Speichermedien,
- effektiver Zugriff,
- keine Reorganisation des Speichers,
- effektives Ablegen und Verarbeiten von Daten variabler Länge,
- Möglichkeit des direkten Zugriffs und sequentiellen Lesens in logischer Reihenfolge,
- Wiederverwendung frei gewordenen Speicherplatzes,

- Verarbeitung von bis zu 60 Zeichen langen Stichworten.

Gelöst wurden die Probleme mittels einer modifizierten Variante des bei relationalen Systemen häufig angewendeten TID-Konzeptes (TID: **T**uple **Id**entifier). Dieses Konzept gestattet eine einfache Verwaltung der sich bei der Arbeit ständig verändernden Satzlänge. Für die Verwaltung des Index bot sich eine vereinfachte B*-Baumorganisation an. Die Daten und der Zugriffsbaum wurden dabei in getrennten Dateien abgelegt, was die Implementierung vereinfachte und kürzere Zugriffszeiten gestattete.

Die Stichworte wurden prinzipiell auf 6 Zeichen verdichtet (Hashcode), der Hashcode mit dem Verweis auf den eigentlichen Datensatz erweitert. Ein Havariemanagement sorgte für die Auflösung auftretender Konflikte bei der Hashcodebildung.

Diese Organisation brachte noch folgende Vorteile:

- Der B*-Baum erfordert keine Reorganisation.
- Die Verweise im Schlüsselbaum sind unabhängig von der Lage des Satzes innerhalb der Datendatei. Das vereinfacht die Verwaltung von Sätzen variabler Länge und erhöht die Effektivität.
- Sortiertes sequentielles Lesen ist möglich (Verkettung der Blattknoten des B*-Baumes über den Wurzelknoten).
- Beim physischen Löschen freigewordener Speicherplatz wird wieder genutzt, was auf Grund der verfügbaren Datenträgerkapazität unbedingt erforderlich war.

3 AIDOS-Technologien zur Inhaltsverarbeitung

Es wurde bereits eingehend auf die Besonderheit des AIDOS eingegangen, die Wiederauffindung von Informationsnachweisen und Informationen unmittelbar mit hoher Vollständigkeit und Genauigkeit zu ermöglichen. Die Qualität der Recherche in einem konkreten Informationsspeicher hängt dabei in entscheidendem Maße davon ab, wie zutreffend eine Quellinformation in dem entsprechenden Nachweis beschrieben, anders gesprochen, wie genau das Dokument indexiert, verschlagwortet und/oder kategorisiert wird. Ein weiterer wesentlicher Faktor ist die Qualität der benutzten Ordnungsmittel. Die Verfahrenstechnologie des AIDOS zeichnete sich gerade durch die Nutzung der verschiedenen Ordnungsmittel aus, wobei die Thesaurusunterstützung eine herausragende Rolle spielte.

Die Eigenschaft des Systems, in der Recherche neben logischen Verknüpfungen von Suchbegriffen auch deren semantische Beziehungen im Thesaurus nutzen zu können, kann als Vorlauf zu aktuellen Verfahren der semantischen Suche im World Wide Web betrachtet werden. In diesem Zusammenhang ist auch die Erweiterung der Funktionalität des AIDOS von einem System zur Speicherung von Informationsnachweisen (Sekundärinformation) hin zu einem System der Speicherung und Wiederauffindung von Primärinformationen, in AIDOS Fakten genannt, zu betrachten

Die AIDOS-Faktenrecherche stellt eine weitere Technologie zur inhaltsgestützten Wiederauffindung von Daten dar. Über so genannte Faktenprofile wurde so zum Beispiel die Erfassung von Wirtschaftdaten oder technischen Parametern möglich, die unmittelbar recherchiert werden konnten. Die Prinzipien zur Faktenstrukturierung, -speicherung und –recherche sind später in Content-Management-Systemen ausgebaut und vervollkommnet worden.

Die bisher dargestellten Maßnahmen zur Sicherung einer hohen Qualität der Informationsrecherche sind verbunden mit der Notwendigkeit der inhaltlichen Analyse von Informationen, die in natürlichsprachigen Texten vorliegen. Da ein Anwender sein Informationsbedürfnis primär auch in natürlicher Sprache formuliert, liegt die Idee nahe, auch die Anfrage an ein IRS in natürlicher Sprache formulieren zu können und somit nicht nur Mittel zur Inhaltserschließung natürlichsprachiger Dokumententexte sondern auch der Anfragen bereitzustellen.

3.1 Automatisches Indexieren

Für die automatisierte Inhaltserschließung wurde im Rahmen von AIDOS das Programmpaket zur **A**utomatischen **I**ndexierung von **D**okumenten (AID) im Rahmen einer Vorlaufarbeit entwickelt. Das geschah in enger Zusammenarbeit mit dem IZWTI in Moskau, wo auch umfangreiche Tests zur Qualität des zugrunde liegenden Indexierverfahrens durchgeführt wurden. Vergleiche hierzu [IZ82].

Ausgehend von Referatetexten und Dokumententiteln werden inhaltsbeschreibende Wortgruppen ermittelt. Stoplisten dienen zum Ausschluss von Banalwörtern. Suffixlisten ermöglichen die Wortformanalyse zur Reduzierung von Flexionsformen. Präpositionslisten gestatten im Zusammenhang damit die Ermittlung von Wortgruppen.

Die gefundenen inhaltstragenden Wortgruppen werden schließlich auf die Deskriptoren des AIDOS-Thesaurus abgebildet. Logisch zusammengehörige Deskriptoren werden in entsprechenden Ketten im resultierenden Dokumentenprofil angeordnet.

Das für ESER-Rechner konzipierte AIDOS-Indexierverfahren wurde in den 1980-er Jahren auch für AIDOS/M angepasst.

Dem AIDOS-Indexierverfahren analoge Technologien zur inhaltlichen Texterschließung haben sich im Rahmen der Web-Suche erst in den letzten Jahren durchgesetzt.

3.2 Das natürlichsprachige Interface NLI-AIDOS

Wie oben dargestellt, war das IRS AIDOS für die Speicherung/Pflege von und zur Recherche in großen Nachweismengen konzipiert, die sich auf verschiedenartige textbasierte Dokumente, wie zum Beispiel Gesetzessammlungen, Kommunalvorschriften, Bauverordnungen, Veröffentlichungskataloge, Verzeichnisse von Kunstobjekten usw. bezogen. Die Recherche nach Merkmalen wie Titel, Datum,

Standort, Urheber, … ließ sich hier mit der AIDOS-Recherchesprache gut und sicher bewerkstelligen.

Es bestand daneben das Ziel, auch Nutzern ohne jede Vorkenntnisse vom System eine Recherche zu ermöglichen.

Die einzige Ausdrucksform, mit der ein naiver Nutzer sein Informationsbedürfnis artikulieren kann, ist der natürliche Sprachgebrauch. Diese Anfragen können beispielsweise von der Art sein wie „Gibt es ein …?“, „Welcher … hat … verfasst?“, „Seit wann liegen … vor?“, „Wie viele … über …. sind seit …veröffentlicht worden?“, „Wo finde ich …, wobei … aber nicht …?“ usw. Und die mit „….“ bezeichneten Leerstellen konnten an den entsprechenden Stellen mit einfachen Wörtern oder auch paraphrastischen Umschreibungen besetzt sein, und man kann nicht einmal immer von einer zwingend korrekten orthographischen Schreibweise ausgehen. Es war also eine Möglichkeit zu finden, solche frei formulierten Anfragen in deutscher Sprache zu Datenbeständen im AIDOS-Format zuzulassen, zu „verstehen“, die Recherche im Datenspeicher daraufhin zu initiieren und eine Antwort zurückzuliefern. Ein Softwarepaket mit diesen Eigenschaften nennt man natürlichsprachiges Interface (NLI).

Auf der Basis von Robotron Forschungsarbeiten zum Gegenstand der Frage-Antwort-Systeme FAS-75 und FAS-80 aus dem Forschungsumfeld der *Künstlichen Intelligenz* (KI), *Computerlinguistik* (CL) bzw. zu *Informationslogischen Systemen*, wurde seit Spätherbst 1980 durch die Mitarbeiter der Forschungseinheit und AIDOS-Entwicklern ein Konzept entwickelt, wie diese NLI-Schnittstelle zu AIDOS zu implementieren sei (vgl. hierzu auch [Zä08], Kap. 5).

Es wurde vorerst von einer schriftsprachlichen (*graphemischen*) Anfrage, also per *Tastatur*, ausgegangen. Die Eingabe wurde von einem linguistischen Prozessor bearbeitet, der eine Wortformenerkennung und Reduzierung auf die Grundformen vornahm (*Lemmatisierung*).

Dann wurde die Wortfolge im Satz einer syntaktisch-semantischen Analyse unterzogen, um die Bedeutung der Anfrage zu ermitteln. Für diese Analyse wurde ein wortklassengesteuertes Verfahren (WCFA, [He86]) eingesetzt.

Die semantische Repräsentation (*Tiefenstruktur* der Anfrage) erfolgte in Ausdrücken einer semantischen Zwischensprache SZS. Die Bedeutungsdarstellung der Sprachausdrücke als SZS-Ausdruck wurde dann mit dem Transformationsmodul in formale Ausdrücke der AIDOS-Recherchesprache mit Operationen wie SEARCH, COUNT, DISPLAY usw. und booleschen Operatoren. übersetzt, mit der danach die Informationsfindung im eigentlichen AIDOS-Sachdatenspeicher ausgelöst wurde. Eine Sprachsynthese zur Ausgabe eines grammatisch vollständigen Antwortsatzes/-textes war hier nicht erforderlich, weil das Rechercheergebnis aus AIDOS heraus schon allein hinreichend verständlich war.

Der Prozess der Analyse, Bedeutungsermittlung und Transformation wurde durch ein NLI-Lexikon, aber auch durch das AIDOS-„Wissen“ in Form des Datenmodells, der Klassifikation und des Thesaurus unterstützt. Diese Struktur war insofern

allgemeingültig und flexibel mehrfach nutzbar, als für wechselnde Datenbankinhalte oder die Kombination mit anderen Informationsrecherchesystemen als AIDOS nur der Transformationsmodul und gewisse Einträge im NLI-Lexikon zu modifizieren waren. Einzelheiten dazu sind bei HELBIG [He91, S 264 ff.] dargestellt.

Die Arbeiten am NLI-AIDOS wurden im Verlaufe des Jahres 1990 eingestellt. Zu dieser Zeit war das System konzeptionell zu Ende durchdacht und experimentell erprobt, es wäre durch eine intensive Programmierung vielleicht nach einem Jahr vollständig praktisch arbeitsfähig gewesen. Das ist umso beachtlicher, als man bis zum heutigen Tag von einem breiten Einsatz praxistauglicher Systeme dieser Art nicht sprechen kann.

4 Softwaretechnologische Aspekte der AIDOS-Entwicklung

Die Entwicklung des AIDOS ordnete sich in die Software-Vorgaben der ESER-Technologie ein, die an die IBM-Technologie angelehnt war.

Die Versionen AIDOS/DOS und AIDOS/OS basierten auf der ESER-Entwicklungstechnologie für betriebssystemnahe Software. Der Softwareentwurf erfolgte auf der Basis von Programmablaufplänen und Struktogrammen. In der sich anschließenden Kodierungsphase wurden die Assemblersprache und entsprechende Makroerweiterungen genutzt. Die Makroerweiterungen vereinfachten die Nutzung von Geräteressourcen sowie die Behandlung von AIDOS-spezifischen Bestandteilen der Datenbasis (z.B. Thesaurus). Spezielle Makroerweiterungen betrafen die Einführung der strukturierten Programmierung als Reaktion auf die doch erhebliche Fehlerbehaftung des Kodierungsprozesses. Ein wesentlicher Grund für die Nutzung der Assemblersprache war der verfügbare Speicher. AIDOS/DOS sollte zum Beispiel in einem Speicher von 64 KByte (R21) lauffähig sein. Dafür kam nur Assembler als Entwicklungsmittel in Frage.

Der Testprozess beruhte auf Testprogrammen, die mit der gleichen Technologie erstellt wurden. Zur Beschreibung von Eingabeformularen, Reports und zur Behandlung von Abfragen an die Datenbank und des Datenaustauschs zwischen verschiedenen Informationssystemen wurden Spezialsprachen unterstützt, für die eine Reihe von Compilern entwickelt wurde. Durch die Nutzung der Compilertechnologie konnte die Stabilität der Nutzerschnittstellen sowie die Entwicklungsproduktivität verbessert werden.

Bei der AIDOS/VS-Entwicklung wurde die dargestellte Entwurfs- und Entwicklungstechnologie durch objektorientierte Methoden ergänzt. Auf der Basis von Makroerweiterungen wurde das Konzept von Datenkapseln eingeführt, die in einem Repository ein Repertoire von Funktionen anboten, die im Prozess ihrer Nutzung verwendet werden konnten. Der bisherige Zugriff auf die internen Datenrepräsentationen wurde damit unterbunden. Mit der objektorientierten Entwicklungstechnologie des AIDOS/VS wurde damit bei Robotron in den 80-er Jahren praktischer ein Vorgriff auf heute gängige Softwaretechnologien unternommen.

Parallel zur AIDOS/VS-Entwicklung wurde mit der Unterstützung von Personalcomputern begonnen. Die Entwicklungstechnologie für das AIDOS/M unterschied sich von der Entwicklung der ESER-Programmsysteme, indem hier von Anfang an in der Hochsprache PASCAL programmiert wurde. Das ermöglichte die schnelle Anpassung des AIDOS/M an verschiedene Betriebssysteme wie CP/M und später MSDOS.

Für die AIDOS-Systeme war für alle Komponenten der Datenbank (Datenbeschreibung, Thesaurus, Klassifikation und Nachweise) ein spezielles Datenaustauschformat definiert worden. Diese Datenaustauschdateien ermöglichten neben der Datensicherung auch die Offline-Übertragung der Daten zwischen den verschiedenen AIDOS-Systemen. Die Dateien erfüllten bzgl. des verwendeten Codes die Bedingungen der unterschiedlichen Plattformen (ISO 7-bit Code), sodass auch eine Online-Übertragung der Dateien möglich gewesen wäre. In AIDOS wurden dafür spezielle Funktionen zum Laden und Entladen der Daten implementiert. Damit war die Erfassung von Informationen für die Großrechnervarianten von AIDOS mit AIDOS/M möglich. Durch Übernahme von Daten anderer AIDOS-Systeme wurde u. a. auch die weitere Auswertung von Recherche-ergebnissen durch AIDOS/M unterstützt.

5 Nachwirkungen der AIDOS-Entwicklung auf Folgeprojekte in den 90-er Jahren

Die Wartung, der Vertrieb und die Weiterentwicklung wurden ab Juli 1990 von der neu gegründeten „Data-Print Robotron System GmbH“ übernommen. In relativ kurzer Zeit war das Informationsrecherchesystem AIDOS/M in drei verschiedenen Ausbaustufen verfügbar.

- Informationsrecherchesystem AIDOS/M zum Speichern und Wiederauffinden von Textdaten
- Bilddatenbank AIDOS/M zum Speichern und Wiederauffinden von Bild- und Textdaten
- Bildrecherchesystem AIDOS/M–Sonografie zum Speichern und Wiederauffinden von Patientendaten und Befunden in der Ultraschalldiagnostik

Die Speicherung der Bilddaten war analog über ein auf Laseroptik basierenden **C**omponent **R**ecording **V**ideo System (CRV-System) oder digital z. B. auf einem WORM (**W**rite **O**nce – **R**ead **M**ultiple)-Laufwerk möglich. Die Digitalisierung und die Anzeige der Ergebnisse erfolgten in der damals üblichen Overlay-Technik mit spezieller Hardware. Bei der Anzeige der Nachweise zeigte AIDOS die zugehörigen Digitalbilder direkt an bzw. veranlasste das CRV-System zur Darstellung der zugehörigen Bilder.

In der zweiten Hälfte der neunziger Jahre begannen die Entwicklungen zur Ablösung des traditionellen Dateisystems von AIDOS/M, zuerst von der Data-Print GmbH Rechnersysteme, dann von der „M. C. Consult & Projektentwicklungs GmbH“ und zuletzt bis 2007 von der AIDOS AG. Ziel war es, einen SQL-Server, der heute

bekanntlich in der Lage ist, auch mehrere Werte pro Datenfeld, Texte und binäre Daten zu verwalten, zur Ablage der Daten zu nutzen. Implementiert wurden wieder die Möglichkeiten und die Leistungsfähigkeit des strukturierten Thesaurus. Durch diese Umstrukturierung wurde es unter AIDOS/M möglich, wesentlich größere Datenmengen zu verwalten und problemlos von verschiedenen Orten innerhalb eines Netzes zuzugreifen. Damit wurden die wesentlichsten Anforderungen moderner Contentmanagement-Systeme erfüllt. Erst unter den Bedingungen des WEB 2.0 werden nicht strukturierte Thesauri mit nicht normierten Schlüsselworten genutzt, um Informationsinhalte zu beschreiben. Normierte Stichworte und Hierarchien sind unter den Bedingungen des WEB 2.0 („jeder soll mitmachen“) nicht realisierbar. Die schnelle Wiederauffindung zusammengehöriger Daten, die durch natürlichsprachige Stichworte indexiert sind, wird dann zur Aufgabe der Suchalgorithmen des Systems.

Literaturverzeichnis

[Ha85] Hartmann, K., Naumann, P.: Informationsrecherchesystem AIDOS/VS, Neue Technik im Büro 29(1985)4

[Ha87] Hartmann, H.-D.: AIDOS/M16-Informationsrecherchesystem, rechentechnik datenverarbeitung 24(1987)3

[He86] Helbig, H.: Syntactic-semantic analysis of natural language by a new word-class controlled functional analysis. In „Computers and AI“, 5 (1986), S. 53-59. Bratislava, 1986

[He91] Helbig, H.: Künstliche Intelligenz und automatische Wissensverarbeitung. Verlag Technik, Berlin, 1991

[IZ82] Internationales Zentrum für Wissenschaftliche und Technische Informationen, Methodische Materialien und Dokumentationen zu Anwendungssoftwarepaketen – Automatische Indexierung auf des Basis des AIDOS, Teil 1 – Anwendungsbeschreibung (in russischer Sprache)

[Na85] Naumann, P.; Wenzel, R.-D. : Das Informationsrecherchesystem AIDOS/VS. - In: 14. Kolloquium über Information und Dokumentation. Nutzung von Informationstechnologien zur Beschleunigung des wissenschaftlich-technischen Fortschritts und zur Durchsetzung der ökonomischen Strategie. Themenkreis 2a . - Ilmenau : Technische Hochschule Ilmenau, Inst. für Informationswissenschaft, Erfindungswesen und Recht (INER) , 1985 , Seiten 83-89

[St78] Stys, Th.: Die Anwendung des Programmiersystems AIDOS OS/ES in der wissenschaftlich-technischen Information der DDR. - In: Informatik 25(1978)4, S. 34-38. Berlin 1978

[St83] Stys, Th.; Nagel, K.; Wenzel, R.-D.: PP AIDOS-FORTE: Ein universelles Formattransformationssystem für den Datenaustausch. In: Informatik 30(1983)5, S. 34-36. Berlin 1983

[We78] Wenzel, R.-D.: Erfahrungen und Tendenzen beim Einsatz von AIDOS. - In: Beispiel der POS-Nutzung im DOS/ES zur Vorbereitung des OS/ES, 27 S. Verlag Wirtschaft, Berlin 1978

[Zä08] Zänker, F.: Künstliche Intelligenz und automatische Sprachverarbeitung – ausgewählte Forschungen im Kombinat ROBOTRON. – In: diesem Band.

50 Jahre „Computer Aided Engineering (CAE)“ von DDR-Wurzeln bis Airbus

Werner Gumpert

Ingenieurbüro IBUL
Weißbacher Straße 51b
09235 Burkhardtsdorf
gumpert.werner@vdi.de

Abstract: Einsatz der „Oprema“ von Zeiß Jena als erstem Digitalrechner der DDR und des EARI, Elektronischer Analogrechner Ilmenau, in Verbindung mit solider Diplomausbildung in Angewandter Mathematik stützten den Beginn einer neuen Ära ingenieurtechnischer Arbeit auch im Automobilbau der DDR. Folgende Hardwareentwicklungen bzw. Importe aus der Sowjetunion ließen schnell erste Erfolge mit Vermittlung durch die Kammer der Technik ausbauen und auf den gesamten Maschinenbau verallgemeinern. Das an der TH Karl-Marx-Stadt entwickelte Programmpaket AIDAM zur automatischen Integration der Bewegungsdifferentialgleichungen allgemeiner Mechanismen diente aus den siebziger Jahren bis zum Ende der DDR für zahlreiche Anwendungen in verschiedensten Zentren von Automobilbau über Landmaschinenindustrie bis zum Schiffbau. Zu vielen CAE-Tools weiterer Entwicklungsstellen wurden durch einen „Programmkatalog Maschinendynamik“ und eine spezielle Rubrik in der Zeitschrift „Maschinenbautechnik“ Nachnutzungen organisiert.

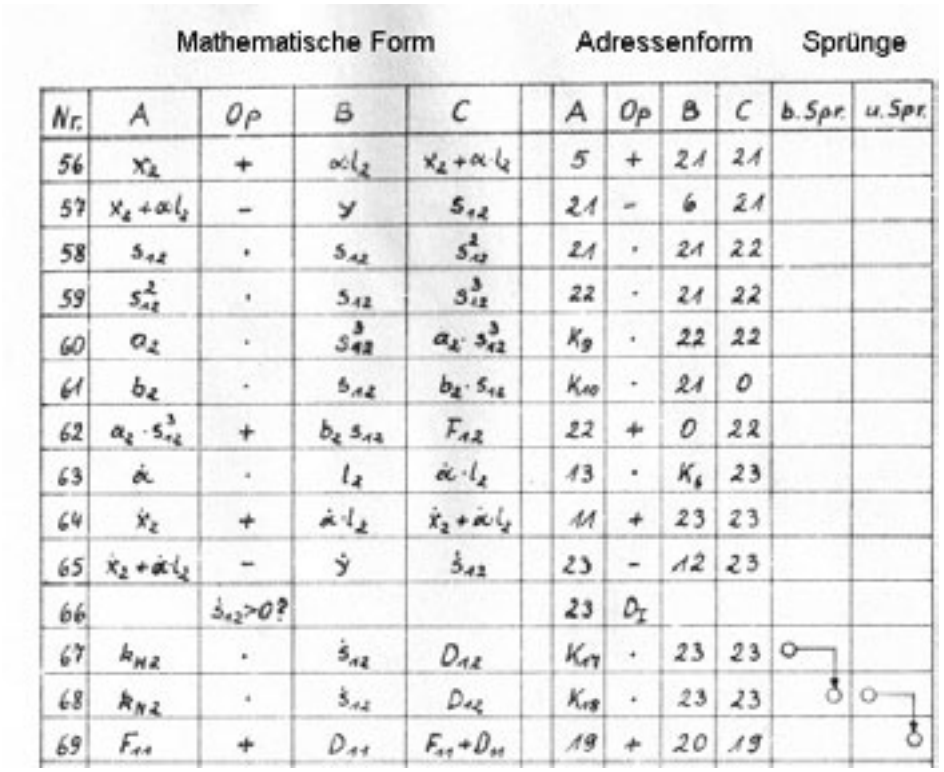
Modernste Tools mit heutigen Hardwareleistungen und Bedienerkomfort übertreffen alle Ahnungen der Anfangszeit. Das Know-how der Spezialisten wird damit nicht überflüssig, sondern kann sogar bei Großkonzernen wie z.B. Airbus noch wichtiger werden.

1 Der Anfang im Automobilbau

Der Start der DDR 1949 war auch im Kraftfahrzeugbau zunächst mit einem weiteren Niedergang durch Abwanderung der Spitzenkräfte gen Westen verbunden. Im Entwicklungszentrum VEB ZEK für den Kraftfahrzeugbau Karl-Marx-Stadt war noch 1956 der Einsatz mathematischer Methoden geradezu verpönt. Durch Einzelinitiativen und die aufkommenden Potenzen für Berechnungen änderte sich das Bild jedoch recht schnell. Mit der Neustrukturierung des wissenschaftlichen Potentials 1963 im VEB WTZ Automobilbau wurde sogar eine Abteilung Mathematik geschaffen und über rein rechentechnische Aufgabenstellungen hinausgegangen.

Erste einfache Schwingungsrechnungen wurden ab 1958 mit numerischer Integration am ersten programmgesteuerten Ziffernrechner der DDR, der „Oprema“ von Zeiß Jena, und dem Analogrechner EARI in Ilmenau ausgeführt.

Für die Oprema wurden als Hauptelemente 17000 Relais und 90000 Selengleichrichter verbaut. Die Befehlsliste umfasste 32 Dreiadressbefehle. Die Rechenzeiten betrugen für Addition 120ms, für Multiplikation/Division 800ms und für Radizieren 1200ms. Abbildung 1 zeigt einen Ausschnitt aus einem Rechenprogramm der Diplomarbeit des Verfassers.



	Mathematische Form				Adressenform				Sprünge	
Nr.	A	Op	B	C	A	Op	B	C	b. Spr.	u. Spr.
56	x_2	+	$\alpha \cdot l_2$	$x_2 + \alpha \cdot l_2$	5	+	21	21		
57	$x_2 + \alpha l_2$	−	y	s_{12}	21	−	6	21		
58	s_{12}	·	s_{12}	s_{12}^2	21	·	21	22		
59	s_{12}^2	·	s_{12}	s_{12}^3	22	·	21	22		
60	a_2	·	s_{12}^3	$a_2 \cdot s_{12}^3$	K_9	·	22	22		
61	b_2	·	s_{12}	$b_2 \cdot s_{12}$	K_{10}	·	21	0		
62	$a_2 \cdot s_{12}^3$	+	$b_2 s_{12}$	F_{12}	22	+	0	22		
63	$\dot{\alpha}$	·	l_2	$\dot{\alpha} \cdot l_2$	13	·	K_6	23		
64	$\dot{x}_2$	+	$\dot{\alpha} \cdot l_2$	$\dot{x}_2 + \dot{\alpha} l_2$	11	+	23	23		
65	$\dot{x}_2 + \dot{\alpha} l_2$	−	$\dot{y}$	$\dot{s}_{12}$	23	−	12	23		
66		$\dot{s}_{12} > 0$?			23	O_I				
67	k_{N2}	·	$\dot{s}_{12}$	D_{12}	K_{17}	·	23	23	○	
68	k_{N2}	·	$\dot{s}_{12}$	D_{12}	K_{18}	·	23	23		○
69	F_{11}	+	D_{11}	$F_{11} + D_{11}$	19	+	20	19		

Abbildung 1: Ausschnitt aus einem Oprema-Programm

Bedingte und unbedingte Sprünge wurden mit Steckschnüren an einer Programmtafel realisiert. Zahlenresultate wurden von einer „Kassenrolle“ abgelesen und als Kurven in Millimeterpapier eingetragen. Diese genügten z.B., um eine Konstrukteursidee für einen neuartigen Stoßdämpfer zu beurteilen, dessen materieller Test die zigfachen Kosten verursacht hätte.

Gegenüber der numerischen Integration mit vielen Detailschritten am Ziffernrechner bot der Analogrechner mit dem Kondensator ein Bauelement, das unmittelbar für zeitliche Integration geeignet war. Da auch seine Handhabung als „Modellbaukasten“ dem Ingenieur entgegenkam, war er in der Anfangszeit der elektronischen Rechentechnik für viele Dynamik-Aufgaben dem Digitalrechner durchaus überlegen. Im VEB WTZ Automobilbau wurde deshalb 1963 ein sowjetischer Rechner MN-7 importiert, der mit 6 Integrierverstärkern, 4 Nichtlinearitätenblöcken und verschiedenen Diodeneinheiten für schnelles Repetieren ebenso wie für Langzeitrechnen ausgelegt war. Die Anschaffungskosten entsprachen damals etwa einem Farbfernseher.

Abbildung 2 zeigt den Analogrechner und Ergebniskurven erster Fahrstabilitätsberechnungen, mit denen der Einfluss der Achslastverteilung zu beurteilen war.

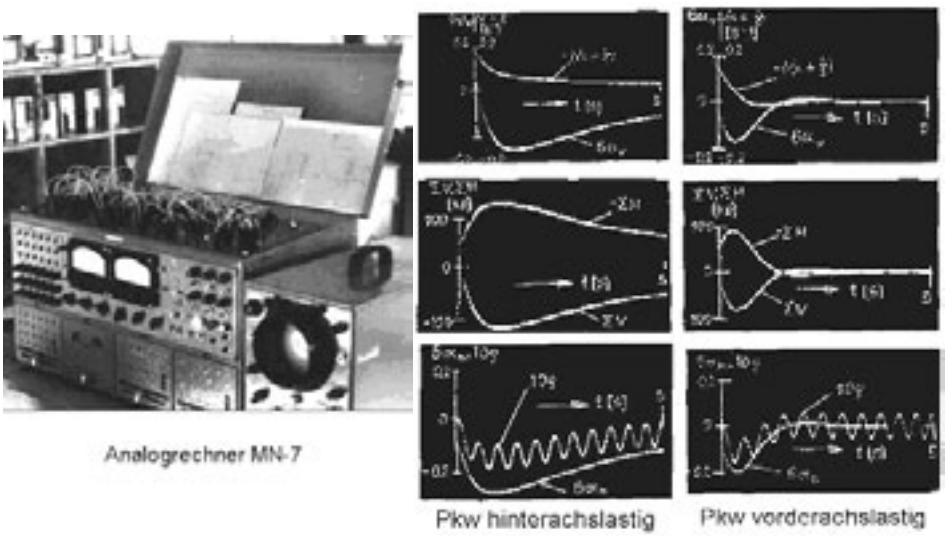


Abbildung 2: Rechner MN-7 und Ergebnisse einer Fahrstabilitätsberechnung

Die mögliche Verbindung des Analogrechners mit der allgemeinen Messtechnik ergab trotz der teilweise erforderlichen manuellen Bearbeitung große Vorteile gegenüber der digitalen Technik. Meines Wissens bietet die kommerziell verfügbare heutige Technik z.B. keine Hard- oder Software, die Oszillografen mit steuerbarer Papiergeschwindigkeit ersetzt. Wie Abbildung 3 zeigt, konnte damit in konzentrierter Form sowohl die Entwicklung von Amplituden als auch die Veränderung der Schwingformen dargestellt werden.

Nachdem in den Anfangsjahren von CAE vor Zahlenfluten gewarnt werden musste, wie das Zitat

„*The purpose of computing is insight, not numbers*“ (R.W. Hamming , 1964?)

belegt, so ist heute oft die Warnung vor einer bunten Grafik-Flut angesagt und nach kompakten Mitteln zu suchen, um die „Insight“ den Entscheidungsträgern zu vermitteln.

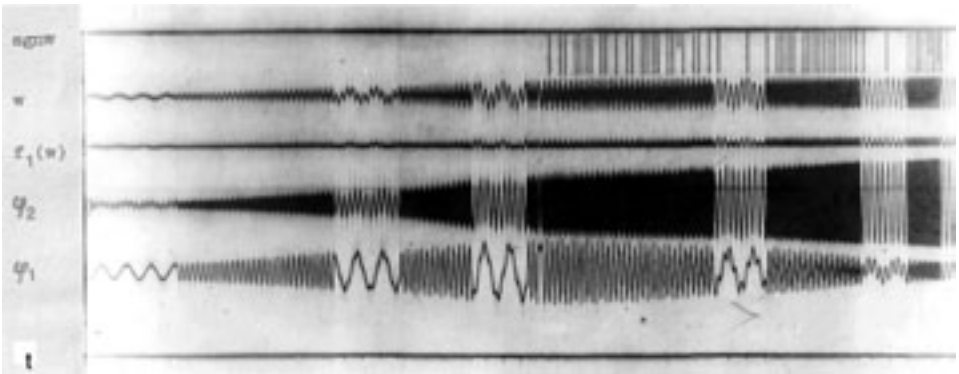


Abbildung 3: Oszillogramm mit variierter Papiergeschwindigkeit

2 CAE in der „Reife-Phase“ von etwa 1965 bis 1990

Ein vom Ministerrat der DDR im Juli 1964 beschlossenes „*Programm zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR in den Jahren 1964 bis 1970*“ umfasste auf 116 Seiten 45 Punkte, darunter auch zur „*Nutzung der Rechen- und Datenverarbeitungstechnik bei der naturwissenschaftlich-technischen Forschung und Entwicklung, der Projektierung, Information und Dokumentation*“. Trotz der üblichen Hemmnisse und Abstriche erleichterten diese Grundlagen zahlreiche Initiativen im allgemeinen Maschinenbau, die oft durch die KDT, das DDR-Gegenstück zum VDI, getragen wurden.

Für ein am 5.11.64 durch das WTZ Automobilbau veranstaltetes Kolloquium über „Untersuchungen der Nutzungsmöglichkeiten mathematischer und rechentechnischer Hilfsmittel ...“ reichten die betrieblichen Räume für die zahlreichen Interessenten des Industriezweigs und weiterer Bereiche nicht aus, und es musste der Stadtverordnetensaal des Karl-Marx-Städter Rathauses gemietet werden. Ein Tagesordnungspunkt „Aufgaben und Organisation des Rechenzentrums der Deutschen Reichsbahn“ vermittelte Impulse aus dieser relativ gut an internationale Bedingungen angeschlossenen Organisation.

Im Gegensatz zum heutigen Konstrukteursanteil an CAE wurden die ersten zwei Jahrzehnte vorwiegend aus den Berechnungsabteilungen gestaltet, anfangs mit überwiegender Mehrkörperdynamik, zunehmend dann mit Anwendungen der Finite-Elemente-Methode und allgemeinen Konstrukteurs- und Technologen-Aufgaben. Eine wesentliche Hürde für Anwendungen der Mehrkörperdynamik war (und ist) die Modell-Findung. Mit adäquaten Vereinfachungen können oft Ad-hoc-Modelle aufgestellt und für wertvolle Erkenntnisse berechnet werden. Die Beurteilung der Adäquatheit setzt aber Know-how voraus, das oft nicht ausreichend vorhanden ist. Deshalb wurden von entsprechenden Zentren aus praxisrelevante Aufgabenklassen definiert und für die Lösung mit speziellen Programmen aufbereitet.

Bereits 1976 war damit eine solche Vielzahl erreicht, dass Programme aus 20 Institutionen in einem Katalog zur Nachnutzung angeboten werden konnten. Abbildung 4 gibt die Übersicht hierzu wieder.

Abbildung 4: Seiten 2 und 3 des Programmkatalogs „Maschinendynamik“

Zehn Jahre später wurden auf Anregung von Prof. Holzweißig, TU Dresden, „Bausteine zur rechnergestützten Konstruktion“ in der Zeitschrift „Maschinenbautechnik“ zwecks Nachnutzung veröffentlicht: Abbildung 5.



Bausteine zur rechnergestützten Konstruktion

Koordiniert durch die Arbeitsgruppe „Rechenverfahren und Programme der Maschinendynamik" des KDT-FUA Maschinendynamik im FA TMM

Verantwortlich:
Doz. Dr. sc. techn. W. Gumpert
TH Karl-Marx-Stadt
9010 Karl-Marx-Stadt, PSF 964

Doz. Dr. sc. techn. W. GUMPERT (KDT); Dr.-Ing. E. PAUSCH (KDT)
Technische Hochschule Karl-Marx-Stadt, Sektion Maschinen-Bauelemente

Rechenprogramme der Maschinendynamik: Beiträge zur rechnergestützten Konstruktion [1]

0. Einleitung

Vor rund zwanzig Jahren wurde von den zuständigen staatlichen Stellen ein „Programm zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR in den Jahren 1964 bis 1970" ausgearbeitet und anschließend weitgehend in die Tat umgesetzt.

der Rechentechnik wie bei der Nutzergemeinschaft für den ZRA-1, den ersten in der DDR serienmäßig produzierten Rechenautomaten, als auch Gemeinsamkeit der Aufgaben wie im Arbeitskreis der Hochschul-Rechenzentren [2] oder einer Arbeitsgemeinschaft von Vertretern der wissenschaftlich-technischen Zentren aller Industriezweige des Allgemeinen Maschinenbaus, mit der insbesondere die ersten Schritte zur Anwendung von Rechenprogrammen für ingenieurtechnische Aufgaben beschleunigt wurden. Später wurde im

Abbildung 5: Kopf des ersten „Bausteins zur rechnergestützten Konstruktion"

Parallel zu dieser „Bedienung" spezieller Aufgabenklassen wurde als universelles Hilfsmittel zur Behandlung verschiedenster Ad-hoc-Modelle ab 1971 vom Verfasser an der TH Karl-Marx-Stadt das FORTRAN-Programm AIDAM (**A**utomatische **I**ntegration der Bewegungs**d**ifferentialgleichungen **a**llgemeiner **M**echanismen) entwickelt und später an vielen Industrie- und Forschungsstellen (einschließlich NAMI Moskau) bis zum Aufkommen entsprechender Software mit grafisch interaktiven Oberflächen genutzt.

Gegenüber numerischer Langzeitintegration bestanden teilweise sehr große Vorbehalte, die bis heute bedacht sein sollten, weil tatsächlich nicht alle berechneten Kurven auch dem technisch-physikalischen Problem entsprechen. So können sowohl stabile Systeme scheinbar instabil wie auch instabile Systeme scheinbar stabil sein. Für den ersten Fall ist die Numerik zuständig, für den zweiten die Modellierung. Zuweilen müssen winzige Störungen, die in der Realität die Instabilität auslösen, mit modelliert werden, um sich auf Integrationsergebnisse verlassen zu können.

Abbildung 6 zeigt eine Rechnung, bei der zu kurze Wortlänge Oberschwingungen entstehen ließ, die mit kleinerer Schrittweite noch krasser wurden, bei doppelter Wortlänge aber verschwanden. Gleichzeitig zeigt dieses Bild, wie damals Kurven mit Ziffernsymbolen gedruckt wurden.

Als Argument gegen die Zweifler an der Genauigkeit von Langzeitintegrationen konnte und kann der Kurvenverlauf bei Abbildung 7 dienen, die einen sich aufschwingenden Verlauf für ein eigentlich „bösartiges" Modell zeigt, einmal mit AIDAM und einmal mit dem tschechoslowakischen Analogrechner MEDA ermittelt. Die Übereinstimmung war selbst für die Bearbeiter erstaunlich genau.

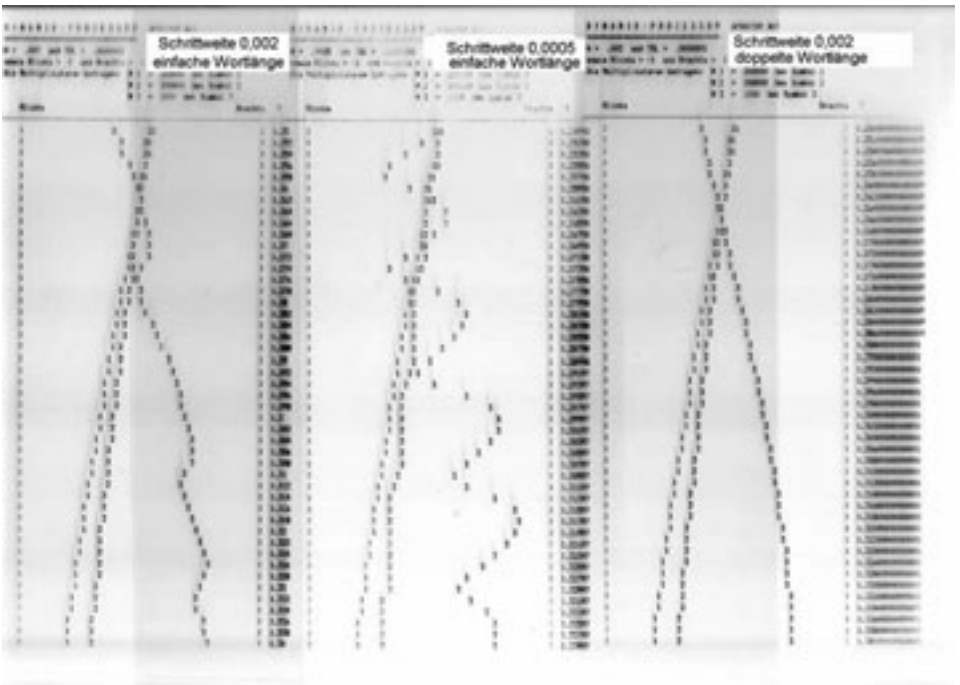


Abbildung 6: Einfluss von Integrationsschrittweite und Wortlänge

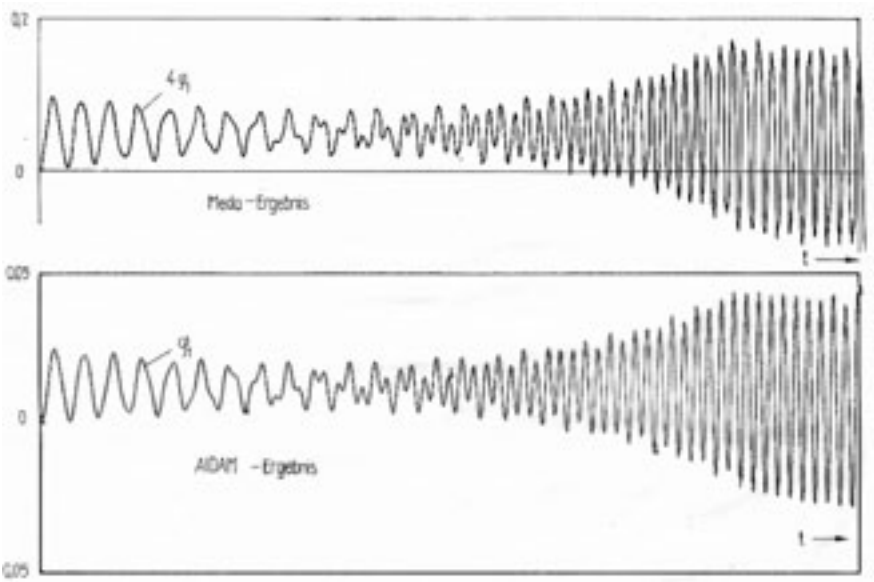


Abbildung 7: Vergleich einer Ratterschwingung; $4\varphi_1$ mit MEDA/analog bzw. φ_1 mit AIDAM/digital

Angesichts der geschilderten Schwierigkeiten waren Vorbehalte der Praxis gegen die Übernahme der neuen Techniken nur natürlich. Einen Beitrag dagegen konnten die ca. ab 1977 an mehreren Hochschulen gegründeten „Studentischen Rationalisierungs- und Konstruktionsbüros“ leisten. An der TH bzw. später TU Karl-Marx-Stadt an der Sektion Maschinen-Bauelemente wurden bei juristischer Verantwortung eines Hochschullehrers ca. 5 Objekte pro Jahr von Studenten mit regulärem Wirtschaftsvertrag bearbeitet. Die Mehrheit der Aufgaben bedeutete CAE.

3 CAE im „erwachsenen“ Stadium, nach 1990

Die mit der politischen „Wende” einhergehenden Prozesse trafen teilweise mit dem Erreichen einer neuen Qualität in der technischen Arbeit zusammen. So war einerseits mit dem Übergang auf die Länderstruktur z.B. die Auflösung eines der beiden Dynamik-Lehrstühle an der TU Chemnitz angemessen. Andererseits stellten sich mit dem massenhaften Einsatz leistungsfähiger PCs und zahlreicher CAE-Software mit grafisch interaktiver und damit nutzerfreundlicher Bedienung wesentlich umfassendere Potenzen für solide technische Arbeit zur Verfügung.

Der Inhalt von CAE lässt sich natürlich nicht an einem persönlichen Lebenslauf definieren und wird auch recht unterschiedlich interpretiert. Abbildung 8 deutet eine der verbreiteten Auffassungen an.

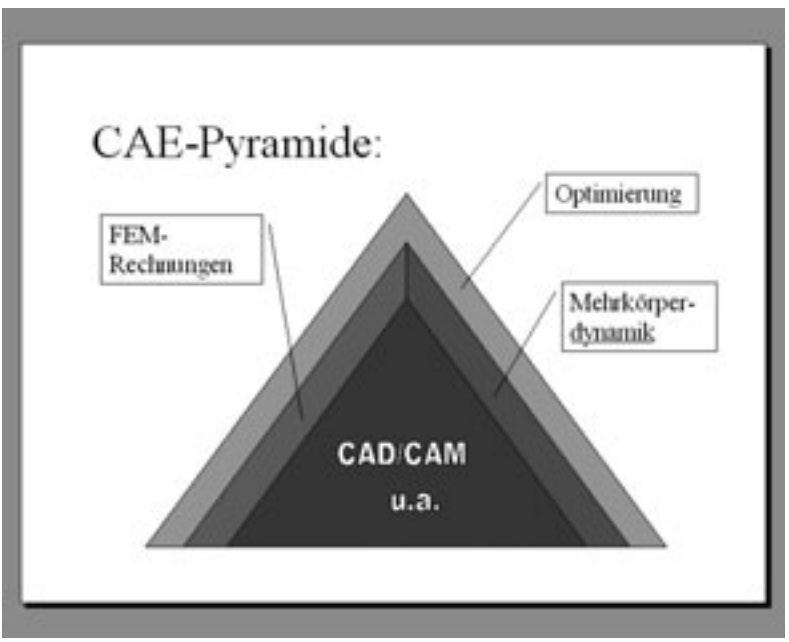


Abbildung 8: Eine Darstellung des CAE-Inhalts

Die Schicht der FEM-Rechnungen und der Mehrkörperdynamik hat im Prozess der Entwicklung mit CAD, CAM und zahlreichen weiteren Komponenten der Ingenieurarbeit eine breite und sehr produktive Basis erhalten. Wie weit die Integration reichen kann, zeigt eine bereits vor Jahren publizierte Vorstellung der amerikanischen Firma ALGOR, Abbildung 9:

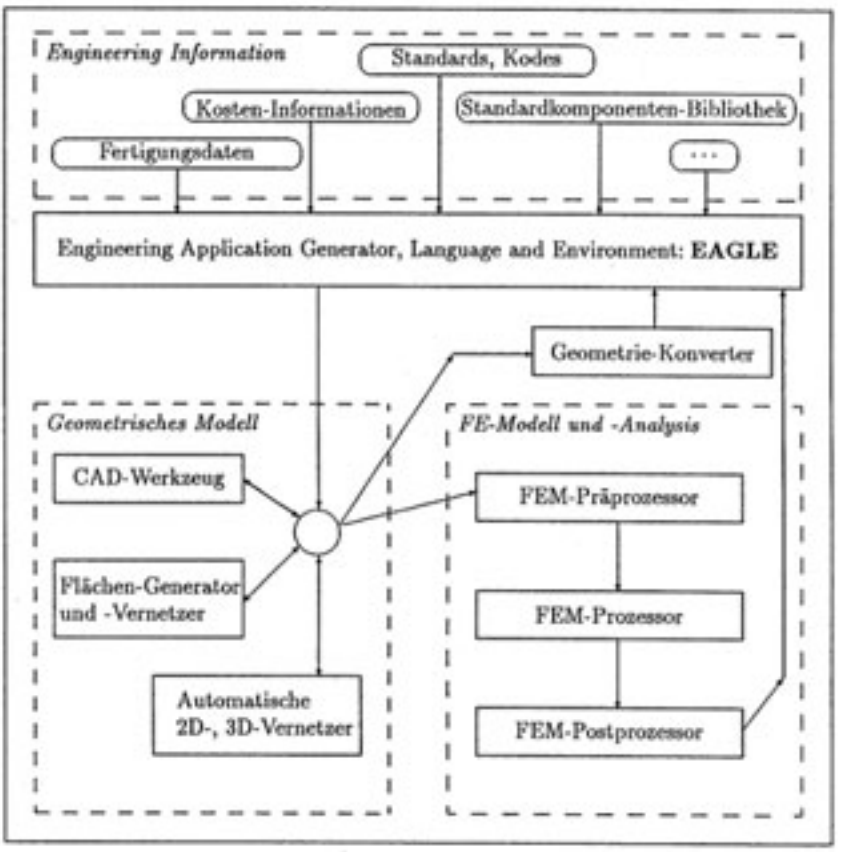


Abbildung 9: Vorstellung von einem Konstrukteurs-Arbeitsplatz

Die äußere Schale der Pyramide, die Optimierung, bleibt allerdings seit nunmehr Jahrzehnten trotz nutzerfreundlicher und oft mathematisch völlig anspruchsloser Software hinter den Möglichkeiten zurück. Obwohl nach zahlreichen Erfahrungen bereits in der vorbereitenden Modellierung meist wesentliche neue Einsichten gewonnen werden können, auf die der Optimierungseffekt noch zuzuschlagen ist, beschränken sich die meisten Entscheidungsträger mit bescheidenen Variantenvergleichen. Offenbar wirkt noch stärker als bei der Mehrkörperdynamik diese Modellierungsaufgabe als Hürde.

Parallel zur Ausrüstungsentwicklung haben sich auch die Vorstellungen von der Produktentwicklung ständig gesteigert. Wenn am Ende der DDR in der Entwicklung von Werkzeugmaschinen bereits die „virtuelle Maschine“ als Vision verfolgt wurde, so gibt es heute tatsächlich bereits aus Journalistenquelle die – natürlich fehlinterpretierte – Behauptung, Boeing hätte seinen „Dreamliner“ am Computer und völlig ohne Bau von Prototypen entwickelt. Zwar taucht auch z.B. bei Airbus in Projektnamen der „virtual bird“ auf, die Investitionen in physikalisch-materielle Arbeit sind aber mit Sicherheit nicht niedrig.

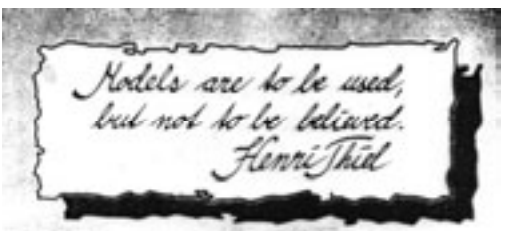


Abbildung 10: Aufforderung zum Umgang mit Modellen

Um ein Modell gemäß Abbildung 10 nutzen zu können, müssen u.a. seine Parameter bekannt sein. Seit über hundert Jahren ist belegt, wie sich auch sehr kleine Unterschiede z.B. in Instabilitäten auswirken können. Und selbst beste FE-Programme oder andere CAE-Tools werden in aller absehbaren Zeit nicht in der Lage sein, alle Steifigkeiten, Dämpfungen und weitere wichtige Größen bei komplexen Objekten präzise zu ermitteln.

Materielle Versuche sind prinzipiell nicht verzichtbar. Darüber hinaus stellt sogar gerade heute mit den gewaltigen Mitteln von Hard- und Software das spezielle menschliche Know-how eine weitere entscheidende Erfolgskomponente dar.

4 Eine abschließende These zum Umgang mit Hard- und Software

Es tut gut, die Mühen der Anfänge hinsichtlich Speichergröße, Programmiersprache, Rechengeschwindigkeit, Fehlersuche u. dgl. zu kennen, denn dann fällt es wesentlich leichter, sich immer über das Viele zu freuen, das geht, anstatt sich über das zu ärgern, was hin und wieder nicht geht.

Zur Entwicklung der Fertigungsinformatik in der DDR

Wilfried Krug

DUALIS GmbH IT Solution Dresden
Tiergartenstraße 32
01219 Dresden
wkrug@dualis-it.de

Abstract: Im Beitrag wird aufgezeigt, wie innovative Komponenten der Angewandten Informatik in den 80 er Jahren in der DDR zur Entwicklung der Fertigungsinformatik für die Ausbildung, Forschung und praktische Nutzung bis zur Gegenwart führten. Zunächst wird an den Beitrag vom 2. Symposium 2006 in Erfurt [Kr06] zur Entwicklung und Anwendung der Analog-, Hybrid– und Digital-Rechentechnik angeknüpft. Danach erfolgt eine Veranschaulichung und Begründung der Entwicklungsaspekte der Fertigungsinformatik an Hand Rechnergestützter Systeme der Entscheidungsfindung mit Simulation und Optimierung von Fertigungsprozessen in CAD/CAM/CIM. Dabei werden kritisch die Hemmschwellen herausgearbeitet, die es vor und nach der Wende 1990 zur gezielten Profilierung der Fertigungsinformatik für Maschinenbau-Ingenieure gab.

1 Politische und fachliche Vorrausetzungen

Mitte der 80 er Jahre entfachte sich in der DDR eine regelrechte Euphorie zum Einsatz der Rechentechnik und Informationsverarbeitung in der Volkswirtschaft. Dies hatte zur Folge, dass im Jahr 1985 die Gesellschaft für Informatik in der DDR (GIDDR) gegründet wurde. In der Fachsektion „Anwendungen“ fanden sich die Fachgebiete CAD/CAM-Systeme und Computergrafik sowie die rechnergestützte Entscheidungsfindung und rechnergestützte Simulation deutlich wieder [Me06].

So erfolgte 1986 vom Vorstand der GIDDR eine Aufforderung an die in der „Volkswirtschaft Tätigen“, die Informatik – und CAD/CAM – Ausbildung als Pflichtaufgabe festzulegen und hinreichende Bedingungen zu schaffen. Gleichzeitig sollten auch langfristige Planungen für derartige Vorhaben angegangen werden, wie in [Me06] ausgeführt.

Bereits im Jahr 1985 wurde an der TU Dresden beschlossen, die CAD/CAM–Ausbildung in der Fakultät Maschinenwesen, Sektion Fertigungstechnik und Werkzeugmaschinen, aufzubauen. Der Aufbau des Lehrkörpers zu dieser Ausbildung und natürlich auch zur Forschung erfolgte zunächst sehr zögerlich, da es an einem

modernen CAD/CAM–Labor fehlte. Nach Konzipierung des CAD/CAM–Labors erfolgte zwar der erste Spatenstich im Jahr 1986 an der Bergstraße im TU–Gelände wie die Abbildung 1 zeigt, aber der Baubeginn zog sich über mehrere Jahre hin.

Dies hatte natürlich negative Auswirkungen auf eine tiefgründige praktische Ausbildung. Zunächst wurde in der Sektion Fertigungstechnik und Werkzeugmaschinen ein Fachbereich CAD/CAM-Labor mit den Lehrstühlen: Technologische Programmierung, Fertigungsleittechnik sowie Modellierung und Simulation aufgebaut



Abbildung 1: Erster Spatenstich zum CAD/CAM – Labor an der TUD (Foto v. S.Jetschny, CIMTT-TUD)

Der theoretische Lehrbetrieb begann 1986 mit einigen Komponenten der rechnergestützten Fertigungsgestaltung vom Entwurf des Produkts über die Planung bis hin zum Fertigungsprozess. Wenige Praktika erfolgten dabei notdürftig im Werkzeugmaschinenlabor des Kutzbach-Bau ab 1987, wie Abbildung 2 erkennen lässt. Dabei wurden bereits die international gängigen englischen Begriffe CAD (für rechnergestützte Konstruktion), CAP (für rechnergestützte Planung), CAM (für rechnergestützte Fertigung), CAQ (rechnergestützte Qualitätssicherung) in die Ausbildung eingeführt und gelehrt [Kr89].

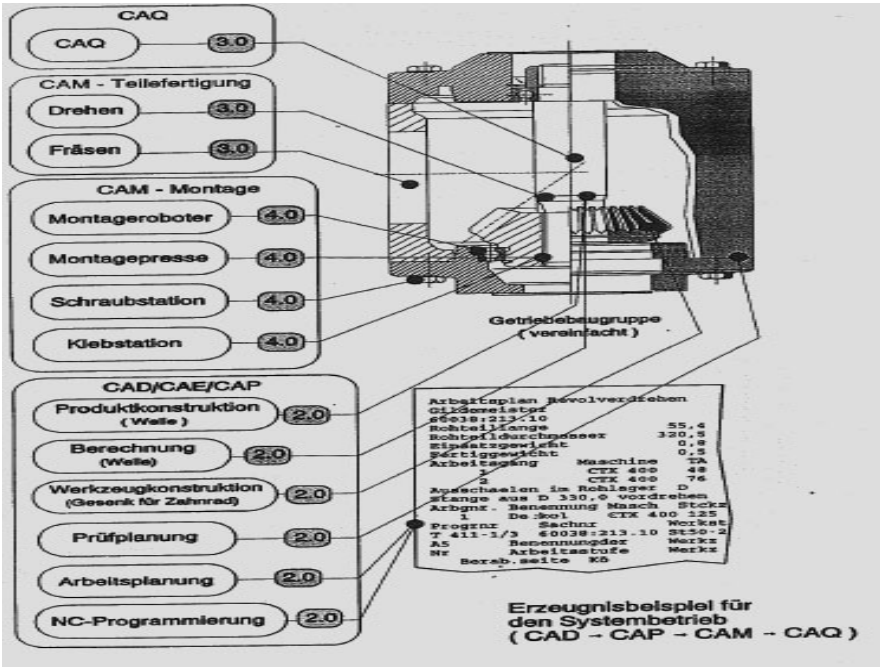


Abbildung 2 : Beispiel einer rechnergestützten Ausbildung mit Teilpraktika [He92]

Erste theoretische Forschungsarbeiten zu den einzelnen Komponenten wurden in den Lehrstühlen des Fachbereichs „CAD/CAM – Labor“ ab 1987 durchgeführt. Besonders erfolgte die weitere Integration der rechnerunterstützten Arbeitsweisen bis hin zur Schaffung eines durchgängigen Informationsflusses von CAD bis CAM. Dies führte schließlich zur rechnerintegrierten Fertigung, was in der Fachwelt mit CIM (Computer Integrated Manufacturing) bezeichnet wurde. Aus der einschlägigen Fachliteratur war eine Herangehensweise bei CIM in der Industrie bekannt, wie in Abbildung 3 dargestellt ist.

Innovative Weiterentwicklungen und Forschungsarbeiten für CIM konnten aber im Fachbereich auf Grund des immer noch fehlenden Praxislabors nicht erschöpfend bearbeitet werden. Weiterhin waren einige Hemmschwellen bezüglich der Hardwareseite zu überwinden, die sich nach wie vor in der DDR noch sehr stark auf die Nutzung von Großrechnern in Rechenzentren orientierte. CIM - Forschungsarbeiten konnten daher nur vereinzelt auf so genannten Embargorechnern (PCs) arbeitsplatzbezogen erfolgen.

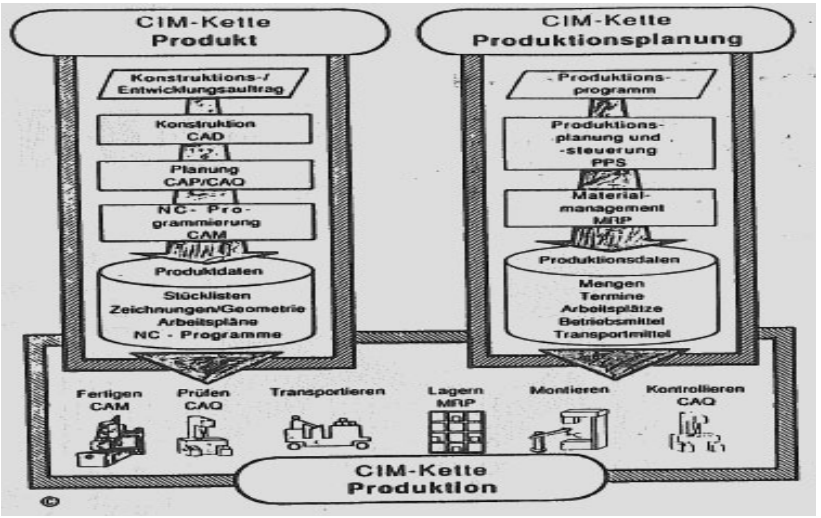


Abbildung 3 : Rechnerunterstützter Informationsfluss - CIM–Ketten [Kr91]

Die Abbildung 3 lässt erkennen, dass zum Ende der 80er Jahre in CAD/CAP/CAM und CIM die Kreativtechnologien auch in der Fachwelt mit CAE (Computer Aided Engineering) bezeichnet, noch keine bedeutende Rolle spielten [Jh90].

Es zeigte sich aber bald, dass beim Einsatz von CAD/CAM und CIM zur Optimierung der Informations- und Produktionsprozesse in der Industrie CAE unumgänglich sein wird.

Dieser Tatbestand stellte sich auch in einer ersten CIM-Einsatzanalyse gemäß Abbildung 3 z.B. bei der durchgängigen Entwicklung und Konstruktion über die Fertigungsplanung bis zur Fertigung der Offset-Druckmaschinen des VEB Planeta Radebeul (CIM-Pilotbetrieb) heraus.

Es wurde daher die Einbettung und Integration von Kreativtechnologien in Form von modellbasierten Simulations- und Optimierungssystemen in die CIM–Ketten vom Entwurf eines Produkts über die Planung bis zur Herstellung des Produkts im Fachbereich „CAD/CAM–Labor“ 1988 veranlasst [Kr90] und in die Ausbildung von Meisterschülern einbezogen [KJ89].

Das inzwischen aus den Erfahrungen der Analog-, Hybrid- und Digital-Rechentechnik abgeleitete Toolset MOSAIK, wurde im Rahmen eines Forschungsprojekts des CAD/CAM – Labors in die CIM– Philosophie integriert, wie Abbildung 4 erkennen lässt. Während im Rahmen des Entwurfs des Produkts die kontinuierliche Simulation stärker zum Einsatz kam, wurde bei den Fertigungsprozessen die ereignisorientierte Simulation (diskrete Simulation) favorisiert. Falls beides zeitgleich auftrat, war die kombinierte Simulation erforderlich.

Dafür wurden in MOSAIK Softwarekerne und Modellbasen integriert, die eine schnelle Modellierung der kontinuierlichen und/oder diskreten Systeme in den jeweiligen Simulatoren vornehmen können. Auch ein Optimierungssystem war Bestandteil von MOSAIK, um Produktentwicklungs-, Planungs– und Fertigungsprozesse nach unterschiedlichen Zielen und Nebenbedingungen optimieren zu können.

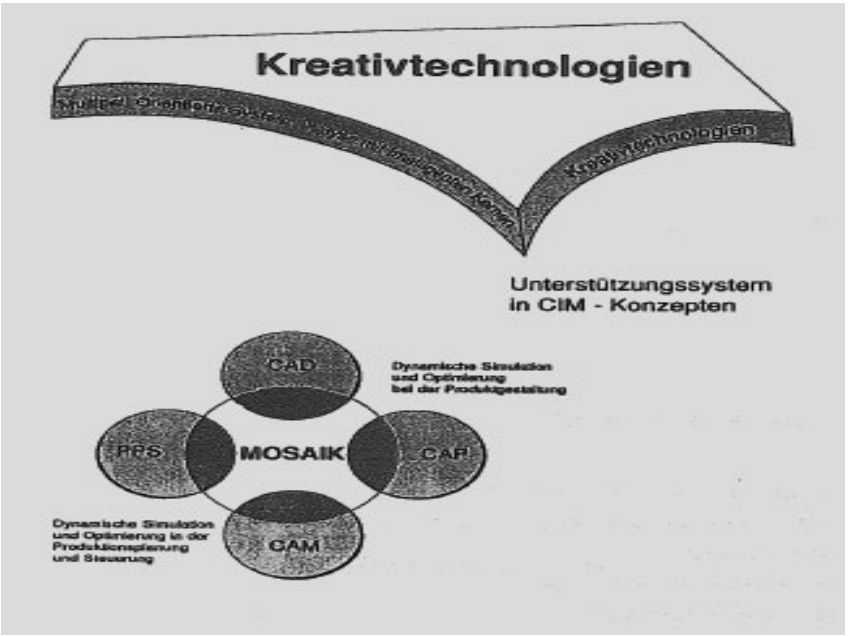


Abbildung 4: Integration von Kreativtechnologien in CIM–Konzepten [Kr90]

Zur multivalenten Nutzung von MOSAIK bei CIM – Anwendungen in der Industrie wurde die in Abbildung 5 dargestellte Softwarearchitektur entwickelt. Es ist ersichtlich, dass einerseits das Prozess– und Datenmanagement von der CIM–Plattform zu MOSAIK und zurück, sowie anderseits die Ankopplung unterschiedlicher Nutzer von Hardware- und Softwarekomponenten über Interfaces ermöglicht wurde.

Damit mussten Ende der 80er Jahre die unterschiedlichsten Rechnerkonfigurationen an der TU Dresden und in der Industrie, wie z.B. beim Vorzeigeunternehmen für CIM, dem VEB Planeta Radebeul, unterstützt werden.

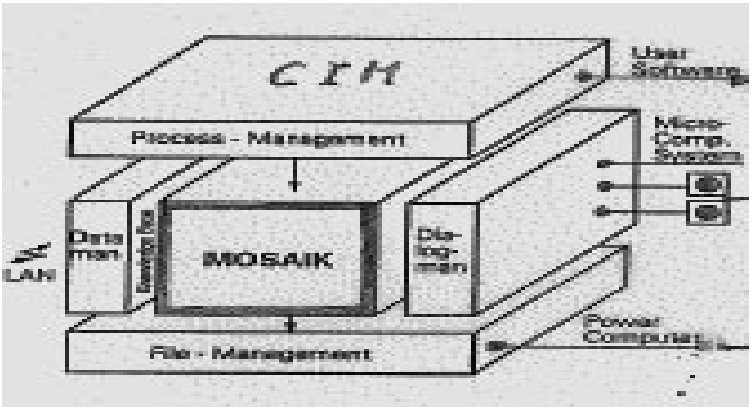


Abbildung 5 : Prozess - und Datenmanagement von MOSAIK [Kr90]

Ein Beispiel zur Entwurfsunterstützung und Optimierung eines Schweißprozesses in einer CIM–Umgebung zeigt Abbildung 6. Der links dargestellte Schweißprozessablauf konnte mit der Modellbasis von MOSAIK in kurzer Zeit in den zur Verfügung stehenden Simulator von MSOKS (Modellierungs-, Simulations-, Optimierungs-Kern) rechts umgesetzt werden. Danach erfolgte die Aktivierung des Optimierers aus MSOKS und es konnte die Dimensionierung, Steuerung und der Produktionsdurchsatz des Schweißprozesses der CIM–Kette: Produktion, mit den Planungsvorgaben aus der CIM–Kette: Entwurf und der CIM–Kette: Planung vorgenommen werden. Über die Nutzeroberfläche von MOSAIK erfolgte im Mensch-Maschine-Dialog die CIM-gerechte Bewertung der Ergebnisse mit dem Grafischen Kernsystem GKS.

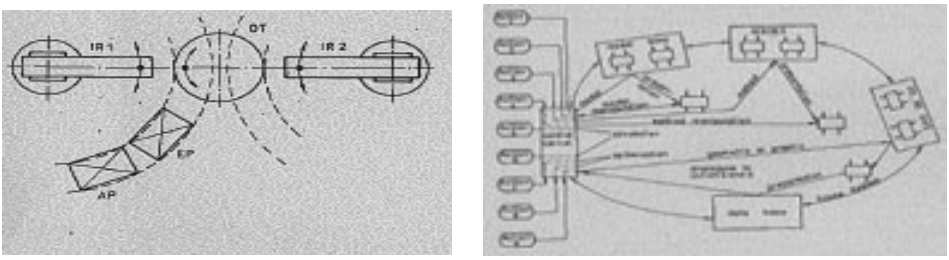


Abbildung 6 : Simulation eines Fertigungsschweißprozesses mit MOSAIK [Bl89]

2 Inhaltliche Gestaltung der Ausbildung

Zum Ende der 80er Jahre war die Informatikausbildung für die Maschinenbauer nicht mehr zeitgemäß. Kurz gesagt, die Schere zwischen Forschung und Lehre auf dem Gebiet der CA–Informationstechnologien ging immer mehr auseinander.

Es wurde daher entschieden, für alle Maschinenbauer (1989/90/91 ca. 400 Studenten des 3. Studienjahres) der Fakultät Maschinenwesen eine Grundlagenausbildung in CAD/CAM und CIM ab 1989 durchzuführen [Kr91]. Die Inhalte sind in Abbildung 7 dargestellt und zeigen einen Umfang von 30 h Vorlesung und 15 Stunden Übungen und Praktika.

Zum Abschluss der Vorlesung wurde eine Videokonferenz zur Fabrik der Zukunft durchgeführt, die von den Stundenten sehr wohlwollend aufgenommen wurde. Das Interesse dieser Ausbildung war sehr groß, weil die Studenten eine große Perspektive in ihrem künftigen Berufsleben sahen. Der Grundstein dazu wurde in den praktischen Übungen und Praktika gelegt. Während des Praktikums mussten CAD/CAM– Aufgaben aus der Industrie gelöst werden.

So waren z.B. Teilaufgaben aus der konstruktiven Entwicklung von Druckmaschinen beim VEB Planeta Radebeul mit dem CAD System ICEM und das Schwingungsverhaltens der Druckerantriebe mit MOSAIK zu lösen. In der Computerunterstützen Fertigung CAM mussten die Prozesse mit MOSAIK modelliert, simuliert und optimiert werden.


Lehrgebiet

CAD/CAM-Grundlagen

Hochschullehrer: Prof. Dr.-Ing.habil. H. Krug
Umfang: 30 h Vorlesung, 15 h Übungen/Praktika

Inhaltliche Gliederung:

1. Analyse und Synthese informationeller Prozesse
1.1. Arbeitsbereiche des Produktionsprozesses
1.1.1. Produktentwicklung
1.1.2. Produktionsplanung
1.1.3. Arbeitsvorbereitung
1.1.4. Fertigung/Montage
1.2. Mensch-Maschine-Interaktion
1.3. Informations- und Materialfluß in Produktionsprozeß
1.3.1. Stand der Durchdringung
1.3.2. Technische Systeme (CAD, CAP, CAM, ...)
1.3.3. Betriebswirtschaftliche Systeme (PPS,...)
1.3.4. Kommerzielle Systeme
1.3.5. Datenstruktur von der Produktionsplanung bis zur Fertigung
1.3.5.1. Planungsdaten
1.3.5.2. Konstruktionsdaten
1.3.5.3. Arbeitsvorbereitungsdaten
1.3.5.4. Betriebsdaten
1.4. Industrielle Logistik
1.4.1. Produktionslogistik
1.4.2. Beschaffungslogistik
1.4.3. Distributionslogistik
1.4.4. Integrierte Logistik

2. CAD/CAM-Architekturen
2.1. Architekturkonzepte
2.1.1. Sicht-Zeithorizont
2.1.2. Sicht-Datenflußorientierung
2.1.3. Sicht-CIM

3. CAD/CAM-Technik
3.1. CAD-Technik
3.1.1. Einordnungsgesichtspunkte
3.1.1.1. Art der Dimension (2 D, 2 ½ D, 3 D)
3.1.1.2. Art der Modellierung
3.1.1.3. Art der Funktionalität
3.1.2. Hard- und Softwarevoraussetzungen


Übungen / Praktika

Zur CAD-Technik:

1. 3D-Modellierung mit System DCEN
- Linien-, Flächen-, Volumendarstellung an Beispiel von Maschinenbauteilen
- Bemaßung von Bauteilen

Voraussetzung: AUTOCAD-Übungen und Vorlesung CAD/CAM-Grundlagen

Umfang: 4 SE

2. Auftragsabwicklung mit System [illegible] und System [illegible]
- Stücklistenverwaltung unter heterogenen Randbedingungen (Baukastenstückliste, Mengenstückliste, Strukturstückliste, Teileliste, Teileverwendungsnachweis, Teilmaterialtausch mit Löschen, Stückliste kopieren)
- Auftragsdurchlaufsimulation
- Produktionsplansimulation

Voraussetzung: Vorlesung CAD/CAM-Grundlagen

Umfang: 2 SE

Zu CAP-Technik

3. Simulation in der Produktionsautomatisierung mit System MOSAIK/ERGOS
- Simulationsmodell (Antriebe, Steuerprozesse)
- Simulationsexperimente und Auswertung

Voraussetzung: Vorlesung CAD/CAM-Grundlagen

Umfang: 2 SE

4. Optimierung in der Produktionsautomatisierung mit System MOSAIK/MOLAB-PC
- Optimierungsmodelle (Entwurf, Auslastung u.a.)
- Auswertung der Optimierungslösungen

Voraussetzung: Vorlesung CAD/CAM-Grundlagen

Umfang: 2 SE



Abbildung 7 : Ausbildungsprofil in CAD/CAM/CIM für Maschinenbauer an TUD [Kr91]

Mit der Umgestaltung der Fakultät Maschinenwesen ab 1990 bildeten sich Institute mit neuen Lehrgebieten heraus. Für die zukunftsweisenden Fachgebiete waren geeignete Namen zu kreieren.

Es war herangereift, in Analogie zur Wirtschaftsinformatik für die Betriebswirtschaftler, die Fertigungsinformatik für Maschinenbauer zu lehren. An Stelle des Fachbereichs: „CAD/CAM–Labor“ entstand 1991das Institut für Fertigungsinformatik IFI [Vo91].

Das inzwischen gegründete CIMTT–Zentrum an der TU Dresden, als Industrietransferzentrum, spiegelte die Inhalte der Fertigungsinformatik ebenfalls anschaulich wider, wie aus Abbildung 8 ersichtlich ist [Vo92].

Die Inhalte von Forschung und Lehre wurden auf den Prüfstand gestellt und 1991 auf der Wartburg bei Eisenach mit Fachleuten aus Ost und West diskutiert und als zukunftsweisend befunden [Gr92]. Auch in Workshops des VDI Arbeitskreises „Produktionstechnik“ gab es regen Zuspruch von Spezialisten aus ganz Deutschland. Bei Diskussionsforen zur modernen fachbezogenen Informatikausbildung auf internationalen Fachkonferenzen wurde dieser Trend ebenfalls bestätigt [Le92].

Damit konnte nahtlos die modernisierte Ausbildung in Fertigungsinformatik als Hauptfach des Instituts für alle Maschinenbauer (1992 waren es noch ca. 300 Studenten) durchgeführt werden.

Doch Ende 1992 kam alles anders. Eine Fach-Evaluierungskommission von der Universität Stuttgart entschied, dass das 1991 im Rohbau fertig gestellte CAD/CAM- Labor an der Bergstraße zu einer juristischen Fakultät umfunktioniert wird. Weiterhin wurde entschieden, dass das Institut für Fertigungsinformatik zu liquidieren ist, weil es solch ein Institut in Analogie zur Universität in Stuttgart nicht gab.

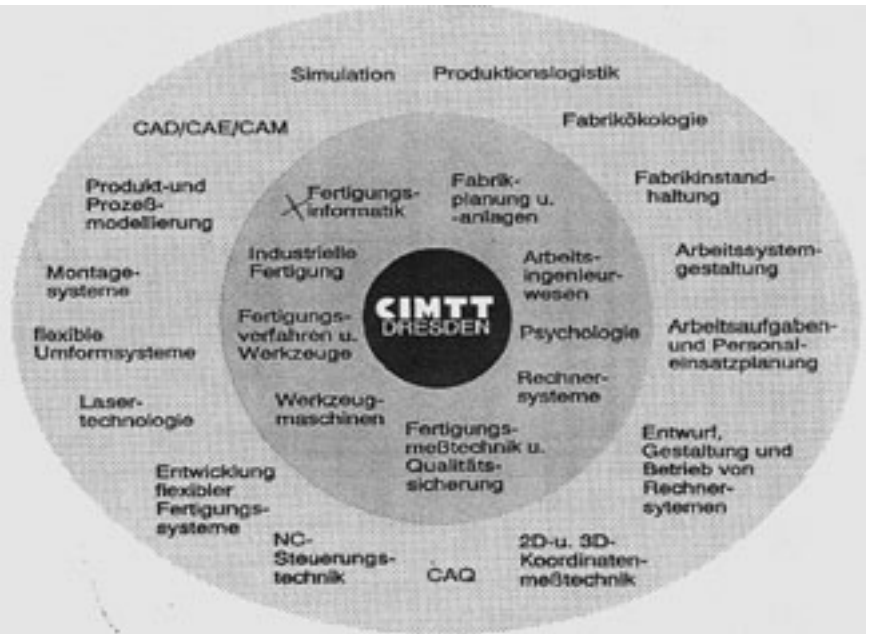


Abbildung 8 : CIM–Technologie-Transferzentrum CIMTT Dresden [Vo92]

Heute gibt es in Stuttgart und an vielen anderen Technischen Universitäten und Hochschulen in Deutschland in der modernen Maschinenbauausbildung die Fertigungsinformatik als etabliertes Ausbildungsfach! Eine derartige Ausbildung ist an der HAW Hamburg in www.haw.de anschaulich dargestellt. Eine Spiegelung der Ausbildungsinhalte von damals zur Fertigungsinformatik an der TU Dresden ist unverkennbar.

3 Impulse für die produzierende Industrie

Bereits 1990 wurde die DUALIS GmbH als Aninstitut der TUD gegründet, um Impulse zur breiten Anwendung von CIM–Lösungen des Instituts für Fertigungsinformatik der TUD in der Industrie auszulösen. Die Schaffung moderner Hard– und Softwarevorrausetzungen konnten recht schnell erfolgen. Darüber hinaus wurden gemeinsam mit DUALIS arbeitslos gewordene Maschinenbauingenieure als CIM-Ingenieure ausgebildet. In der aufstrebenden Sächsischen Industrie fehlte es besonders an dieser Berufsgruppe. Mit den Einsätzen der ausgebildeten Ingenieure konnten somit erste CIM-Lösungen in Maschinenbauunternehmen, wie z.B. im Verpackungsmaschinenbau Hainsberg, geschaffen werden.

Mit der Liquidation des Instituts für Fertigungsinformatik an der TUD, Ende 1992, wurde die universitäre Forschung mit der Industrie jäh in Mitleidenschaft gezogen. Damit konnte die komplette innovative Weiterentwicklung der MOSAIK- und CIM-Komponenten nur noch von DUALIS durchgeführt werden. Dabei standen vor allem Planungs-, Simulations– und Optimierungslösungen für CIM–Systeme der produzierenden Industrie im Mittelpunkt und wurden in [Kr97], [An97], [VK02] und [Kr02] publiziert.

Heute werden damit komplette Produktionssysteme mit IT (Informationstechnologien) projektiert, wie in Abbildung 9 für eine Digitale Fabrik sichtbar ist.

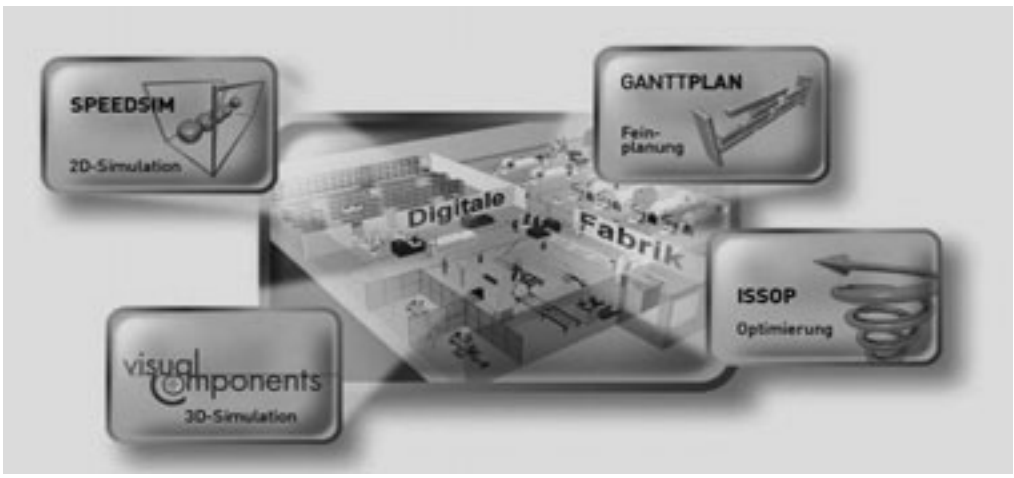


Abbildung 9 : Fertigungsinformatik im DUALIS – Umfeld einer Digitalen Fabrik [Du05]

Jährlich finden von DUALIS IT - Anwenderforen „FIT FOR IT“ für den Mittelstand in Dresden statt, wo ein reger Erfahrungsaustausch aller Fachleute auf dem Gebiet Digitaler Fabrikprozesse erfolgt [Du05], [Du07].

4 Ausblick

Das Fachgebiet der Fertigungsinformatik hat sich im letzten Jahrzehnt rasant entwickelt und ist heute ein wichtiger Bestandteil der Aus- und Weiterbildung der Informatik an den Universitäten und Hochschulen sowie in der produzierenden Industrie. Bis zur Geburt der Fertigungsinformatik musste ein steiniger Weg sowohl zu DDR-Zeiten in den 80er Jahren, als auch einige Jahre nach der Wende bis in das Jahr 2000 hinein zurück gelegt werden. Treibende Kraft für die Ausbildung an den Universitäten und Hochschulen war immer wieder die Industrie. Für die konsequente Digitalisierung und Visualisierung der Produktionsprozesse der Zukunft, wurden in der Industrie hoch qualifizierte IT–Spezialisten für die Planung, Entwicklung, Projektierung und den Betrieb der Digitalen Fabrik benötigt. Im Gegensatz zu CIM ist der Mensch heute und in Zukunft voll mit all seinen Fähigkeiten und Fertigkeiten integriert. Damit ist die Fertigungsinformatik mit den Teilgebieten der Modellierung, Simulation und Optimierung für einen sicheren und komplexen Fabrikbetrieb und für die Mensch–Maschine–Kommunikation zur Steuerung der Produktionsprozesse als zukunftsweisend anzusehen. Bis heute konnten alle gesammelten Erfahrungen und Entwicklungen aus der Architektur von MOSAIK und CIM aus DDR-Zeiten bis heute in leistungsfähige Produkte innovativ ständig weiterentwickelt werden. Wesentliche Impulse kamen dabei aus industriellen Fabrikprojekten der Automobil-, Prozess-, Lebensmittel-, Metallverarbeitungs- und Pharma– Industrie. DUALIS beteiligt sich ständig an den Weiterentwicklungen der Fertigungsinformatik über den Arbeitskreis „Produktionstechnik“ des VDI und die Bildungsgremien der Hochschulen im IT–Umfeld. Seit 1992 hat DUALIS über 200 Ingenieure auf dem Gebiet der Fertigungsinformatik ausgebildet und weiter qualifiziert und wird dies künftig noch stärker über die Innovationsforen „FIT FOR IT“ tun.

Literaturverzeichnis

[Kr89] Krug, W.: Simulation für Ingenieure, in CAD/CAM-Systemen, Verlag Technik Berlin, 1989, Hirarchieebenen der digitalen Simulation (kontinuierlich/diskret) und Einsatzbebiete in CAD/CAM/CIM, S. 52 – 70, Autorenkollektiv: O. Beyer, M. Frank, G. Buhle, W. Jetschny, W. Mylius, R. Sperling

[KJ89] Krug, W.: Zur Befähigung rechnerunterstützter Arbeitsweisen in der Produktions-Automatisierung (Meisterklasse), in: Wissenschaftliche Zeitschrift der TU Dresden, Heft 3, 1989, Herausgeber: J. Jacobs, S. 31 – 32

[Bl89] Blume, F.: System Analysis, Modeling and Simulation in CIM by the Example of the Flexible Automated Welding Process, in: Wissenchaftliche Zeitschrift der TU Dresden, Heft 3, 1989, Herausg. J. Jacobs, Mitautoren: F. Erfurth, H. Bauer, W. Krug, F. Blume, S. 197 - 199

[Kr90] Krug, W.: Kernsystem MOSAIK zur Modellierung, Simulation und Optimierung in CIM, in: Wissenschaftliche Zeitschrift der TU Dresden, Heft 6, 1990, Herausgeber: G. Landgraf, Mitautoren O. Beyer, J. Weber S. 43 – 52

[Jh90] Jhanker, B.: CAE und Produktdatenaustausch – Stand und Perspektiven in: CAE und Produktdatenbanken als Integrationsbausteine in CIM, Tagungsband zu Symposien u. Workshops CAD/CAM/CAE und CIM : Juli 1990 in Dresden, Edition B. Shanker, H. Grabowski, W. Krug, S. 3 – 15

[Vo91] Völkner, W.: CIM – Technologietransfer – eine Investition in die Zukunft, CIMTT Mitteilungen Dezember 1991, Herausg. W. Völkner, Forschungsberichte: B. Hetschel, H.-W. Eberl, W. Krug zu CAD/CAP/CAM/CAE/MOSAIK, S. 51-69

[Kr91] Krug, W.: Lehrgebiet CAD/CAM – Grundlagen, Institut für Fertigungsinf., TU Dresden, Stand 1991, Institut für Fertigungsinformatik (Herausgeber), Lehrbeauftragter: Prof. W. Krug

[He92] Hentschel, B.: Von der Konstruktion zur Fertigung, Mitteilung der Institute, Industrielle Fertigung IIF und Fertigungsinformatik IFI, Autoren: Engelhardt, Köhler, Geipel, Schwarz, Eberl, 1992

[Gr92] Grabowski, H.: Voraussetzungen in CAD/CAM - Systemen für Simultan. Engineering in: Integrierte Produkt – u. Prozessgestaltung, Tagungsband April 1992 auf der Wartburg, S. 2 – 24/

[Le92] Lehmann, A.: Simulation and AI in Comp. Aided Techniques, (CAD/CAM/PPS/CIM) Poceedings of the 1992 EUROPEAN SIMULATION SYMPOSIUM, November 5-8, 1992, Dresden, Germany, Edited by W. Krug, A. Lehmann

[Vo92] Völkner, W.: CIM – Technologie-Transferzentrum Dresden, CIMTT Mitteilung, März 1992, Herausg. W.Völkner, Aus - und Weiterbildung, Beratung und Forschung J. Schaller, G. Nestler, W. Krug, D. Hegt zu CAD/CAM/PPS/CAE/CIM des Instituts für Fertigungsinformatik, TU Dresden, S. 10 -24

[Kr97] Krug, W.: Intelligentes Simulations – und Optimierungssystem für Prozesse der Fertigung, Organisation und Logistik, 1997, Applikationen in CAD/CAM, Edited by R. Rimane, SCS EUROPE , 1997

[An97] Anulf, S.: Vision of Manufacturing in the 21st Century, in: Proceedings, "The European Conference on Integration in Manufacturing IiM 1997, September 1997, Herausgeber: D. Fichtner, Speaker: B. Anulf, S. 5 - 20

[VK02] Verbraeck, A.; Krug,W.: Simulation in Industry, 14th European Simulation Symposium, Oct.2002, Dresden, Proceeding in: www.scs.com

[Kr02] Krug, W.: Modelling, Simulation and Optimization for manufacturing, organisation and logistical processes, SCS – European Publishing House Delft, Erlangen, Ghent, San Diego, ISBN 3- 936150 – 20-6, 2002

[Du05] Virtueller Service -reale Verbesserungen, Softwarebasierte Materialflussoptimierung, handling, 2005 S.76-77, www.handling.de, www.dualis-it.de

[Kr06] Krug, W.: Entwicklung und Anwendung der Analog-, Hybrid – und digitalen Rechentechnik zur Modellierung, Simulation und Optimierung technischer Prozesse u. Syst., in: Informatik in der DDR - eine Bilanz, Tagungsband zu Symposien: Okt. 2004 in Chemnitz u. Mai 2006 in Erfurt, GI – Edition, F. Naumann, G. Schade, S. 556 - 567

[Me06] Merkel, G.: Bildung und Wirken der Gesellschaft für Informatik der DDR, in: Informatik der DDR – eine Bilanz, Tagungsband zu den Symposien: Oktober 2004 in Chemnitz u. Mai 2006 in Erfurt, GI – Edition, F. Naumann, G., Schade,S., S. 451 – 461

[Du07] Prozess– und Kostenoptimierung mit IT der Digitalen Fabrik, in: Vortragsband, 4. Sächs. IT–Anwenderforum „FIT FOR IT", H. Krug, DUALIS GmbH IT, Nov. 2007

Zur Gestaltung medizinischer Informationssysteme und zur Entwicklung der medizinischen Systemforschung in der DDR

Klaus Fuchs-Kittowski

Fachhochschule für Technik,
und Wirtschaft Berlin
Treskowalle 8, 10313 Berlin
fuchs-kittowski@t-online.de

Hildebrand Kunath

Institut für Medizinische Informatik
und Biometrie der Medizinischen Fakultät
der TU Dresden
kunath@imib.med.tu-dresden.de

Abstract: Es soll über die Probleme der theoretischen und praktischen Bewältigung des Einsatzes der Informationstechnologien in Medizin und Gesundheitswesen der DDR berichtet werden. Dabei ist insbesondere über die Entwicklung des Institut für Medizinische Informatik an der „Medizinischen Akademie Gustav Carus“ in Dresden, über die Erfahrungen der Einsatzgruppe im Krankenhaus Friedrichshain (automatisierte Patienteninformationssystem FRIDA), die Entwicklung der Medizinischen Informatik an der Charitè und über Arbeiten im Bereich Systemgestaltung und automatisierte Informationsverarbeitung der Sektion WTO der Humboldt-Universität zu Berlin, über die Anwenderkonferenzen: “EDV im Gesundheitswesen“ 1974, 1975, 1976, 1982 und 1987 und den Kongress „Probleme der Informatik in Medizin und Biologie“ 1982 in Berlin zu berichten, sowie über die internationale Zusammenarbeit zum „health care modeling“ und die Entwicklung der medizinischen Systemforschung bzw. Health Care Research (heute Versorgungsforschung) an der Akademie für ärztliche Fortbildung.

1 Wichtige Impulse für die Entwicklung der medizinischen Informatik in der DDR

1.1 Zu den Entwicklungen gegen Ende der 60-er und zu Beginn der 70-er Jahre

Als Mitarbeiter der Gruppe Datenverarbeitung des 1964 an der Humboldt Universität, gegründeten Rechenzentrums, wurde ich (K. Fuchs-Kittowski) 1968 gebeten, auf dem Symposium: „Sozialismus, wissenschaftlich-technische Revolution und Medizin“ ein Referat zum Thema: „Medizin und Datenverarbeitung“ zu halten [FU68]. Im Referat wurden drei Problemkreise herausgearbeitet, für die die Anwendung der Kybernetik, Mathematik und der modernen Rechen- und Datenverarbeitungstechnik besonders wichtig ist: „Erstens für die experimentelle Forschung, zweitens für die ärztliche Praxis

und drittens für die Organisations- und Leitungspraxis der einzelnen Klinik und des Gesundheitswesens.“ Zum Abschluss wurde formuliert: „Es kann kein Zweifel darüber bestehen, dass die fortschreitende Automation von grundsätzlicher Bedeutung für die weitere Entwicklung der Medizin ist. Sie wird in einem entscheidenden Maße die medizinische Forschung aber auch das Berufsbild des Arztes bereichern. In die Lehre müssen daher unbedingt die Probleme der medizinischen Dokumentation und die Methoden der Datenverarbeitung in der Medizin sowie Kenntnisse über die Datenverarbeitungstechnik, Fragen der maschinellen Auswertung und statistischen Analyse, aber auch der Organisation des Krankenhauses aufgenommen werden. Es sind weitgehende soziologische Untersuchungen angebracht, die die Auswirkungen des Einsatzes elektronischer Datenverarbeitungsanlagen auf die Struktur medizinischer Einrichtungen und die Konsequenzen für die Tätigkeit des Arztes sowie die Konsequenzen für den Patienten deutlich werden lassen. Ich hoffe, dass von diesem Kongress die notwendigen Impulse ausgehen werden.“ [FU68 S. 222]

Mit dem Symposium des Rates für Planung und Koordinierung der medizinischen Wissenschaften: “Sozialismus, wissenschaftlichtechnischen Revolution und Medizin” erfolgte auch in der Medizin und im Gesundheitswesen der DDR eine stärkere Hinwendung zu den Problemen des EDV-Einsatzes. Dies führte (1969) zur Bildung eines ORZ`s an der Medizinischen Akademie Carl Gustav Carus in Dresden. Das Organisations- und Rechenzentrum führte unter der Leitung von G. Penzel regelmäßig Informationslehrgänge zur „EDV im Gesundheitswesen“ durch. Es sei hier insbesondere auf den 3. 1973 und den 4. 1974 in Rehefeld verwiesen [PE74], [PE75]. Dass sich daraus, unter Leitung von H. Kunath, entwickelnde Institut für Medizinische Informatik wurde zur Leiteinrichtung für die Einführung der IKT in Medizin und Gesundheitswesen. Die Auswahl der Dresdner medizinischen Hochschuleinrichtung als Leiteinrichtung entsprach dem Strukturkonzept der Konzentration der Rechentechnik im Raum Dresden, im Zusammenhang mit der Gründung des Kombinats Robotron [KU04].

1.2. Die prognostische Studie „Wissenschaftliche Grundlagen für die Optimierung menschlicher Lebensprozesse

Mit der vom Forschungsrat der DDR in Auftrag gegeben „Prognostischen Studie: Wissenschaftliche Grundlagen für die Optimierung menschlicher Lebensprozesse“ sollte in engem Zusammenhang mit der Biologieprognose, stärker auf die notwendigen Wissenschaftsentwicklungen in der Medizin orientiert werden. Über die Ergebnisse der Studie wurde auch im Internationalen Institut für Angewandte Systemanalyse (IIASA) vorgetragen [FRS75]. Das IIASA war von der USA, der UdSSR und der DDR mit dem Ziel gegründet worden, zur Sicherung der friedlichen Koexistenz eine den Eisernen Vorhang überwindende bzw. überbrückende Zusammenarbeit an globalen Problemen der Industrieländer zu realisieren. Aus der Mitarbeit von DDR-Wissenschaftlern am Projekt „Systems Modeling in Health Care“ [SH77], wie auch am „IIASA Computer Network“[BU75], bis hin zur Beteiligung des Instituts für Wissenschaftsinformation in der Medizin an der Charité an der Erprobung des IIASA-Netzes (R. Hauben [HA04]),

ergaben sich weitere Impulse für die medizinische Informatik sowie die Forschung zu System Aspects of Health Planning bzw. zur medizinischen Systemforschung.[1]

2 Zur Entwicklung des Instituts für Medizinische Informatik an der Medizinischen Akademie Carl Gustav Carus in Dresden

Das Fachgebiet der Medizinischen Informatik (MI) konnte sich zu Beginn der achtziger Jahre in Wissenschaft und Gesundheitswesen der DDR etablieren. Die Notwendigkeit der Gestaltung medizinischer Informationssysteme – der ersten Generation – bot dazu die Grundlage. Im Krankenhausbereich wirkte die Einführung der vierstelligen Diagnosekodierung mittels ICD 8 in die Krankenblattdokumentation und der Ausbau der Gesundheitsberichterstattung als Stimulus für die Entwicklung von Krankenhaus-informationssystemen. Patientenadministration, Basisdokumentation sowie fachgebiets-spezifische Anschlussdokumentationen, Arztbriefschreibung und spezielle Krankheits- und Autopsieregister bildeten die wesentlichen Anwendungsgebiete für die Zentral-rechner der 2. Generation mit ihren großen Magnetbandspeichern. Dies ermöglichte die Verarbeitung großer Datenmengen, die kooperativ genutzt werden sollten. Damit begannen die Probleme mit nicht standardisierten medizinischen Nomenklaturen und Begriffssystemen und damit die Arbeiten zur Erweiterung der ICD8.

Ende der siebziger Jahre wurde das Dresdner Patienteninformationsverarbeitungssystem (PIV) auf zahlreichen Plattformen der aufkommenden Kleinrechentechnik implementiert. Die rechnergestützte Automatisierung des klinisch-chemischen Labors war ein wesentlicher Schritt für die Entwicklung der Medizinischen Informatik in Dresden und auch in Berlin. Ende der siebziger Jahre war die verfügbare Mikrorechentechnik (INTEL 1980:8008, 1983: ZILOG80, 1984:INTEL8086). Dies ermöglichte die Entwicklung und den Einsatz kostengünstiger dezentraler Systemzugänge in Arbeitsplatznähe. Die Entkopplung von zentralen und peripheren DV-Prozessen wurde später mit dem Begriff Client-Server bezeichnet. Ein Beispiel dafür war die Entwicklung eines Mikrorechnerarbeitsplatzes (siehe Abbildung 1) für die 400 Betten umfassende Medizinische Klinik der FSU Jena.



Abbildung 1: Mikrorechnerarbeitsplatz

[1] Durch entsprechende Kontakte konnten z.B. nach langem Ringen, die Voraussetzungen dafür geschaffen werden, dass die Medizininformatik der DDR in die IMIA aufgenommen wurde. Denn der japanische Vertreter im IIASA-Projekt, Shigeko Kaihara, Leiter des Hospital Computer Center, University of Tokyo, bewerkstelligte einen Kooperationsvertrag zwischen der University of Tokyo und der Humboldt-Universität zu Berlin, wurde der Organisator der MEDINFO 80 in Tokio , [FKW80] und bald darauf der Präsident der IMIA.

Numerische Verfahren für die lineare Algebra und Statistiksoftware für die aufkommenden multivarianten Verfahren wurden entwickelt. Anwendungen in der Messwertverarbeitung (Neurowissenschaften, EEG, komplexe psychophysische Diagnostikprozeduren, Nystagmographie, Geburtsüberwachung durch automatisierte Analyse des Kardiotomogramms, EKG-Analyse) entstanden und mit ihnen Forschungs- und Entwicklungsgruppen. Die Beispiele zeigen deutlich, „dass die Ostgrenze der MI-Pioniere nicht der Main war“ [KU04]. Oder wie es R. Tschirschwitz zu gegebenen Anlass empört ausgedrückt hat: gab es „nicht nur weiße Flecken“ der Informatik in der DDR [TS02].

Bereits in den frühen achtziger Jahren gelang es der Medizinischen Akademie Dresden für die Medizinische Informatik eine wissenschaftliche Kooperation mit der Universität Uppsala zu realisieren. In diesem Rahmen spielt die Zusammenarbeit auf dem Gebiet Dantenbankmanagement (Einführung des DBMS MIMER) eine besondere Rolle. Es gelang das erste relationale System für die Wissenschaft und das Gesundheitswesen in der DDR zu entwickeln. Ab 1987 konnte zunächst in Dresden das Fach Medizinische Informatik neben der etablierten Vorlesung für Medizinische Statistik in der Lehre für Studenten der Medizin und der Zahnmedizin in den Lehrplan eingeführt werden. Ende der achtziger Jahre war die Medizinische Informatik in der DDR in einem breiten Spektrum etabliert. Die Medizinische Informatik der DDR litt allerdings unter einem technologischen Rückstand und Mangel, insbesondere unter der starken Vernachlässigung der öffentlichen Nachrichtennetze sowie unter dem eingeschränkten Wissenschaftsaustausch. Wie schon auf dem vorangegangenen Symposium Informatik in der DDR festgestellt wurde, betraf dies nicht nur die Kontakte mit dem Westen. Es gab wissenschaftliche Beziehungen in die Sowjetunion, so z.B. zum Leiter des Rechenzentrums, des Allunionsforschungsinstitut für Sozialhygiene und Organisation des Gesundheitsschutzes (Semaschko) in Moskau, V.M. Timonin, wie auch zu entsprechenden Einrichtungen in den anderen Ländern der Staatengemeinschaft, aber doch nicht mit der erforderlichen Intensität und Nachhaltigkeit.

3. Automatisierte Informationsverarbeitung zur Unterstützung der Dispensairebetreuung am Beispiel von FRIDA u. a. Anwendungen

Entscheidende Probleme der angewandten Informatik beim Aufbau von patientenbezogenen Informationssystemen wurden theoretisch wie praktisch durch die Pionierarbeit der Einsatzgruppe um P. Gudermuth und F. Manglus im Krankenhaus im Friedrichshain bewältigt [GM82]. Das automatisierte Informationsverarbeitungssystem FRIDA (Friedrichshainer Datenbank) wurde seit 1971 auf der Basis von Rechnern der ESER-Serie (ES1020/ES1040) entwickelt. Mit der medizinischen Datenbank FRIDA sollte ein System geschaffen werden, das neben einer stationären Einrichtung vor allem auch die Poliklinik und ausgewählte Territorien für Dispensairebetreuung und Screening untersuchungen einbezieht. Hier sei auf das EDV-Projekt „Dialyse und Nierentransplantation“ (siehe Abbildung 2) sowie auf das EDV-Projekt „Herzinfarktnachsorge“ verwiesen.

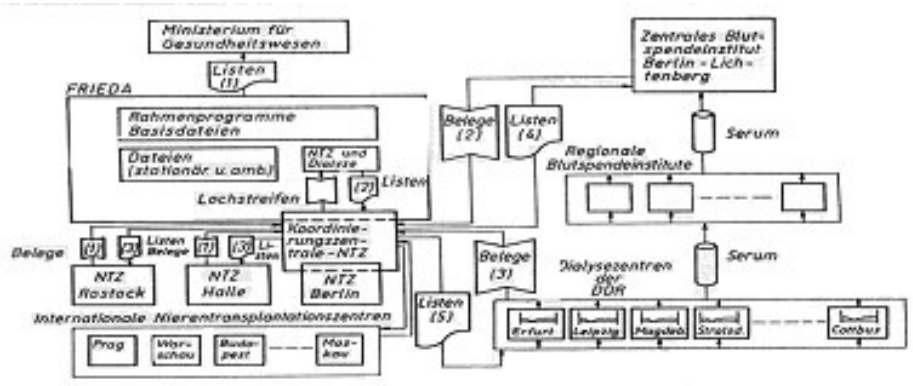


Abbildung 2: Dispensairebetreuung „Dialyse und Nierentransplantation“

Am Forschungsinstitut für Medizinische Diagnostik in Dresden wurde ein Modell zur Gesundheitsversorgung in einem Territorium unter Einbeziehung zentral organisierter Screeningdaten entwickelt [FFG77] [FG77], [FG82]. Nach langem Ringen wurde auch an der Charité, dem Universitätsklinikum der Humboldt-Universität, ein Institut für Medizinische Informatik unter Leitung von H. Weihrauch [WEI82] gegründet. In der Charité war immer schon intensiv auf Gebieten der Medizinischen Informatik gearbeitet worden. Verwiesen sei hier auf Arbeiten zur Laborautomatisierung, zur medizinischen Messwertverarbeitung von J. Michel [MIC82], zur Wissenschaftsinformation am Institut für Wissenschaftsinformation in der Medizin unter Leitung von H. David und später Paul Weiss [WD82], zur Sprachverarbeitung bei der Arztbriefschreibung von R. Steiger. Im Rechenzentrum des Klinikums Berlin Buch unter Leitung von Jürgen Pilgrim [PF83] und zuvor von Jens Reich [FR70], sowie durch die EDV-Einsatzgruppe im Städtischen Klinikum Berlin-Buch u. a. Gruppen [LÄ82], wurde speziell der Einsatz moderner Infomationstechnologien im bio-medizinischen Forschungsprozess vorangetrieben..[2]

[2] In enger Zusammenarbeit mit diesen Einrichtungen konnten die Mitarbeiter des Bereichs Systemgestaltung und automatisierte Informationsverarbeitung der Sektion WTO der Humboldt-Universität theoretische und praktische Erfahrungen in der Informationssystemgestaltung und Softwareentwicklung gewinnen und für die Entwicklung des Konzepts einer „komplexen, nutzerorienten Informationssystemgestaltung [FU89] mit Studenten, Absolventen und Doktoranden des Lehrstuhls praktisch erproben.

4. Auswertung der Konferenzen EDV im Gesundheitswesen und des Kolloquiums: Probleme der Informatik in Medizin und Biologie

Einen guten Überblick über Anwendungen der Informationstechnologien im Gesundheitswesen der DDR erhält man durch das Studium der Proceedings bzw. der Berichte über die Anwenderkonferenz: „EDV im Gesundheitswesen“, die von der Medizinischen Akademie „Carl Gustav Carus“ und dem Organisations- und Rechenzentrum 1973/74 [PE74/75], 1976 [DDS76] in Dresden, vom Institut für Sozialhygiene und Organisation des Gesundheitsschutzes „Maxim Zetkin“ und dem Organisation und Rechenzentrum der Akademie für ärztliche Fortbildung 1982 [FKW83]. und dem Institut für Medizinische Statistik und Datenverarbeitung 1987 [KSW87] durchgeführt wurden.

Die Anwenderkonferenzen „EDV im Gesundheitswesen“ waren der entscheidende Konzentrationspunkt und Stimulus für die Entwicklung der Medizinischen Informatik in der DDR. Im Jahre 1976 wurde immerhin schon die 5. und 1987 die 9. Anwenderkonferenz durchgeführt. Auf der Tagung 1976 [DDS76] z.B. treten die Wissenschaftler auf, die auch in der Folgezeit auf dem Gebiet der medizinischen Informatik in der DDR bestimmend sein werden.[3] Einen guten Überblick geben auch die Beiträge zum Kolloquium: „Probleme der Informatik in Medizin und Biologie“ [FAM82] welches vom 24. bis 26 Januar 1978 an der Humboldt-Universität zu Berlin durchgeführt wurde. Sie zeigen eine ganze Reihe eigenständiger Arbeiten zur Modellierung und Anwendung mathematischer Verfahren beim Einsatz der Informationsverarbeitungstechnik in der biologischen und medizinischen Forschung, Systemanalyse und Modellierung für Leitung, Planung und Prognose sowie IKT-Anwendungssysteme im Gesundheitswesen der DDR. Es wird auch über die Modellentwicklung zur Gesundheitsversorgung im IIASA vorgetragen. Ebenso aufschlussreich ist die Anwenderkonferenz: “EDV im Gesundheitswesen“1982 [FKW83] [4]. Hier wird u. a. über wesentliche Weiterentwicklungen des Krankenhausinformationssystems an der MAD [STR82], über die Entwicklung des Informationssystems der Leitung und Planung im GSW [FIC82] berichtet. Es wird schon eine wesentlich stärkere Orientierung auf die

---

[3] Es sprachen: U. Lochmann, R. Straube über: Patientenbezogene Informationsverarbeitung an der Medizinischen Akademie „Carl Gustav Carus“ – Stand und Einschätzung , W.-D. Grimm, D. Händel, W. Schröder, D. Schuh: über: Einschätzung des Standes und der Ergebnisse der patientenbezogenen Informationsverarbeitung an der Medizinischen Akademie Dresden .- Teil REUD, P. Gudermuth über: Einschätzung des Standes und der Ergebnisse der patientenbezogenen Informationsverarbeitung im Klinikum Berlin-Friedrichshain, G. Steyer, T. Tscharnke über: Einführung eines neuen Krankenblattes im Krankenhaus der Volkspolizei .Ferner: gab es Referate zur automatisierten Informationsverarbeitung in Laboratorien auf der Basis des KRS 4201 sowie einen weiteren Block zu speziellen Anwendungen. Hier sei auf die Ausführungen von G. Enderlein, P. Wulke über: Konzeption der EDV-Auswertung der Arbeitsmedizinischen Tauglichkeits- und Überwachungsuntersuchungen und N. Fichtner, Erfahrungen bei der Anwendung mathematisch-statistischer Verfahren im Zeitraum 1971-1975 im ORZ des Instituts für Sozialhygiene und Organisation des Gesundheitsschutzes verwiesen. Den letzten Block bildeten Vorträge über die Funktion und Bedeutung von Datenbanken für eine patientenbezogene Informationsverarbeitung im Krankenhaus.

[4] Von uns wurde über die Differenzierung verschiedener AIVS-Grundtypen vorgetragen [FK82], wie auch SCAMCY [FU82] und die im TC8 und TC9 der IFIP erarbeiteten Forderung nach Einbeziehung der Arbeits- und Organisationsgestaltung in die Informationssystemgestaltung bekräftigt

Möglichkeiten der Klein- und Mikrorechner für diagnostische Zwecke deutlich. Für eine Beurteilung der Erfolge und Begrenzungen der medizinischen Informatik in der DDR, insbesondere der Gestaltung medizinischer Informationssysteme ist die Anwenderkonferenz: „EDV im Gesundheitswesen“ von 1987 von besonderem Interesse [KSW87]. Schon mit dem Untertitel Mikrorechneranwendungen wird auf den Schwerpunkt der Tagung verwiesen.

Ein besonderer Meilenstein in der Entwicklung der Medizinischen Informatik in der DDR war die IMIA-Working Conference mit dem Titel: „Progress in Biological Function Analysis by Computer Technologies“, die von J. Michel und H. Weihrauch an der Charité in Berlin 1987 organisiert wurde. An dieser Tagung nahm auch der Gründungspräsident des BVMI, Peter L. Reichertz aus Hannover teil. Sie eröffnete Möglichkeiten zum internationalen Wissensaustausch und auch zur Zusammenarbeit mit Medizininformatikern aus der BRD. Dabei wurde zugleich ein beklagenswerter Rückstand der in der DDR verfügbaren Technologien offensichtlich. P. Reichertz meinte dazu, dass dem auch Vorteile abzugewinnen seien, „viel mehr Zeit zum Nachdenken über Anwendungsprobleme aufbringen zu können anstatt mit immer veränderten Technologien.“ Damals erschien dieser Rat eher als wohlgemeinter Trost. Heute wurde klar, dass dieser Rat kein Trost war, sondern das Ergebnis langjähriger Erfahrung.“ [KU04 Die Anzahl der Computeranwendungen in der medizinischen Praxis und im Gesundheitswesen und damit der Bedarf an qualifizierten Medizininformatikern wuchsen rapide. Dies führte zu verschiedenen Formen der Weiterbildung, zur wissenschaftlichen Qualifizierung durch Promotionen. In Kopperation mit den medizinischen Einrichtungen z.B. an der Ingenieur-Hochschule bzw. TU Dresden sowie am Lehrstuhl Informationsverarbeitung der Sektion WTO der Humboldt-Universität[5].

[5] Hier kann nur auf einige der Promotionsverfahren am Lehrstuhl Informationsverarbeitung der Sektion WTO der Humboldt-Universität verwiesen werden:
Dahme,Ch., Methodologische und theoretische Voraussetzungen für die Analyse komplexer Entscheidungssituationen, Dissertation (B), Humboldt-Universität, 1988
Hager, Th., Systemanalytische Methoden und Instrumentarien zur Bewältigung komplexer Entscheidungssituationen in der Wissenschaftsorganisation und bei der Organisation des Gesundheitswesens. Dissertation (B), Berlin: Humboldt-Universität 1985
Hartmann, Ch., Probleme der Realisierung des Informationssystems Wissenschaft und Technik (IWT) Medizin – Dargestellt am Beispiel der Forschungsinformation, Dissertation (A), Humboldt-Universität, 1978
Koitz, K., Die Gestaltung von automatenunterstützten Informationssystemen im Bereich der Medizin – wissenschaftstheoretische und wissenschaftsorganisatorische Ausgangspunkte und Probleme, Dissertation (A), Humboldt-Universität, 1979
Lochmann, U., Zur Gestaltung von medizinischen Informationssystemen – Dargestellt am Beispiel eines Klinikinformationssystems Neurologie/Psychiatrie, Dissertation (A), Humboldt-Universität, 1986
Melzer, J. Zur Problematik automatenunterstützter vielseitiger Auswertung vorhandener Dateien – dargestellt an Beispielen aus der patientenbezogenen Informationsverarbeitung, Diss. (A), Humboldt-Universität, 1986
Pilgrim, J. Benutzer-EDV-Kommunikationsbeziehungen in der Forschung – untersucht im Bereich der Biowissenschaften, Dissertation (B), Humboldt-Universität, 1982
Siedler, A., Zur Entscheidungsvorbereitung bei der Organisation der Dispensairebetreuung, Dissertation (A) Humboldt-Universität, 1986
Seffner, S., Gestaltungsgrundlage einer Informationssprache zur Automaten Indexierung von Sachverhaltsaussagen medizinischer Befund-Texte, Dissertation (A), Humboldt-Universität, 1985
Thoelke, H., Aspekte der Automatisierbarkeit und Automatisierungswürdigkeit von Informationsverarbeitungsprozessen der Dispensairebetreuung, Dissertation (A), Humboldt-Universität, 198

5. Zur Entwicklung der medizinischen Systemforschung in der DDR

Im Institut für Sozialhygiene an der Akademie für ärztlich Fortbildung Berlin unter Leitung von Günter Ewert, sowie an der Humboldt-Universität u. a. Bereichen des Gesundheitswesen wurde an der Entwicklung eines Gesundheitsinformationssystems gearbeitet [EW97]. Das Gebiet der medizinische Systemforschung bzw. Health Care Research (heute Versorgungsforschung) wurde entwickelt. Hierbei geht es um die Anerkennung einer Forschung, bei der das Gesundheitswesen selbst Objekt der Forschung und nicht nur Anwendungspraxis aller anderen Forschungsbemühungen darstellt [EW01]. Dies verlangt nach Methoden zur vertieften Erforschung und verstärkter prognostischer Aussagekraft. Von der Anwendung der Modellmethode kann hier eine entsprechende Unterstützung erwartet werden bei: der inhaltlichen Präzisierung der Grundrichtung der Gesundheitspolitik, der Klärung der Art und Weise sowie Rangfolge der Befriedigung der Bedürfnisse der Bürger nach medizinischer Betreuung und der Begründung möglicher Organisationsformen im Gesundheitswesen, die einen hohen Wirkungsgrad der lebendigen und vergegenständlichten Arbeit sichern helfen. [HA85], [DH87]. International entwickelte sich die Gesundheitssystemforschung (vgl. IIASA-Konferenzen on Systems Aspects of Health Planning [BT75] and on Systems Modeling in Health Care [SH77] und die International Conferences on System Science in Health Care des Instituts für Medizinische Informatik und Systemforschung [EEF84]).[6]

Weckbach, G. Zur Gestaltung eines rechnergestützten Informationssystems in einer territorial geleiteten Einrichtung des Gesundheitswesens, Dissertation (A), Humboldt-Universität 1983

Weiß, P. Probleme, Analysen und Lösungswege der wissenschaftlichen Information und Kommunikation in der Medizin – Unter besonderer Berücksichtigung des Internationalen Zweiginformationssystems MEDINFORM der sozialistischen Länder, Dissertation zum Dr. sc. med, Humboldt-Universität 1977

[6] Über die Zusammenarbeit mit dem Internationalen Institut für Systemanalyse (IIASA) ergaben sich Kontakte zur Johns Hopkins University (Steven Muller, Renate Wilson, Ch. D. Flagle) zum National Institut for Health Care Reseach in Bethesda (Gail Wilenski), zur Veterans Administration (Ruth Dayhoff) und zur University of Maryland (Vincent Brannigan)[BR 83] zum Institut für Medizinische Informatik und Systemforschung in München (W. van Eimeren, R., Engelbrecht] [EEF84] und auch zum Zentrum für medizinische Informatik in Frankfurt/Main. (Bernd Beier, Wolfgang Giere) [BE 86], [GI 95] Es ergaben sich daraus wichtige Erkenntnisse und neue Fragestellungen zur Gestaltung medizinischer Informationssysteme und die Entwicklung der medizinischen Systemforschung in der DDR, sowie für die internationalen Entwicklung auf diesen Gebieten, zu der wir beitragen konnten [FKR84].

6. Gesundheitssystemvergleich in der Systemauseinandersetzung

Es war ein besonderes Anliegen der Arbeit im IIASA Systemvergleiche zu initiieren. Dazu gehörte insbesondere auch der Vergleich der Gesundheitssysteme. Einen ersten Ansatz lieferte das Simulationsmodell: „A Simpel Sick-Leave Model -Used For International Comparison“ von Peter Fleißner et all [FFH80].[7.]

In der Zusammenarbeit mit dem National Institute for Health Cervice Research und der Johns Hopkins beschränkte sich der Vergleich zunächst darauf, dass über die wesentlichen Kennziffern des Gesundheitswesens der DDR entsprechend seiner Selbstdarstellung berichtet wurde. In der amerikanischen Literatur lagen auch schon interessante Vergleiche der Gesundheitssysteme beider deutschen Staaten vor.[8] Sicher gibt es heute kritischere und auch selbstkritischer Analysen des DDR Gesundheitswesens, doch wurde in damaligen Studien, wie auch heute, speziell das System der Polikliniken besonders gelobt. Nach der Wende ergab sich daraus folgende interessante Konstellation. Während es den Verantwortlichen in der BRD vorrangig um die Abwicklung ging, ohne zu fragen, ob es auch Erhaltenswertes gibt, hatten z.B. Wissenschaftler der Hopkins University School of Hygiene and Health Care eine andere Einstellung dazu.[9] Für die „Fifth International Conference on System Science in Health Care“ in Prag wurde ein Beitrag erarbeitet, mit dem Titel: „New European Perspectives: An Agenda for use of Health Service Research – to Restructure and Evaluate Medical

---

[7] Der Vergleich der Krankschreibungen und damit verbundenen ökonomischen Verluste von Österreich, England und der DDR erbrachte, dass Österreich am günstigsten lag, England weniger kurzfristige Krankschreibungen aber mehr langfristige Krankschreibungen hatte und die DDR, durch mehr kurzfristige Krankschreibungen die höchsten Verluste hatte.
Als das Ergebnis publiziert worden war, hätte es beinahe Ärger wegen Verleumdung der DDR gegeben. Es rettet jedoch die Interpretation, die verdeutlichte, dass es offensichtlich einen Zusammenhang zwischen einer höheren Zahl kurzfristiger Krankschreibungen und Vollbeschäftigung d.h. weniger Angst um den Verlust des Arbeitsplatzes gibt.

[8] Doch das Interesse an einem Vergleich der beiden Gesundheitssysteme war nicht so groß, wie man dies, bei diesem besonderen Fall eines sozialen Experiments, hätte vielleicht erwarten können. So schrieb Donald W. Light: “In the extensive literature on political values and health care, no work addresses the divided Germany that was forced into existence following the conclusion of World Ware II: The omission is surprising since the resultant individualistic West and Communist East provide the researcher with a „natural experiment“ in history, one in which a third of the world’s oldest and perhaps most sophisticated health care system itself became an experiment in the application of Marxist/socialist ideology.” [LI85] Er beschreibt weiterhin, dass bei der Entwicklung der Gesundheitssysteme nach dem Krieg auf beiden Seiten auf die Erfahrungen in der Weimarer Republik zurückgegriffen wurde (auch in der sowjetischen Besatzungszone, denn die russischen Kulturoffiziere hatten in der Weimarer Zeit in Deutschland studiert und wollten nun das Gelernte umsetzen). Ein entscheidender Nachteil für das Gesundheitssystem der BRD wäre, dass sie, die von den Nazis abgeschafften Betriebsärzte, nicht wieder eingeführt hätten. Im Gegensatz zur DDR, der dies zum entscheidenden Vorteil wurde. Der Nachteil des staatlichen Gesundheitswesens der DDR wäre vor allem die höhere Bürokratie. Aufgrund der jeweiligen Vor- und Nachteile beider Systeme könnte man, wenn man von Unterschieden z. B. etwa der höheren Säuglingssterblichkeit in der BRD und der höheren Sterblichkeit der Männer in der DDR u. a. Differenzen absieht, eine prinzipiell gleiche Leistungsfähigkeit feststellen.

[9] Die unterschiedliche Haltung wurde auch dadurch deutlich, dass zur gleichen Zeit, als der Bereich Systemgestaltung und automatisierte Informationsverarbeitung, entgegen breiten nationalen und internationalen Protest, mit der Sektion Wissenschaftstheorie und Wissenschaftsorganisation (WTO) abgewickelt wurde, die Johns Hopkins University K. Fuchs-Kittowski zu ihrem assoziierten Mitglied machte.

Care"[WKF92]. Der amerikanische Gesundheitssystemforscher und Begründer dieser Konferenzreihe, Ch. D. Flagle, veranlasste mich, in das Vortragsmanuskript den Hinweis aufzunehmen, dass *das sich jetzt entwickelnde „Brandenburger Modell", den Wandel der früheren Polikliniken zu modernen Gesundheitszentren so ermöglicht, dass die Überwindung der Begrenzung der bisherigen Polikliniken, bei Beibehaltung der Vorteile von Gesundheitszentren erfolgt, dass dieses Modell den schon seit Jahrzehnten propagierten Vorstellungen, wie sie an der Johns Hopkins University entwickelten wurden, entspricht. Hinsichtlich der Nutzung der modernen Informationstechnologien wurde in dieser Arbeit insbesondere auf die Weiterentwicklung und Verbesserung des „record system to monitor cost containment efforts and allow quality of care and efficentcy assesment" orientiert.* In Dresden an der Medizinischen Akademie war die politische Wende verbunden mit der großen Hoffnung auf schnelle Überwindung der offensichtlichen Entwicklungsprobleme der Medizininformatik der DDR. Das auf dem Gebiet der Medizininformatik in Dresden die Integration weitgehend gelang, ist den erbrachten Leistungen, aber vor allem auch dem weitsichtigen Engagement vieler Fachvertreter und den Präsidenten des BVMI und des Präsidenten der GMDS zu danken. Nun, konfrontiert mit der Realität der gegenwärtigen Modernisierung des Gesundheitswesens, bei der die Medizinische Informatik eine bestimmte Rolle zukommt, zeigen sich die Schwierigkeiten des „individualisierten" Gesundheitswesens, wie es Light [LI85]nannte. „Mit der zunehmenden Realität der Strukturen des bundesdeutschen Gesundheitswesen wuchsen in den neuen Ländern neben den Chancen für die Medizinische Informatik aber auch deren Barrieren" [KU04].

Literaturverzeichnis:

[BT75] Bailey, N., Thompson, M. (Editors): System Aspects of Health Planning, Proceedings of the IIASA Conference, Austria, August 20-22, 1974. Amsterdam-Oxford-New York: North-Holland Publishing Company 1975

[BU75] Butrimenko, A. (Editor): Workshop on Data Communications, International Institute for Applied Systems Analysis 2361 Laxenburg, Austria, 1975

[BE86] Beier, B. Rechtsprobleme der Datenverarbeitung in der Medizin, medizinisches Computerrecht, RdV 1986, S. 186ff.

[BR83] Brannigan V., Compensation or Regulation: The Problem of Medical Computer Software, Journal of Consumer Policy, Nr. 6, 1983

[DDS76] Donat, C.D., Dörre, F., Schreiter, D., EDV im Gesundheitswesen, Medizinischen Akademie „Carl Gustav Carus" Dresden, 1976

[DH87] Dahme, Ch., Hager, Th.: Intentional and Operational Aspects of the Decision Behaviour and their Modeling. In: P. Docherty, K. Fuchs-Kittowski, P. Kolm and L. Mathiassen. (Editors): System Design for Human Development and Productivity – Participation and beyond, North Holland, Amsterdam, 1987

[EEF84] Eimeren, W. van, Engelbrecht, R., Flagle, Ch. D. (Editors): Third International Conference on System Science in Health Care, Springer-Verlag, Berlin, Tokyo, et al. 1984

[EW97] Ewert, G., Entwicklungsschritte zu einem nationalen Gesundheitsinformationssystem. Interessengemeinschaft Medizin und Gesellschaft. Berlin 1997.

[EW01] Ewert G., Institut für Sozialhygiene und Organisation des Gesundheitswesen „Maxim Zetkin" (ISOG), Medizin und Gesellschaft. Heft 32, Berlin, 2001

[FFH80] Fleissner, P., Fuchs-Kittowski, K., Hughes, D. J.: A Simple Sick-Leave Model Used For International Comparison, International Institute for Applied Systems Analysis A-2361

[FKW83] Fichtner, N., Keil, W., Weihrauch, H., EDV im Gesundheitswesen – Berichte über die Anwenderkonferenz 1982, Institut für Sozialhygiene und Organisation des Gesundheitsschutzes „Maxim Zetkin“ Berlin, 1983

[FU68] Fuchs-Kittowski, K., Medizin und Datenverarbeitung. In: Zeitschrift für ärztliche Fortbildung 62. Jg. H.4, VEB Gustv Fischer Verlag Jena, 1968, S. 218 – 222

[FR70] Fuchs-Kittowski, K., Reich, J.G., Zur Darstellung von Regulationsprozessen des Zellstoffwechsels auf elektronischen Rechenautomaten.. Rechentechnik Datenverarbeitung, 1. Beiheft 1970, S. 53 - 58

[FRS75] Fuchs-Kittowski, K., Rosenthal, S., Schlutow, G.: Methods to select problems in medicine. In: Noman T.J. Bailey, Mark Thomson (Editors): System Aspects of Health Planning,, North Holland Publishing Company, Amsterdam, et al. 1975, S. 319 - 333

[FFG77] Fuchs-Kittowski, K., Friedemann, H., Gudermuth, P.: Community Screening Systems to Ensure the Data Necessary for a Health Care Model. In: Workshop Modelling Health Care Systems IASA, Laxenburg, Österreich, 1977

[FG77] Fuchs-Kittowski, K., Gudermuth, P.: Providing Data for the Management and Planning of Public Health. In: Shigan, E.N (Editor): System Modeling in Health Care, Proceedings of an IIASA Conference November 22-24, 1977, International Institute For Applied Systems Analysis Laxenburg, Austria, S. 300 – 313

[FKW80] Fuchs-Kittowski, K., Koitz, K., Wendland, B.: On Concepts for Designing Patient Informations Systems (PIS) According to Various User Requirements. In: Linberg, D. A.B., Kaihara, S. (Editors): MEDINFO 80, Third World Conference on Medical Informatics, North-Holland Publishing Company – Amsterdam, et al 1980, S. 1079

[FAM82] Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): III. Wissenschaftliches Kolloquium zur Organisation der Informationsverarbeitung: Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982

[FG82] Fuchs-Kittowski, K. Gudermuth, P., Grundfragen der Informatik in der Medizin und Biologie. In: Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982

[FU82] Fuchs-Kittowski, K.: Information Theory of Organization - A Conceptual Framework For System Design Of Automated Medical Information Systems. In: Blum B.I. (Editor): Proceedings – The Sixth Annual Symposium on computer applications in medical care, IEEE Computer Society, Washington, D.C., 1982

[FK83] Fuchs-Kittowski, K., Koitz, K.: Informatik und Organisationstheorie als konzeptioneller Rahmen für die Analyse und Gestaltung automatenunterstützter Informationssysteme in der Medizin. In: G. Ewert, N. Fichtner (Hrsg.): EDV im Gesundheitswesen, Anwenderkonferenz, 1982, S. 8 - 24

[FKR84] Fuchs-Kittowski, K., Koitz, K., Rudeck, Ch.: Methodological Problems of Dialog Oriented Components in Medical Information Systems. In: Eimeren, W. van, Engelbrecht, R., Flagle, Ch. D. (Editors): Third International Conference on System Science in Health Care, Springer-Verlag, Berlin Heidelberg, et al. 1984, S. 1302 - 1305

[FKW83] Fichtner, N., Keil, W., Weihrauch, H., EDV im Gesundheitswesen – Berichte über die Anwenderkonferenz 1982, Institut für Sozialhygiene und Organisation des Gesundheitsschutzes „Maxim Zetkin“ Berlin, 1983

[GI95] Giere, W.: Medical Information Processing, the BAIK-Model, in E. Fleck, Open Systems in Medicine, 1995, S. 24ff.

[GM82] Gudermuth, P., Manglus, F.: Das automatisierte Patienteninformationssystem Frieda (Ziel und Möglichkeiten der Einbeziehung großer ambulanter Patientengruppen in eine medizinische Datenbank am Beispiel von Frieda). In: Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982, 117 – 126

[HA85] Hager, Th., Systemanalytische Methoden und Instrumentarien zur Bewältigung komplexer Entscheidungssituationen in der Wissenschaftsorganisation und bei der Organisation des Gesundheitswesens. Dissertation (B), Berlin: Humboldt-Universität 1985.

[HA04] Hauben, R.: The Vision of Computer Networking Communication and its Influence on East-West Relations and the GDR. In: Friedrich Naumann, Gabriele Schade (Eds.): Informatik in der DDR - eine Bilanz. GI-Edition Lecture Notes in Informatics, Gesellschaft für Informatik, Köllen Verlag, Bonn 2006, S. 273-28

[KSW87] Kunath, H. Strach, H. , Weihrauch, H. (Hrsg.): EDV im Gesundheitswesen – Berichte über die Anwenderkonferenz 1987 – Mikrorechneranwendungen-, Institut für Medizinische Statistik und Datenverarbeitung,

[KU04] Kunath, H.: Im Spagat von Hoffnung und Realität – Medizinische Informatik in den neuen Bundesländern, Forum der Medizin -Dokumentation und Medizin- Informatik, Sonderheft Nr.1/2004, S. 20-23

[LÄ82] Läuter, J.: Anwendung multivarianter statistischer Verfahren in Medizin und Biologie. In: Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982, S. 173 - 180

[LI85] Light, D. W.: Milbank Memorial Fund Quarterly/Health and Society. Vol. No. 4. 1985, S. 615 ff.

[MGC82] Michel, J., Guráth, B. Camman, H.: Möglichkeiten und Probleme der automatisierten Messwerterfassung und – verarbeitung in der Medizin. In: Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982, S. 215 - 222

[PE74] Penzel, G. (Hrsg.): EDV im Gesundheitswesen, Bericht über den Informationslehrgang 1973 der Medizinischen Akademie „Carl Gustav Carus“ Dresden, 1974

[PE75] Penzel G. (Hrsg.): EdV im Gesundheitswesen, Bericht über den EDV-Informationslehrgang 1974, Medizinische Akademie „Carl Gustav Carus“ Dresden, 1975

[PF83] Pilgrim, J., Fuchs-Kittowski, K.: Interdisziplinäre Funktion der Informationsverarbeitung in der biowissenschaftlichen Forschung. In: H. Parthey, K. Schreiber (Hrsg.) Interdisziplinarität in der Forschung , Akademie-Verlag Berlin, 1983, S. 277 – 301

[SH77] Shigan, E.N (Editor): System Modeling in Health Care, Proceedings of an IIASA Conference, International Institute For Applied Systems Analysis Laxenburg, 1977

[WKF92] Wilson, R., Kasper, J., Fuchs-Kittowski, K.: New European Perspectives: An Agenda for the use of Health Services Research to Restructure and Evaluate Medical Care. M.K. Chytil, G. Duru, W. van Eimeren, Ch.D. Flagle (Editors): Fifth International Conference on System Science in Health Care, Omnipress Publisher, Prague 1992, S. 1512 - 1517

[WD82] Weis, P. David, H.: Automatisierte Informationsverarbeitung in der medizinischen Wissenschaftsorganisation. In: Fuchs-Kittowski, K., Adam, J., Mühlenberg, E. (Hrsg.): Probleme der Informatik in Medizin und Biologie, Akademie-Verlag, Berlin 1982, S. 287- 293

Forschung zur Leistungsmodellierung von Rechnerinstallationen in Rechenzentren der DDR

Gerhard Bergholz

Dorfhainer Straße 26
01189 Dresden
gerhard-bergholz@t-online.de

Abstract: In diesem Beitrag werden einige ausgewählte Ergebnisse auf dem Gebiet der Forschung zur Leistungsmodellierung von Rechnerinstallationen dargestellt, die der Autor und seine Mitarbeiter in den 70er und 80er Jahren erzielt haben. Ausgangspunkt dieser Forschung war die Betrachtung der Rechnerinstallation als Einheit von Arbeitslast und Rechnerkonfiguration. Wissenschaftliche Ergebnisse sind in zahlreichen Veröffentlichungen von mir und meinen Mitarbeitern vorgestellt worden. Dabei standen analytische Bedienungsmodelle, Modelle der ereignisdiskreten stochastischen Simulation, Methoden der Arbeitslast- und Leistungsmessung und Verfahren der Leistungsverbesserung im Vordergrund. Die praktischen Ergebnisse dieser Arbeit fanden ihren Niederschlag in verschiedenen Rechenzentren durch Bereitstellung von Werkzeugen und durch entsprechende Rationalisierungseffekte.

1 Einordnung der Forschung zur Leistungsmodellierung

1.1 Leistungsmodellierung als Forschungsgebiet

Mit der wachsenden Komplexität der Rechnersysteme wurden in den 60er und 70er Jahren, die in den Anlagen ablaufenden Prozesse immer komplizierter, wodurch die Diskrepanz zwischen der Leistungsfähigkeit der rechentechnischen Komponenten und ihrer effektiven Nutzung ständig zunahm. Zur Beseitigung dieses Widerspruchs wurde im internationalen Maßstab das Forschungsgebiet Leistungsmodellierung von Rechnerinstallationen entwickelt.

Die Leistungsmodellierung von Rechnerinstallationen dient vor allem der Verbesserung der Arbeitslastverteilung zwischen den Ressourcen und nicht so sehr der Erhöhung der Ressourcengeschwindigkeiten und Vergrößerung der Speicherkapazitäten. Sie ist in ihren Methoden durch ihre Strukturorientierung weitgehendst unabhängig vom aktuellen Niveau der Computertechnik und verliert auch für die Zukunft bei der Leistungsmodellierung von verteilten Systemen nicht ihre Bedeutung.

Die Leistungsmodellierung von Rechnerinstallationen hat auch eine wesentliche strukturbeschreibende Eigenschaft, die weit über die Leistungsmodellierung hinausgeht. Sie beschreibt für das Ziel der Leistungsbewertung, die Wechselwirkung zwischen Gerätetechnik, Betriebssystem und Anwenderprogrammen unter den Bedingungen ihrer Anwendung als geschlossene Einheit.

1.2 Forschungskollektiv Leistungsmodellierung

Bereits während meiner Tätigkeit in der Industrie bei der Einsatzvorbereitung von Prozessrechnern, habe ich mich mit der Leistungsmodellierung beschäftigt.

Nach meiner Berufung als Hochschullehrer an die Sektion Informationsverarbeitung der TU Dresden im Jahr 1985, konnte ich die während meiner Industrietätigkeit begonnene Forschung auf dem Gebiet der Leistungsmodellierung fortsetzen und ein Forschungskollektiv für die Leistungsmodellierung von Rechnerinstallationen im Wissenschaftsbereich Rechnersysteme dieser Sektion aufbauen. Es zeigte sich, dass der bei der Leistungsmodellierung von Prozessrechnern ausgearbeitete methodische Ansatz auch für die Leistungsmodellierung des Multiprogrammbetriebs in Rechenzentren geeignet war.

Nach meiner Berufung auf den Lehrstuhl Informatik und (in Personalunion) zum Direktor des Rechenzentrums der Bergakademie Freiberg wurde diese Arbeit fortgesetzt, wobei vor allem Doktoranden oder promovierte Assistenten als Mitarbeiter beteiligt waren.

Die vom Kollektiv entwickelten Methoden der Leistungsmodellierung von Rechnerinstallationen haben universellen Charakter für die verschiedenen Anwendungsgebiete der Rechentechnik. Eine Verbindung der Forschung mit dem aktuellen praktischen Rechenbetrieb war aber aus Gründen der Modellbildung und Validierung der Modelle von großer Bedeutung. Nur bei der Orientierung auf die Anwendung, die in der Praxis auch genutzt wird, kann die Arbeitslast festgestellt werden.

Dieser Beitrag hat als Schwerpunkt die Forschungsarbeit für die Leistungsmodellierung des Multiprogrammbetriebs von ESER-Anlagen in Rechenzentren. Die dabei verwendeten Methoden sind von dieser Anwendung unabhängig.

1.3 Rahmenbedingungen in der DDR

Ab Mitte der 70er Jahre fand in vielen Rechenzentren der DDR der Übergang zum Einsatz von Rechnersystemen des ESER statt (s. [Me04]). Damit verbunden war der Übergang zum Multiprogrammbetrieb (Multitasking) in den Rechenzentren.

Die Zahl der Rechenanlagen in der DDR reichte zu keiner Zeit für den vorhandenen Datenverarbeitungsbedarf aus. Das bedeutet jedoch nicht, dass wir uns auf spezielle Lösungen, die allein von der absoluten Ressourcenknappheit ausgingen, konzentriert haben. Unser Anliegen war vielmehr allgemeine Methoden für die verschiedenen Einsatzgebiete der Rechnerinstallationen auszuarbeiten und in den aktuell vorhandenen Anwendungen zu erproben.

1.4 Zusammenarbeit mit Rechenzentren

Im Rahmen der Tätigkeit meines Kollektivs fand eine direkte Zusammenarbeit mit einem Rechenzentrum der Massendatenverarbeitung dem Datenverarbeitungszentrum (DVZ) Dresden (Direktor K. H. Müller) statt. Die Ergebnisse dieser Zusammenarbeit wurden in Berichten von Ingenieurpraktika, Großen Belegen, Diplomarbeiten und Dissertationen niedergelegt. Im Rahmen des Forschungsseminars des Wissenschaftsbereichs Rechnersysteme (Leiter H. Löffler) wurden diese ausführlich diskutiert.

In den unter meiner Leitung im Zweijahresabstand durchgeführten 6 Problemseminaren von 1980 bis 1990 mit der Thematik „Leistungsverbesserung von Rechnerinstallationen“ wurde eine Basis des Erfahrungsaustauschs mit den Mitarbeitern verschiedener Rechenzentren geschaffen. Neben der Darlegung unserer Ergebnisse trugen die Vertreter der Rechenzentren dabei eigene Ergebnisse der anwendungsorientierten Forschungsarbeit vor.

1.5 Forschungskoordinierung

Die Forschung zur Leistungsmodellierung von Rechnerinstallationen wurde im Rahmen der Hauptforschungsrichtung (HFR) „Informationsverarbeitungssysteme“ (Leitung H. Meier) koordiniert. Über die jährlichen Tagungen der HFR und anderen Veranstaltungen entstand ein reger Meinungsaustausch mit Informatikern des Zentrums für Rechentechnik der Akademie der Wissenschaften und verschiedener Hochschulen. Da in dieser HFR auch die Rechnernetzentwicklung koordiniert wurde, hatten wir immer einen Einblick in die Entwicklung künftiger Rechnerkonfigurationen als Rechnernetze.

Ich war in den Jahren 1975 - 1980 der DDR-Vertreter in der Arbeitsgruppe „Statistische Methoden der Bewertung von Betriebssystemen“ der Kommission für Rechentechnik der Akademien der sozialistischen Länder. Die Arbeitsgruppe stand unter der Leitung des ungarischen Vertreters M. Arato. Vom Leiter der Arbeitsgruppe wurden zu diesem Thema jährliche Tagungen in Visegrad mit internationaler Beteiligung organisiert. Die Beratungen der Arbeitsgruppe fanden meist am Rande dieser Tagungen in Visegrad statt.

2 Rechnerinstallation – Einheit von Arbeitslast und Rechnerkonfiguration

Nachdem im ersten Abschnitt dieses Beitrags die von uns durchgeführte Forschung organisatorisch eingeordnet ist, erfolgt in den nächsten Abschnitten eine Darlegung der Methoden und inhaltlichen Ergebnisse dieser Forschung.

2.1 Rechnerinstallation

Eine Rechnerinstallation wird von uns als Rechnersystem während des Rechenbetriebs angesehen und besteht aus der Wechselwirkung zwischen der Rechnerkonfiguration und einer Arbeitslast. Eine Rechnerinstallation stellt eine Abstraktion bezüglich der

gerätetechnischen Komponenten, bezüglich des Betriebssystems, bezüglich der Anwenderprogramme und bezüglich der Nutzereingriffe dar. Die gerätetechnische *Rechnerkonfiguration* umfasst die gerätetechnischen Ressourcen eines Rechnersystems.

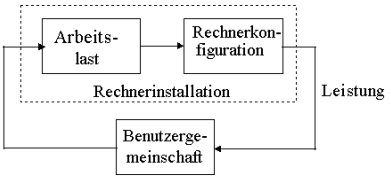


Abbildung 1: Zusammenhang zwischen Arbeitslast und Rechnerkonfiguration

Bei der Betrachtung des Rechenbetriebs ist nicht allein die Rechnerkonfiguration Bestandteil des Systems, sondern auch die Gesamtheit der Programme, Daten und Nutzerkommandos, die seitens der Benutzergemeinschaft während des Rechenbetriebs auf die Rechnerkonfiguration einwirken, gehören dazu. Diese werden zusammenfassend *Arbeitslast* genannt und stellen Forderungen an die Ressourcen der Rechnerkonfiguration dar (s. Abbildung 1).

2.2 Ressourcenorientierte Darstellung der Bedienungsmodelle

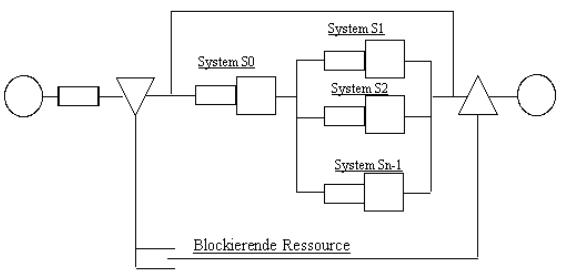


Abbildung 2: Bedienungsnetz für den Multiprogrammbetrieb

Die Wechselwirkung zwischen der Arbeitslast in Form von Forderungen und der Rechnerkonfiguration in Form der Bedienstationen und einer blockierenden Ressource kann durch ein Bedienungsnetz ressourcenorientiert dargestellt werden (s. Abbildung 2). In der Literatur erfolgt die grafische Darstellung der Bedienungsmodelle (Warteschlangenmodelle) meist ressourcenorientiert, weil damit eine gute Grundlage für den Übergang zur analytischen Untersuchung mit Hilfe der Theorie der Markowschen Prozesse vorhanden ist.

2.3 Forderungslaufplan als arbeitslastorientierte Darstellung

Bei der Modellbildung des praktischen Rechenbetriebs ist aber zunächst eine arbeitslastorientierte Darstellung der Bedienungsmodelle erforderlich, bevor zur ressourcenorientierten Betrachtung übergegangen werden kann. Dazu wurde von mir das

Instrument des Forderungslaufplans geschaffen. In der Abbildung 3 ist der Forderungslaufplan eines Forderungsstromes für den Multiprogrammbetrieb mit Berücksichtigung der Kanalbelegung dargestellt. Die arbeitslastorientierte Beschreibung kommt besonders in der zweimaligen Belegung des Kanals zum Ausdruck, was bei einer ressourcenorientierten Beschreibung nicht darstellbar ist.

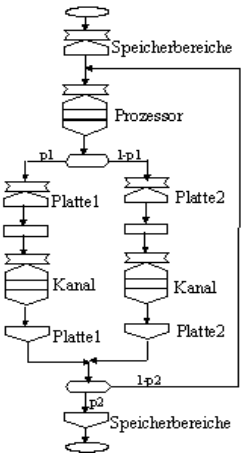


Abbildung 3: Forderungslaufplan für den Multiprogrammbetrieb mit Berücksichtigung der Kanalbelegung

Bei der Analyse von Betriebssystemen wurden in der Literatur Petrinetze angewendet. Dabei wurden verschiedene Erweiterungen des Petrinetzes für die Leistungsmodellierung vorgenommen. In [Be00] wurde gezeigt, wie sich ein GSPN (Generalized Stochastic Petri Net) in einen Forderungslaufplan überführen lässt und umgekehrt. Der Vorteil des Forderungslaufplans gegenüber dem GSPN liegt in der durch die Betrachtung als Wechselwirkung von Forderungen und Ressourcen vorhandene Problemnähe, die beim GSPN verloren geht.

Die Unterteilung der Rechnerinstallation in Forderungen und Ressourcen und die Nutzung des Forderungslaufplans ist auch heute bei verteilten Systemen nicht überholt. Die Methode des Forderungslaufplans war in unseren Untersuchungen ein wichtiges Mittel für die Modellbildung. Sie wurde in allen Abschnitten der Leistungsmodellierung einschließlich der Arbeitslast- und Leistungsmessung genutzt.

3 Arbeitslastmodell

Neben der Entwicklung von Leistungsverhaltensmodellen ist die Arbeitslastanalyse eine wichtige Aufgabe bei der Leistungsmodellierung. Dazu ist ein Arbeitslastmodell erforderlich.

Wir können hier zwei Modellschichten unterscheiden: die Modellschicht der geteilt genutzten Ressourcen und die Modellschicht der (jeder Forderung) fest zugeordneten Ressourcen. Dabei handelt es sich bei der Modellschicht der geteilt genutzten

Ressourcen meist um *Prozessoren* (auch E/A-Prozessoren) und bei der Modellschicht der fest zugeordneten Ressourcen meist um *Arbeitsspeicher*.

Als Arbeitslastkenngrößen wurden für die geteilt genutzten Ressourcen die *Beschäftigungskoeffizienten* x_i (i=1,...,n) eingeführt, die den Anteil der Arbeitslast angeben, der auf die entsprechende Ressource fällt. Das Arbeitslastmodell der geteilt genutzten Ressourcen stellt einen Zusammenhang zwischen den Operationszeiten T_{ok} der einzelnen geteilt genutzten Ressourcen und den Beschäftigungskoeffizienten dieser Ressourcen her.

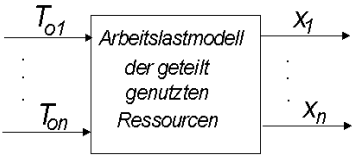


Abbildung 4: Arbeitslastmodell der geteilt genutzten Ressourcen

In der Abbildung 4 ist der Zusammenhang zwischen den Operationen und den Beschäftigungskoeffizienten der geteilt genutzten Ressourcen für das Arbeitslastmodell der geteilt genutzten Ressourcen dargestellt. Die Operationszeiten können aus den gerätetechnischen Daten der einzelnen Ressourcen gewonnen werden. Für den Verarbeitungsprozessor wurde hier der ESER-Mix (s. [SEWI75]) genutzt. In [HALO82] wurde die Erstellung von Befehlsmixen für SKR-Einsatzfälle aus Messungen vorgestellt. Für die Wechselplattenspeicher hängt die Positionierzeit vom Zylinderabstand zwischen zeitlich benachbarten Plattenoperationen und damit von der Anordnung der Dateien auf den Plattestapeln ab. Mit Hilfe des Forderungslaufplanes können die Beschäftigungskoeffizienten aus den Operationszeiten der einzelnen Geräte bestimmt werden.

Als komprimierte Arbeitslastkenngröße für die festzugeordneten Ressourcen wurde der *Multiprogrammfaktor M* eingeführt, der die Parallelität eines zwischen den Forderungen angibt.

4 Entwicklung analytischer Bedienungsmodelle

4.1 Einfacher Ansatz für ein dimensionsloses Leistungsverhaltensmodell

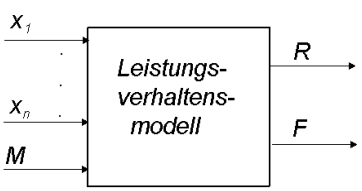


Abbildung 5: Ansatz für ein dimensionsloses Leistungsverhaltensmodell

Aus der Physik ist bekannt, dass die dimensionslose Betrachtung die Allgemeinheit von Modellen erhöht. Bereits bei der Einführung der Arbeitslastkenngrößen wurden dimensionslose Größen betrachtet. Für den Multiprogrammbetrieb wurden die beiden dimensionslosen Leistungsbewertungsgrößen *Leistungsfaktor R* und *Verweilzeitfaktor F* eingeführt. Der Leistungsfaktor gibt an, wie sich die Leistung eines Forderungsstapels durch den Multiprogrammbetrieb erhöht und der Verweilzeitfaktor zeigt um wie viel sich die Verweilzeit einer Forderung gegenüber der Summenbedienzeit vergrößert.

Die mathematische Beschreibung der Rechnerinstallation als Zusammenhang zwischen den Arbeitslastkenngrößen und den Leistungsbewertungsgrößen wurde *Leistungsverhaltensmodell* genannt. In der Abbildung 4 ist als Ansatz für ein dimensionsloses Leistungsverhaltensmodell der Zusammenhang zwischen den dimensionslosen Arbeitslastkenngrößen und den Leistungsbewertungsgrößen dargestellt.

4.2 Anwendung der Theorie der Markowprozesse

Mit Hilfe der Bedienungstheorie wurde von uns eine Reihe analytischer Bedienungsmodelle entwickelt und untersucht, die den Zusammenhang zwischen den Arbeitslastkenngrößen und den Leistungsbewertungsgrößen herstellen.

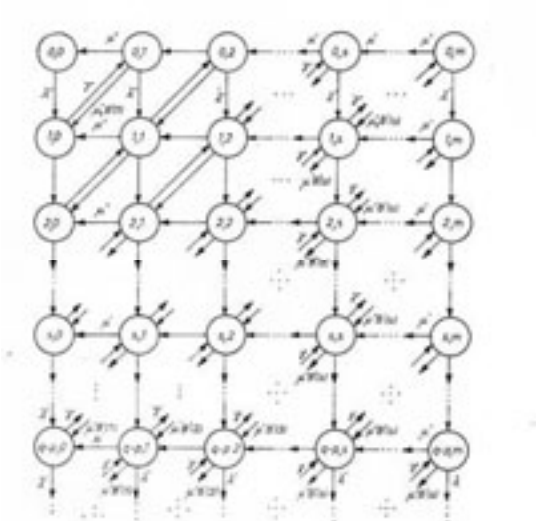


Abbildung 6: Zustandsgraph für das Bedienungsnetz mit Blockierung [Be89]

Im Rahmen der Bedienungstheorie wurde die Theorie Markowscher Prozesse genutzt. Dazu musste ein Zustandsgraph aufgestellt werden. In der Abbildung 6 ist der *Zustandsgraph* für ein Einklassen-Bedienungsnetz mit Blockierung angegeben. Ausführliche Untersuchungen von Einklassen- und Mehrklassen-Bedienungsmodellen vom Central-Server-Typ wurden in [Le82] durchgeführt. Es wurde von ihm dazu das Werkzeug KOMPLEX (s. [BL88]) geschaffen. Die Ergebnisse der Untersuchungen fanden ihren Niederschlag in der Dissertation A [Le83] und der Dissertation B [Le89] von D. Lehmann.

4.3 Analytisches Zweischichtenmodell

Für das Leistungsverhaltensmodell einer Rechnerinstallation wurde der entsprechende Zustandsgraph in die beiden Bereiche der geteilt genutzten Ressourcen und der festzugeordneten Ressourcen, die wesentlich unterschiedliche Übergangsraten haben, eingeteilt und in die Form eines schichtenförmigen Aufbaus gebracht [Be86]. Das wird in der Abbildung 7 mit den Schichten A, B schematisch dargestellt.

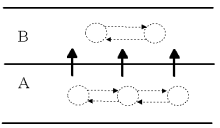


Abbildung 7: Zum Zweischichtenaufbau des Zustandsgraphen

Dabei wurde für die untere Modellschicht A die Zustandsgraphen einer Schar geschlossener Bedienungsnetze mit den geteilt genutzten Ressourcen als Bedienstationen erhalten. In der Abbildung 8 ist ein geschlossenes Bedienungsnetz als Teil der unteren Modellschicht dargestellt. Die einzelnen Bedienungsnetze unterscheiden sich durch die Zahl der in jedem Netz aktiven Forderungen (k=1,2,...,n).

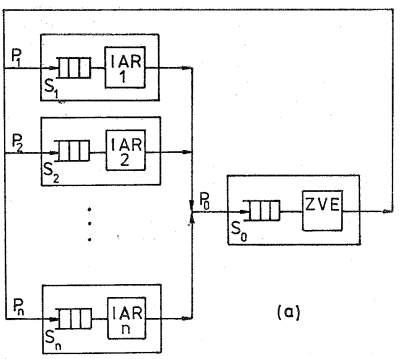


Abbildung 8: Geschlossenes Central-Server-Bedienungsnetz

Für die obere Modellschicht B wurde ein Zustandsgraph erhalten, der für ein virtuelles Mehrkanal-Bedienungssystem (s. Abbildung 9) gilt. Diese Modellschicht wird Modellschicht der fest zugeordneten Ressourcen genannt, wobei der Mehrkanalbedienungsknoten die fest zugeordneten Ressourcen repräsentiert.

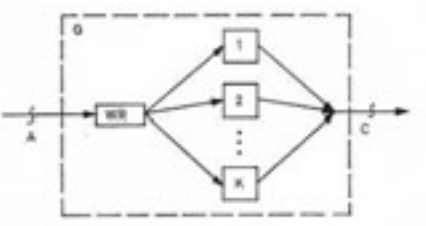


Abbildung 9: Virtuelles Mehrkanalbedienungssystem für die obere Modellschicht

5 Entwicklung eines hybriden Zweischichtenmodells

Eine wichtige Ergänzung der analytischen Modellierung durch die Simulation war in der oberen Schicht, die Einbeziehung der Parallelität innerhalb eines Auftrages in einem Jobnetz.

Für die ereignisdiskrete stochastische Simulation wurde das von ROBOTRON bereitgestellte Programmiersystem SIMDIS genutzt, das mit der GPSS-Sprache programmiert werden kann. Im Rahmen der Forschungsarbeiten des Kollektivs wurden umfangreiche Simulations-Untersuchungen durchgeführt (s. z. B. [HM82]). Der Versuch ein größeres praxisrelevantes Zweischichtenmodell durch die Simulation direkt nachzubilden scheiterte am zu großen Rechenzeitaufwand. Die großen Unterschiede der Übergangsraten zwischen den Schichten, ließen die vollständige Simulation des Zweischichtenmodells scheitern.

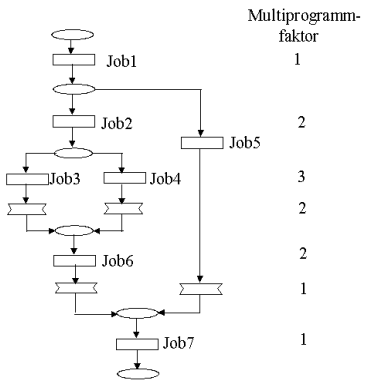


Abbildung 10: Einfaches Simulationsmodell für ein Jobnetz

Aber da die Simulation vor allem für die obere Schicht notwendig war, bot sich als Ausweg die hybride Modellierung des Zweischichtenmodells an. Wenn man die untere Modellschicht als Schar geschlossener Bedienungsnetze analytisch nachbildet und in der oberen Schicht die Simulation durchführt, ist das Rechenzeitproblem gelöst. In diesem Fall muss nur mit der niedrigen Frequenz der oberen Modellschicht die Simulation durchgeführt werden (s. Abbildung 10). Der zusätzliche Aufwand für die analytische Berechnung der Schar geschlossener Bedienungsnetze hatte dabei kein großes Gewicht. So wurde in ein solches hybrides Zweischichtenmodell für die Jobablaufplanung entwickelt und untersucht. In der Dissertation A von G. Mrosko (s. [Mr85]) wurden die Forschungsergebnisse auf diesem Gebiet zusammenfassend dargestellt.

6 Arbeitslast- und Leistungsmessung

Die standardmäßig bei ESER vorhandenen Messmittel genügten nicht allen Anforderungen, die sich aus der Leistungsmodellierung ergaben. Für die Bestimmung der Beschäftigungskoeffizienten des Leistungsverhaltensmodells war zusätzlich die Messung der E/A-Bedienungszeiten erforderlich. Im Rahmen der Zusammenarbeit mit

dem DVZ Dresden wurde dafür ein Leistungsmonitor EAMON für OS/ES entwickelt, implementiert (s. [De83], [De86]) und genutzt. Die Ergebnisse der Arbeit zu EAMON wurden in der Dissertation A von M. Dettke die er vor der Fakultät für Elektrotechnik/Elektronik der TU Dresden 1987 verteidigte, zusammenfassend dargestellt. Der Leistungsmonitor EAMON wurde in enger Zusammenarbeit mit L. Zipfel als Grundlage für die Leistungsverbesserung durch Umverteilung der Betriebssystemdateien genutzt (s. Abschnitt 8.2).

7 Synthetisches Arbeitslastmodell

Von M. Müller (s. [BM86]) wurde ein synthetisches Arbeitslastmodell SAMOS für das Betriebsystem OS/ES entwickelt. Im Gegensatz zur passiven Leistungsmessung im Nutzrechenbetrieb, war mit diesem Modell eine Möglichkeit zur aktiven Erzeugung einer Arbeitslast, bestehend aus synthetischen Programmen mit gewünschten Arbeitslastkenngrößen, vorhanden. Unter Nutzung von vorgegebenen Arbeitslast-kenngrößen wurden Arbeitslasten mit vorgegebenem Profil, für die Realisierung in einer Rechnerinstallation des ESER in Form von synthetischen Jobs erzeugt und anschließend abgearbeitet. Dabei wurde auch die Erzeugung von abhängigen Jobs, die in einem Jobnetz abgearbeitet werden, in das Modell einbezogen. Die Arbeiten zu diesem Thema fanden ihren Abschluss in der Dissertation A von M. Müller (s. [Mü87]).

8 Leistungsverbesserung

Die Forschung zur Leistungsmodellierung sollte nicht nur neue Erkenntnisse und Werkzeuge zur Leistungsmodellierung schaffen. Für den praktischen Rechenbetrieb sollte sie auch direkte Leistungsverbesserungen realisieren.

8.1 Leistungsverbesserung durch Umverteilung der Betriebssystemdateien

Durch die E/A-Operationen der Wechselplatten wird ein wesentlicher Anteil der Arbeitslast erzeugt. Dabei wird davon ausgegangen, dass die meisten E/A-Zugriffe zu den Betriebssystemdateien erfolgen. Unter den Bedingungen der Massendatenverarbeitung beträgt der Anteil Zugriffe zu den Systemdateien mehr als 50%. Als Maßnahme zur Leistungsverbesserung bot sich hier eine Umverteilung der Systemdateien über die Plattenstapel an (s. [Be84]). Ziel dieser Umverteilung war die Verringerung der Positionierzeit der Systemdatenzugriffe. Die entsprechenden Algorithmen zur Optimierung der Systemdateien wurden im Werkzeug OPDA [BZ86] implementiert und im DVZ Dresden genutzt. Die Ergebnisse der Arbeit zu diesem Thema fanden Eingang in die Dissertation A von L. Zipfel (s. [Zi87]).

8.2 Jobnetzoptimierung

In den Projekten der Massendatenverarbeitung der DVZ wurde von den Datenverarbeitungszentren für eine Reihe von Projekten, das periodisch zu verarbeiten

waren, ein Jobnetz aufgebaut (s. Abbildung 10), das durch ein erweitertes Betriebssystem automatisch abgearbeitet wurde. Die im Projekt vorhandenen Jobabhängigkeiten wurden dabei berücksichtigt. Über die optimale Gestaltung der Jobnetze wurde eine Maximierung des Leistungsfaktors erreicht. Bei der Jobnetzoptimierung wurden die bekannten mathematischen Verfahren der Netzplantechnik genutzt (s. [Sc84]).

9 Breitenanwendung und Nachhaltigkeit der Ergebnisse

Die Leistungsmodellierung von Rechnersystemen ist eine wichtige Grundlage für die Leistungsverbesserung dieser Systeme. Dabei war die hier betrachtete *Leistungsmodellierung von Rechnerinstallationen* nicht auf die Erhöhung der Ressourcengeschwindigkeiten und verfügbaren Speicherkapazitäten, sondern auf die Verbesserung ihrer Struktur gerichtet.

Eine umfassende Analyse, die alle Aspekte von Arbeitslast und Ressourcen enthält, war nur beim praktischen Betrieb einer Rechnerinstallation möglich. Dabei setzte die Erfassung repräsentativer Arbeitslasten die Analyse von Rechnerinstallationen voraus, die sich im Breiteneinsatz befanden. Das war, neben den Zielen der unmittelbaren Leistungsverbesserung, der wichtigste Grund, warum wir damals die Hauptrichtung unserer Forschung auf den praktischen Rechenbetrieb in Rechenzentren orientiert haben.

Aus Gründen der Nachhaltigkeit galt dabei die Vorrausetzung, dass die strukturellen und mathematischen Ansätze so getroffen werden mussten, dass ihre Anwendbarkeit sich nicht nur auf den Rechenbetrieb in Rechenzentren beschränkte. Die von mir vorher für die Leistungsmodellierung von Prozessrechner und später zur Leistungsmodellierung von parallelen und verteilten Rechnersystemen durchgeführten Arbeiten, haben gezeigt, dass die bei der Leistungsmodellierung der Rechnerinstallationen in den Rechenzentren verwendeten strukturellen und mathematischen Ansätze für die Modellbildung bei einer breiten Klasse von Rechnerinstallationen mit Rechnernetzkonfigurationen anwendbar waren und sind.

Literaturverzeichnis

[BL88] Bergholz, G.; Lehmann, D.: The KOMPLEX Performance Prediction Tool. Syst. Anal. Model. Simul. 5 (1988)3, 243-251.

[BM86] Bergholz, G.; Müller, M.: Synthetische Arbeitslastmodelle. edv-aspekte 4 (1986) s. 60-64.

[Be79] Bergholz, G.: Zur Leistungsanalyse von Rechnersystemen der Stapelverarbeitung. Problemseminar „ Analyse von Rechnersystemen und der Kommunikation in komplexen Informationsverarbeitungssystemen“, Weißig (1979)

[Be80] Bergholz, G.: Zur Leistungsanalyse und Leistungsverbesserung im praktischen Rechenbetrieb. Problemseminar „ Leistungsverbesserung von Rechnerinstallationen“, Weißig (1980)

[Be84] Bergholz, G.: Leistungsbewertung im Rechenbetrieb. rechentechnik/datenverarbeitung 21 (1984) S.10-13.

[Be86] Bergholz, G.: Modellierung Einer EDVA durch ein Bedienungssystem mit virtuellem Bedienungsknoten. Elektron. Inf.verarb. Kybern. EIK 22 (1986) 4 167-177.

[Be89] Bergholz, G.: Leistungsmodellierung von Rechnersystemen. Akademieverlag Berlin (1989) 223 Seiten, 102 Abbildungen und 31 Tabellen.

[Be00] Bergholz, G.: Zur graphischen Darstellung von Modellstrukturen im Rahmen der Leistungsmodellierung. ASIM-Workshop Multiagentensystems and Individual-based Simulation Würzburg 3 (2000)

[BR92] Bergholz, G.; Richter, K.: Performance evaluation of computer networks with blockings. Computional Systems Analysis, A. Sydow (Editor) Elsevier Science Publishers , Amsterdam (1992)

[BZ86] Bergholz, G.; Zipfel, L.: OPDA – Optimale Dateiverteilung auf externen Geräten. edv-aspekte 4 (1986) S. 62-64

[De82] Dettke, M.: Leistungsmessung mittels Softwaremonitor EAMON. Problemseminar „Leistungsverbesserung von Rechnerinstallationen“, Weißig (1982)

[De86] Dettke, M.: Leistungsmessung und –bewertung unter Einsatz empirischer Modelle. Problemseminar „ Leistungsverbesserung von Rechnerinstallationen“, Weißig (1986)

[HL82] Hamann, C.-J.; Lotzkat, O.: Erstellung von Befehlsmixen für SKR-Einsatzfälle. Problemseminar „ Leistungsverbesserung von Rechnerinstallationen“, Weißig (1982)

[HM28] Hamann, Cl.-J.; Mrosko, G: Simulation von Verhaltensmodellen zur Leistungsbewertung von Rechnerkonfigurationen. 27. Intern. Wiss. Koll. TH Ilmenau (1982)

[Me07] Merkel, G.: Computerentwicklung in der DDR – Rahmenbedingungen und Ergebnisse. GI –Edition Friedrich Naumann, Gabriele Schade (Hrsg.) Informatik in der DDR – eine Bilanz Tagungsband zu den Symposien 7. –9. Oktober 2004 in Chemnitz 11.-12. Mai 2006 in Erfurt. S. 40-54

[Le82] Lehmann, D.: „Central-Server“- Modelle – Mittel zur Leitungsvorhersage. Problemseminar „Leistungsverbesserung von Rechnerinstallationen“, Weißig (1982)

[Le83] Lehmann, D.: Analytische Verhaltensmodelle zur Leistungsbewertung im praktischen Rechenbetrieb. Dissertation A. Fakultät für Elektrotechnik/Elektronik TU Dresden (1983)

[Le89] Lehmann, D.: Leistungsmodellierung von Rechnerinstallationen – Stand – Tendenzen – Ein verallgemeinerter Zugang über analytische Verfahren. Dissertation B. Fakultät für Elektrotechnik/Elektronik TU Dresden (1989)

[Mr85] Mrosko, G.: Untersuchung der Jobablaufsteuerung mit Hilfe hybrider Modelle. Dissertation A. Fakultät für Elektrotechnik/Elektronik der Tu Dresden (1985)

[Mü87] Müller, M.: Ein synthetisches Arbeitslastmodell als Mittel der zur Leistungsverbesserung von Rechnersystemen. Dissertation A. Fakultät für Elektrotechnik/Elektronik der TU Dresden (1987)

[SW75] Seidel, G.; Wiedemuth, D.: ESER - Funktionsprinzipien des Einheitlichen Systems der Elektronischen Rechentechnik. Schriftenreihe Informationsverarbeitung. Verlag Die Wirtschaft, Berlin (1975)

[Sc84] Schreiber, E.: Programme zur Leistungsbewertung im DVZ Dresden. rechentechnik/datenverarbeitung 21 Heft 1 (1984) S. 8-9

[Zi87] Zipfel, L.: Optimale Dateiverteilung auf Direktzugriffsdatenträger – Ein Beitrag zur Erhöhung der Leistungsfähigkeit von Datenverarbeitungsanlagen. Dissertation A. Fakultät für Elektrotechnik und Informatik der TU Dresden (1987)

Die Verarbeitung unformatierter Daten am Beispiel der Beschreibung von wissenschftlich-technischen Dokumenten

Harald Löhr

Zöllmener Straße 5
01157 Dresden
loehr@dd.sda.de

Abstract: Wissenschaftlich-technische Dokumente wie Zeitschriftenartikel, Patente und Bücher waren schon lange Gegenstand von Informationsmitteln wie Kataloge und kumulierte Inhaltsverzeichnisse. Im 20. Jahrhundert wuchs die Flut an Dokumenten und damit die Aufgabe, effektiver über ihren Inhalt zu berichten. Nach der weltweiten Anwendung der Dezimalklassifikation als Ordnungsmittel für Kurzreferate in Karteikartenform und von meist fachgebietsbezogenen Referateblättern stand von den 1960-er Jahren an die rechnergestützte Informationsverarbeitung für die Aufgaben der Information und Dokumentation zur Verfügung.

Bereits mit Magnetbändern konnten Dokumentennachweise effektiv gespeichert und rechnergestützte Informationsmittel preisgünstig und schnell bereitgestellt werden. Für die Rechenanlage R300 wurde ein Typenfeinprojekt für die Information und Dokumentation bereitgestellt und DDR-weit angewendet. Das AIDOS war die Produktreihe für die Folge-Rechnerfamilien.

Am Beispiel von Rechnerpraktika auf der Basis dieser Softwaresysteme wurden bereits seit den 1970-er Jahren Verfahren zur effektiven Verarbeitung unformatierter Daten in die Informatik-Ausbildung einbezogen.

1 Speicher für wissenschaftlich-technische Dokumente - historischer Abriss -

1.1 Primär- und Sekundärdokumente. DK-Zahlen

Als **Primärdokument** wollen wir hier ein Dokument verstehen, in dem über einen (neuen) Sachverhalt, Zusammenhang o. ä. berichtet wird.

Ein **Sekundärdokument** ist ein Dokument, das über ein anderes Dokument berichtet, darüber hinaus aber keinen neuen Sachverhalt vermittelt.

Unter dem Oberbegriff wissenschaftlich-technische Primärdokumente wurden zuerst schriftlich fixierte Dokumente verstanden, die sachlich-fachlichen Inhalt haben und niedergelegt wurden, um die Mitwelt und die Nachwelt über die Gedanken des Verfassers zu informieren und ggf. auch einen Urheberanspruch zu begründen. Ohne Anspruch auf Vollständigkeit sind das Zeitschriftenartikel, Bücher, Patentschriften, Prospekte, Kurzmitteilungen, Gesetze und Standards, aber auch Gerichtsurteile und Festlegungen aller Art.

Dokumenten größeren Umfangs, wie z. B. Fachbücher und nicht alphabetisch geordnete Nachschlagewerke, wurde schon seit langem eine spezielle Ergänzung hinzugefügt, mit der man zu Einzelinformationen im Buch zugreifen kann: das Register/der Index. Diese 'Schlagwortliste' findet man (wenn vorhanden) meist am Ende des Werkes. In diesem Verzeichnis von Fachbegriffen sind zu jedem enthaltenen Stichwort die Nummern der Seiten genannt, auf denen der Fachbegriff (signifikant) auftritt. Der Gestalter des Registers legte fest, welche Inhaltskomponenten er für registerwürdig ansah. Gegenwärtig unterstützen Textverarbeitungssysteme das Registermachen technisch - die Entscheidung über die Aufnahme eines Wortes fällt aber in jedem Einzelfall der Mensch! Automatisierte Verfahren, die z. B. die Häufigkeit eines Fachbegriffs als Kriterium für die Aufnahme ins Register verwenden, verschieben das Problem nur auf eine andere Ebene, denn die Programme, die die Entscheidung treffen, werden auch von Menschen entworfen.

Mit dem Register wurde die zweistufige Recherche geboren. Einstufig soll hier die sequentielle Suche nach einer relevanten Textstelle im Gesamttext (Seite für Seite) bedeuten. Zusätzlich bietet ein Register die Möglichkeit, zu einem Fachbegriff relevante Seitennummern zu finden (Stufe 1) und dann zum gesuchten Kontext auf den in der Liste vermerkten Seiten zuzugreifen (Stufe 2). Die Register gestatten nur den Zugriff auf relevante Textseiten innerhalb eines Werks.

Seit etwa 200 Jahren nahm die Anzahl der wissenschaftlichen Zeitschriften nahm ständig zu und damit die Schwierigkeit, zu den Primärdokumenten meist geringeren Umfangs zuzugreifen, die in den Zeitschriftenbänden/-jahrgängen abgelegt waren.

Die Hauptaufgabe der Weiterarbeit mit den gespeicherten **Primärdokumenten** besteht darin, die zu einer bestimmten Fragestellung relevanten Dokumente wieder zu finden. In den ersten Jahrhunderten des wissenschaftlichen Austauschs waren dazu die persönlich formulierten Briefe zu analysieren und die entsprechenden Zeitschriften gezielt zu durchsuchen. Welche Verluste bei dieser Rechercheart entstehen, ist u. a. seit der Nichtveröffentlichung vieler Erkenntnisse von Carl Friedrich Gauss selbst oder seit der (teilweisen) Vernichtung der Aufzeichnungen von Bernhard Riemann bekannt.

Aus diesen Gründen spielte die Veröffentlichung der Arbeitsergebnisse in wissenschaftlich-technischen Zeitschriften und Jahrbüchern, also in Periodika, eine hervorragende Rolle. Schnell wurde das so gespeicherte Wissen unübersehbar: Es musste aufbereitet werden. Deshalb entstanden verschiedene Formen von **Sekundärliteratur**, deren Inhalt nicht in originär neuem Wissen besteht, sondern darin, auf Originalquellen hinzuweisen. Der wichtigste Typ eines Sekundärdokuments sind durch Fachleute verfasste Kurzfassungen, so genannte Kurzreferate von 10 - 20 Zeilen á ca. 40 Zeichen. Sie sind außer in gebundener Form gut auf Karteikarten im A6-Format unterzubringen. Damit war eine neue Aufgabe entstanden: Die Sekundärdokumente mussten für eine erfolgreiche Suche sinnvoll gespeichert werden.

In der Mitte des 20. Jahrhunderts waren weltweit zwei Hauptformen von Speichern für Kurzreferate verbreitet. Es waren nach der international standardisierten universellen Dezimalklassifikation (UDK) geordnete **Karteikartenspeicher** und **Referatzeitschriften** mit einer jeweils eigenständigen fachlichen Gliederung des Fachgebiets.

Wichtigstes Kennzeichen einer **Referatekarte** war die DK-Zahl, die das Referat einem Fachgebiet zuordnet. Bei mehreren Fachgebieten mussten Duplikate der Karte angefertigt werden, damit das Sekundärdokument unter allen relevanten Fachgebieten wieder gefunden wird.

Das Prinzip einer **Dezimalklassifikation** besteht darin, dass das gesamte Wissen der Menschheit in Gruppen unterteilt wird, deren jede durch eine Dezimalziffer von 1 bis 9 gekennzeichnet wird. Jede Gruppe wird nun bei Bedarf in 9 Untergruppen unterteilt. Die Unterteilung kann beliebig oft fortgesetzt werden, bis die notwendige Feinheit der Fachgebiete erreicht ist. Nach jeweils 3 Ziffern wird als Gliederungshilfsmittel ein Punkt eingefügt. DK-Zahlen von 15 und mehr Ziffern sind keine Seltenheit, weil manches Wissensgebiet bei der Erst-Definition des Systems der DK-Zahlen in seinem künftigen Umfang nicht richtig eingeschätzt werden konnte. Das trat besonders bei den Naturwissenschaften und in der Technik auf. Im praktischen Einsatz orientierte man darauf, hinter der Leitkarte für eine DK-Zahl zwischen 5 und 30 Referatekarten stehen zu haben, um leicht zu den Einzelkarten zugreifen zu können. Wurde diese Richtzahl bleibend überschritten, dann musste die nächste Untergliederungsebene eröffnet werden. Automatisch ergab sich dadurch die notwendige Feingliederung der DK-Zahlen in einer Informationsstelle. Einige Beispiele für DK-Zahlen sind in Tabelle 1 zusammengestellt. Man erkennt gut die unterschiedliche Größe der durch die ausgewählten vierstelligen DK-Zahlen repräsentierten Wissensgebiete.

Tabelle 1 Beispiele für DK-Zahlen nach [De05]

DK-Zahl	Bedeutung
621	Angewandte Physik
621.3	Elektro-, Magnettechnik, technische Optik, Kommunikations-, Computertechnik, Elektronik, Lichttechnik
621.381	Elektronik
621.381.528.7	Thyristoren
621.39	Computertechnik
621.4	Antriebsmotoren und Wärmetechnik
621.6	Gebläse, Lüfter, Pumpen

Parallel dazu, dass die Abteilungen der Speicher für Sekundärinformationen immer größer und damit schlechter überschaubar wurden, ertönte zunehmend die Forderung danach, nicht nur nach Fachgebieten, sondern auch außerhalb von Fachbüchern nach Inhaltskomponenten der Dokumente suchen zu können.

Die Abhilfe stellten vorerst fachbezogene Bibliographien dar, in denen für vorher bekannten Nutzerbedarf relevante Auszüge aus Referatespeichern zusammengestellt wurden. Der Begriff Bibliographie wurde hier über seinen Ursprung im Bibliothekswesen hinaus erweitert.

Als ergänzender Schritt wurde dann das Prinzip der Register für eine neue Speicherform verwendet und erweitert. So wurde die Forderung an die Recherche als Suche nach für eine Aufgabe relevanten Dokumenten über ihre Inhaltskomponenten erfüllt. Dabei entstand die **Sichtlochkartei**.

Sie bildete die erste Möglichkeit zur freien Recherche nach relevanten Dokumenten - nicht nur nach Textstellen in *einem* Gesamtdokument - entsprechend der gekennzeichneten Inhaltskomponenten.

1.2 Sichtlochkartei

Diese Speicherungsform für Informationen war nicht nur originell, sondern begründete auch eine wissenschaftlich fundierte Recherchestrategie. In diesem Sinn ist die Sichtlochkartei eine direkte Vorstufe der rechnergestützten zweistufigen Informationsrecherche über Inhaltskomponenten. Das sah anfangs allerdings niemand so deutlich.

Das **Prinzip der Sichtlochkartei** ist denkbar einfach. Zu jedem Fachbegriff (Deskriptor) wird eine Sichtlochkarte angelegt, in die an den entsprechenden Positionen der Nummern der Primärdokumente, für die der Deskriptor vergeben wurde, ein Loch gestanzt wird. Sucht man nun alle Dokumente, für die der Deskriptor 'Recherche' als bedeutsam erklärt wurde, so greift man die Sichtlochkarte zu diesem Deskriptor 'Recherche' und schaut gegen eine Lichtquelle. Alle Löcher, die hell erscheinen, weisen auf die relevanten Dokumentennummern hin - die Rechercheaufgabe ist gelöst. (s. dazu [Gü71])

1.3 Logische Grundoperationen in Rechercheanfragen

Mit solchen Sichtlochkarten lassen sich die Grundoperationen der Aussagenlogik auf einfache Weise für Rechercheanfragen (Anfragen) realisieren.

Die **Konjunktion**, das logische UND mehrerer Deskriptoren, ergibt sich durch das Übereinanderlegen der zutreffenden Sichtlochkarten. Das Licht scheint nur für die Nummern der Dokumente hindurch, für die **sowohl** der eine Deskriptor **als auch** der zweite Deskriptor **als auch** der dritte … vergeben wurde.

Die **Disjunktion**, das logische ODER, besteht in der **Vereinigung** der Dokumentennummern, die auf den Sichtlochkarten für den ersten Deskriptor mit denen für den zweiten Deskriptor mit denen für den dritten … abgelocht wurden. (Logisches ODER bedeutet hier: Relevant sind die Dokumente, die mit dem ersten Deskriptor **oder** mit dem zweiten **oder mit beiden** usw. gekennzeichnet worden waren.)

Beide Operationen ergeben die Liste der Nummern der relevanten Dokumente. Die **Negation** sollte in Recherchefragen tunlichst vermieden werden.

2 Nutzung eines Programmsystems für die Speicherung von Sekundärdokumenten (R300-Projekt)

Am Beispiel des R300-Typenprojekts für die Information und Dokumentation soll der Anfang der 1970-er Jahre in der DDR erreichte Stand der Arbeit mit wissenschaftlich-technischen Dokumenten dargestellt werden, die in Form von weitgehend unformatierten Daten programmgerecht aufbereitet und dann in R300-Dateien gespeichert und recherchiert werden konnten. Derartige Programmsysteme können heute noch genutzt werden. Allerdings sind die zu lösenden Aufgaben infolge des Fortschritts sowohl der Speicherungsverfahren als auch der Recherchestrategien auf moderneren effektiveren Wegen zu lösen. Ob dabei immer die gewünschte Treffergenauigkeit erreicht wird, steht allerdings in Frage. Hier muss ein Kompromiss zwischen Wirtschaftlichkeit einschließlich Zeitersparnis und Genauigkeit gefunden werden.

2.1 Unformatierte Daten. R300 als rechentechnische Basis

Die bis Anfang der 1960-er Jahre verarbeiteten Daten waren entweder Zahlen (für reine Berechnungsaufgaben) oder zumindest fest formatierte Datensätze (für ökonomische Aufgaben und Berechnungen). Derartige **formatierte Daten** lassen sich relativ einfach in Dateien schreiben und wieder auslesen. Auch Änderungen innerhalb der Datensätze bzw. das Löschen und Hinzufügen ganzer Sätze war mit den verfügbaren Standardverfahren leicht möglich. Die wichtigsten davon arbeiteten mit den Schlüsseln der Datensätze.

Bei der Abbildung wissenschaftlich-technischer Informationen auf Datenstrukturen entstehen Datenelemente und Datensätze, deren Länge in sehr weiten Grenzen variieren können und die sich deshalb nur mit großer Speicherplatzvergeudung auf Strukturen mit festem Format des bisherigen Verständnisses abbilden lassen oder nur jeweils eine vordefinierte begrenzte Anzahl von Werten in einem Datenelement zulassen. Diese Daten waren mit den bis in die PL/1-Zeit hinein gebräuchlichen Verfahren nicht speicherplatzeffektiv zu verarbeiten. Für diese Daten bürgerte sich die Bezeichnung **'unformatierte Daten'** ein. So musste z. B. auch die bisherige Festlegung aufgegeben werden, dass ein Datenelement nur einen (1) Wert enthalten darf. Als Beispiele seien genannt, dass ein Buch häufig nicht nur einen einzigen Verfasser hat oder dass der Inhalt eines Dokuments i. allg. mit mehr als genau einem Deskriptor gekennzeichnet werden muss.

Am Beispiel des Dokumentennachweises, im R300-Projekt [Ro69] Nachweis genannt, wird gezeigt, wie wissenschaftlich-technische Sekundärdokumente speichergünstig und rechercheangepasst auf Datenstrukturen abgebildet werden können (s. Abschnitt 2.2).

Der ersten Digitalrechenanlage der DDR, dem ZRA1 (Zeiss Rechenautomat 1) folgte aus der Produktion des Kombinats Robotron der R300 (Robotron 300), der mit seinen 8 Magnetbandgeräten zur externen Speicherung und Verarbeitung von Massendaten geeignet war. Damit konnten die bei der Verarbeitung von beliebig vielen Datensätzen entstehenden Datenmengen extern gespeichert werden. Das war für formatierte Datensätze im Datenbanksystem Robotron (DBS/R) realisiert.

Für Datensätze mit unformatierten Daten wurde die R300-gestützte Verarbeitung im Typenfeinprojekt für die Information und Dokumentation [Ro69]) angeboten. Im praktischen Betrieb standen für die Nutzerdateien jeweils maximal 6 Magnetbandeinheiten zur Verfügung. Die Daten wurden über 8-Kanal-Lochband eingegeben, das standardmäßig auf dem Organisationsautomaten Optima 528 kodiert wurde. Die Datenausgabe erfolgte über einen 80spaltigen Paralleldrucker.

2.2 Struktur des Dokumentennachweises im R300-Projekt

An der Gestaltung des Nachweises und seinem Inhalt lässt sich der damalige Wissensstand auf dem Gebiet der rechnerunterstützten Verarbeitung unformatierter Daten charakterisieren.

Die zitierten Quellen außer [Ro69] wurden aus der praktischen Arbeit mit wissenschaftlich-technischen Dokumenten in einer betrieblichen Informationsstelle abgeleitet und sind als Arbeitsberichte anzusehen, z. B. [Gü71], [Gü72], [Lö72] und [Rö74]. An zahlreichen Stellen war die dargestellte theoretische Fundierung erforderlich, um zu einer stabilen und erfolgreichen Arbeit zu gelangen. Der Nachweis ist im R300-Projekt in 10 Eingabesätze, Kategorien genannt, gegliedert (s. Bild1). Intern ist der Gesamtnachweis ein Datensatz. Die zehn Kategorien sind dann Datenelemente mit Werten beliebiger Länge und z. T. auswertbarer Struktur.

---

0 ZAC01407

1 F.3.4.; F.3.5.;

2 Ferrit, Eigenschaft; hoechst, Frequenz; Teil, magnetisieren;

Berechnung; Theorie; Magnet, Eigenschaft; Permeabilitaet; Literatur;

4 Schloemann, Ernst;

5 =Microwave behavior of partially magnetized ferrites.

(=Verhalten teilmagnetisierter Ferrite bei Mikrowellenfrequenzen).

6 J. Appl. Phys., Lancaster 41 (1970) 1, s. 204-214, 10 Abb., 20 Lit.,

Bde 01

8 D221: Reg.-Nr. M 0556/08

9 E

Bild 1 Struktur des Dokumentennachweises im R300-Projekt des Kombinats Schaltelektronik (s. [Gü71])

Jede Kategorie beginnt mit ihrem Namen, dargestellt durch eine der Ziffern von 0 bis 9, gefolgt von einem Leerzeichen. Folgezeilen in einer Kategorie werden an zwei führenden Leerzeichen erkannt. Für jeden Einzelwert, der recherchierbar sein soll, ist ein Trennzeichenformat festgelegt: Das Endekennzeichen für einen solchen Einzelwert (Kategorie1:Notation, Kategorie2: Deskriptor(gruppe) ist das Semikolon. In den anderen Kategorien kann nur rein sequentiell im gesamten Eintrag gesucht werden.

In den Kategorien 1 bis 8 sind Folgezeilen zugelassen. Die maximale externe Zeilenlänge beträgt 60 Zeichen. Damit wird die im Format A4 mögliche Zeilenlänge eingehalten, die für die Kontrollmitschrift beim Ablochen der Eingabedaten und für die rechnerunterstützt erzeugten Nutzer-Informationsmittel von Bedeutung ist.

Kategorie 0 - externe Dokumentennummer = Nachweisnummer. Diese Kategorie ist als einzige auf genau 8 Zeichen formatiert. Die laufende Nummer wird durch Kodezeichen für die Dokumentenart und den Vertraulichkeitsgrad ergänzt.

Kategorie 1 - Notationen. Nach dem gleichen Prinzip wie bei der UDK ist bei der Einsatzvorbereitung für das R300-Projekt eine für den konkreten Anwenderbereich spezifische Klassifikation zu erarbeiten. Sie heißt im R300-Projekt **Systematik der Notationen**. Diese Klassifikation muss vor dem Beginn der Nachweisespeicherung eingegeben werden. Jede Notation beginnt mit einem Buchstaben, gefolgt von Ziffern oder Ziffernpaaren, die jeweils vom Vorgänger durch einen Punkt zu trennen sind. Im Dokumentennachweis können mehrere Notationen vergeben werden, wenn das Dokument mehr als ein Fachgebiet betrifft. Bei der Eingabe des Nachweises werden die Werte in Kategorie 1 mit den Notationen in der Systematik verglichen. Jeder Fehler wird protokolliert, und der Nachweis wird abgewiesen. Die fehlerhaften Nachweise erscheinen mit allen erkannten Fehlern im Eingabeprotokoll. Sie können extern korrigiert und dann erneut eingegeben werden.

Kategorie 2 - Deskriptoren. Der **Thesaurus**, die Liste der zum Indexieren zugelassenen Deskriptoren, ist ebenfalls im Rahmen der Einsatzvorbereitung zu erarbeiten und anhand praktischer Erfahrungen weiter zu entwickeln [Lö72]. Da die Deskriptoren zur Kennzeichnung der wichtigen Inhaltskomponenten des Primärdokuments bestimmt sind, sollten in den Thesaurus keine Bezeichnungen der Fach- oder Wissensgebiete aufgenommen werden. Die Gebiete werden durch die Notationen kodiert! Eine wichtige Funktion eines Thesaurus in diesem Sinne ist die Zusammenführung synonymer bzw. quasisynonymer Bezeichnungen, die als Nichtdeskriptoren im Thesaurus verzeichnet sind, auf eine aus dieser Menge - den vereinbarten Deskriptor. Die Verwendung einer nicht als Deskriptor definierten Bezeichnung führte im R300-Projekt ebenfalls zum Abweisen des Nachweises.

In vielen Fällen ist es sinnvoll, nicht zu viele **Komposita** als Deskriptoren zu definieren, zumal die Komposita in Zusammenschreibung ohnehin eine Eigenheit der deutschen Sprache darstellen. Deshalb bietet fast jeder Thesaurus die Möglichkeit, bis zu drei Wörter zu verbinden, was im R300-Projekt durch ein verbindendes Komma kodiert wird. Die Reihenfolge *Grundwort, Bestimmungswort1, Bestimmungswort2* sollte als allgemeine Regel im Thesaurus verwendet werden.

Am Nachweis Nr. 01407 (Bild 2) erkennt man gut, dass am Anfang der Arbeit mit dem R300-Projekt im Arbeitsthesaurus die eben diskutierte normierte Aufgliederung der Komposita noch nicht berücksichtigt war. Das war Gegenstand einer grundsätzlichen Überarbeitung des Thesaurus, die bald folgte ([Lö72], s. a. [Lö75a]).

Kategorie 4 - Verfasser. Hier werden die Verfasser in der Reihenfolge *Nachname, Vorname;* angegeben. Die im Bibliothekswesen geltenden Richtlinien sind bei der Erfassung der Verfassernamen verbindlich. Unstimmigkeiten bei der Schreibweise von Verfassern treten unabhängig von diesen Richtlinien dann auf, wenn unterschiedliche Schriftsysteme Anwendung finden (s. dazu z. B. [Lö71] für die unterschiedliche resultierende Schreibweise japanischer Autorennamen).

Kategorie 5 - Titel des Primärdokuments. Beim Kodieren der Angaben für diese Kategorie sind ebenfalls die Festlegungen des Bibliothekswesens einzuhalten. Ein aus einer Fremdsprache übersetzter Titel ist in Klammern anzugeben.

Kategorie 6 - Quelle des Primärdokuments. Die Quellenangaben folgen gleichermaßen den im Bibliothekswesen geltenden Richtlinien.

Kategorie 8 - Standort des Primärdokuments. Hier wird der Standort des Dokuments in der für die Nutzer zuständigen wissenschaftlich-technischen Bibliothek, gekennzeichnet durch ihr Sigel, angegeben, bei selbständigen Dokumenten ergänzt durch seine Signatur.

Kategorie 9 - Sprache. Für die Kennzeichnung der Sprache des Primärdokuments sind eigene Kurzzeichen festzulegen. Die Angabe mehrerer Sprachen ist zulässig.

2.3 Aus dem Speicher abgeleitete Informationsmittel: Bibliographie, SIV und Recherche

Die konventionelle Informationsversorgung der Nutzer einer Informationsstelle erfolgt über Breiteninformationsmittel wie Neuerwerbungslisten und die speziellen anforderungsbezogenen Recherchen. Diese beiden Leistungen werden auch vom R300-Projekt erbracht: Bibliographie und Recherche. Zusätzlich besteht die Möglichkeit, im Rahmen der *Selektiven Informationsverbreitung* anhand eines gespeicherten Nutzerprofils periodisch im gleichen Rhythmus wie die Bibliographie, also nach jeder Neueinspeicherung von Nachweisen oder in anderem Abstand, die Nutzer mit genau für sie selektierten Informationen zu versorgen. Ein Nutzerprofil in diesem Sinn ist eine relativ stabile und dauerhaft gespeicherte Recherchefrage zur ständigen Abarbeitung. Tabelle 2 zeigt die Informationsmittel in der Übersicht.

In einer Speicheranfrage (Recherchefrage oder Nutzerprofil) lassen sich die Suchelemente beliebig durch Konjunktion, Disjunktion und Negation (Vorsicht! S. Abschnitt 1.3) kombinieren. Die sprachliche Formulierung einer solchen Aufgabe wird leicht ungenau: das 'Und' kann als *'sowohl A als auch B'* (konjunktiv), aber auch als *'A und auch B'* (disjunktiv) gedeutet werden. Formuliert man das sprachlich genau, dann wird die Anfrage sprachlich kompliziert - und außerdem: Wie spricht man die Klammern? Die Darstellungsmöglichkeit als Schaltnetzwerk bietet den Ausweg: Die Anfrage wird exakt und übersichtlich (s. dazu Bild 3 und Bild 4).

Tabelle 2 Informationsmittel im R300-Projekt

Art / Häufigkeit	ohne Anfrage: Breiteninformationsmittel	auf Anfrage: nutzerbezogene Informationsmittel	Anfrage formuliert als
periodisch	Bibliographie	selektive Informationsverbreitung	Nutzerprofil (gespeichert und leicht zu editieren)
einmalig	nicht relevant	Recherche	Recherchefrage

Der fachliche Teil einer einfachen Anfrage in einem Nutzerprofil oder in einer Recherchefrage hat im R300-Projekt z. B. folgende Gestalt:

f (1:(F.4.3. ;* H.2.1.;)! + 2:(DDR + Informatik) * (Rechentechnik + Information/Dokumentation)!?

Durch die Nutzung einer Deskriptorgruppe *Informatik, DDR* wird die Anfrage schärfer, weil die Nachweise, in denen die Deskriptoren *Informatik* und *DDR* zwar beide aber einzeln auftreten, nicht angezeigt werden. Da die Konjunktion kommutativ ist, spielt im Falle der Einzeldeskriptoren, der offeneren Anfrage, ihre Reihenfolge oder ob sie in der Kategorie 2 überhaupt benachbart sind, keine Rolle für die Relevanz des Nachweises.

Mit diesem Beispiel wird gezeigt, auf welcher guten theoretischen Grundlage bereits Anfang der 1970-er Jahre im Gebiet Information und Dokumentation, also bei der Arbeit mit unformatierten Daten gearbeitet wurde. Die rechnergestützte Arbeit im Fachgebiet zwang uns dazu, sie theoretisch zu durchdringen, um alle Möglichkeiten, die die neue Technik eröffnete, auch nutzen zu können.

2.4 Zusammenarbeit mit den Nutzern

Das Ziel der gesamten Arbeit mit der Speicherung von Sekundärinformationen besteht in der Befriedigung des Nutzerbedarfs an Informationen für die Forschungsarbeit bzw. die Informationstätigkeit der Nutzerbereiche im weitesten Sinn. Bis Ende der 1960-er Jahre bestand die Recherche in den vorhandenen Speichern im Suchen relevanter Nachweise in gedruckten Bibliographien oder Referatezeitschriften wie z. B. Technisches Zentralblatt oder Реферативный Журнал sowie in der Suche relevanter Karteikarten in diesem Referatespeicher. Das Problem komplizierter Anfragen stellte sich nicht ausdrücklich. Einen Fortschritt brachte die Anwendung des Sichtlochkartenverfahrens mit der Möglichkeit, nach mehreren (und auch verknüpften) Deskriptoren als Merkmal für die Relevanz eines Dokuments zu suchen. Damit war die Frage nach den Kombinationsmöglichkeiten für die Deskriptoren entstanden - und damit nach der Anwendung der Grundoperationen der Aussagenlogik in einer Recherchefrage. Das blieb anfangs noch unterschwellig, doch mit der Nutzung von Rechnerprogrammen für diese Aufgaben wuchs die Forderung nach bestmöglich strukturierten Recherchefragen.

Es gilt noch heute: Der Nutzerbedarf wird erst dann klar formuliert, wenn der Nutzer gezwungen wird, ihn zu formalisieren. Nun ist es wohl kaum nutzerfreundlich, vom Fachmann eines beliebigen Gebiets zu fordern, seine Anfrage in der sehr formalen Schreibweise eines Programmsystems zu formulieren. Eine heute ganz normale Zwischenstufe besteht in der Verwendung einer aufgabengerechten, aber noch nicht programmbezogenen Sprache. Für die Formulierung einer Recherchefrage bietet sich dafür bis heute ein Schaltnetzwerk an - wie oben gezeigt (Bild 3 und Bild 4). Die klar erkennbare logische Struktur der Anfrage hat auch den großen Vorteil, dass sich Fachleute jedweden Gebiets sehr schnell in diese formale Beschreibungsart hineindenken können. Damit stehen sie als gleichberechtigte aktive Partner bereit, um ihren Informationsbedarf möglichst genau zu formulieren, denn eine verbale Anfrage muss normalerweise mehrfach durchdacht werden, ehe man dafür die richtige Recherchefrage herausgearbeitet hat. Das ist bei komplexen Anfragen ein Prozess mit mehreren Zyklen. In der Praxis haben wir bis zu vier Varianten erarbeitet, bis die Anfrage dem Nutzerbedarf wirklich genau entsprach. Am Beispiel des Schaltnetzwerks konnten dem Nutzer auch die logischen Grundoperationen anschaulich erläutert werden - bis hin zur Gefährlichkeit der Negation eines Suchmerkmals in einer Anfrage.

Die genannten Probleme waren neben der theoretischen Bedeutung auch praktisch relevant, wenn man bedenkt, dass Rechercheläufe in einem R300-Nachweisespeicher mehrere Stunden dauern konnten - und die dabei entstehenden Kosten, die der Nutzerbereich tragen musste, sollten sich in einem bestmöglichen Ergebnis niederschlagen!

3 Unformatierte Daten in der Ausbildung

An den Sektionen Informationsverarbeitung der TU Dresden und der IH Dresden wurde ab 1974 in der Spezialisierungsphase je ein Praktikum unter Nutzung des R300-Projekts [Ro69] durchgeführt ([Lö75b]).

In diesen Praktika wurden die Studenten mit den Besonderheiten der Verarbeitung wissenschaftlich-technischer Informationen bekannt gemacht. Dazu gehörte die aktive Arbeit mit einem Arbeitsthesaurus einschließlich der Festlegung neuer Deskriptoren und die Anwendung von Grundsätzen einer systemspezifischen Klassifikation. Außerdem waren aktuelle Primärdokumente auszuwerten und die dafür geeigneten Sekundärdokumente (Nachweise) zu erarbeiten und einzuspeichern. Schließlich mussten aus dem Gesamtspeicher aufgabengerechte Informationsmittel abgeleitet werden.

Die Studenten lernten auch die mit diesen Verfahren verbundenen technologischen Fragen der Verarbeitung dieser unformatierten Daten kennen, so z. B. die Notwendigkeit, die ständig anwachsende Zahl der Nutzerdatensätze extern zu speichern und sicher zu verwalten. Zu diesem Zeitpunkt wurde bereits erkannt, dass die bisher als qualitativ unterschiedlich behandelten Verfahren zur Speicherung und Weiterverarbeitung von Datensätzen mit formatierten und unformatierten Informationen in absehbarer Zeit zusammengeführt werden müssen. Die sich entwickelnde Datenbanktechnik sollte das in Zukunft ermöglichen. Das Praktikum wurde nach der Praxiseinführung des AIDOS (s. a. [Nau08]) auch in der Ausbildung auf dieses System umgestellt.

Literaturverzeichnis

[De05] Dewey-Dezimalklassifikation DDC22 Band 3 Haupttafeln 600-999. KG Saur München (2005), 1076 S.

[Gü71] Güldner, R.; Löhr, H.: Aufgaben und Arbeitsweise der Informationsstelle eines Kombinats. Gleitender Übergang von der Steilkartei über die Sichtlochkartei zur Datenverarbeitung. Informatik, Berlin 18(1971) 6, S. 46-49, 4 Abb., 1 Tab.

[Gü72] Güldner, R.; Löhr, H.: Stand der EDV-Anwendung in der Zentralen Informationsstelle des VEB Kombinat Schaltelektronik. Informatik, Berlin 19(1972) 2, S. 35-37, 2 Abb., 1 Tab., 3 Lit.

[Lö71] Löhr, H.: Transliteration japanischer Wörter über die kyrillische in die lateinische Schrift. Informatik, Berlin 18(1971) 5, S. 48-52, 1 Abb., 3 Tab., 5 Lit.

[Lö72] Löhr, H.: Anpassung eines Arbeitsthesaurus an übergeordnete Thesauren. Informatik, Berlin 19(1972) 6, S. 30-34, 1 Abb., 1 Tab., 4 Lit.

[Lö75a] Löhr, H.: Analyse der Struktur eines Begriffs - ein Beitrag zur Festlegung von Deskriptoren. Berlin 22(1975) 5, S. 28-33, 3 Abb., 4 Tab., 8 Lit.

[Lö75b] Löhr, H.: Erste Erfahrungen bei der Ausbildung von Hochschulingenieuren im Fachgebiet wtI. Informatik, Berlin 22(1975) 4, S. 36-40, 1 Abb., 2 Tab., 5 Lit.

[Nau08] Naumann, P.: Das Robotron Informationsrecherchesystem (IRS) AIDOS: Software-Technologie von gestern? In diesem Band S.

[Ro69] Typenfeinprojekt "Einsatz des Robotron 300 für die Information und Dokumentation", Dresden: VEB Kombinat Robotron, Dresden (1969)

[Rö74] Löhr, H.; Römer, R.: Überführung des R300-Typenprojekts in den Routinebetrieb. Informatik, Berlin 21(1974) 3, S. 30-34, 5 Abb., 4 Tab., 5 Lit.

Entwicklung eines verteilten bildschirmorientierten Buchungs- und Auskunftssystems für die Materialwirtschaft

Rüdiger Liskowsky
Kaitzer Str. 131
01187 Dresden

Klaus Panceram
Tauernstr. 34
01279 Dresden

1 Ausgangsituation

Das Projekt „Datenerfassungs-, Buchungs- und Auskunftssystem für die Materialwirtschaft im online Betrieb (DEBAMO)“ entstand in den Jahren 1974-77 im Stammbetrieb des Kombinats Robotron, dem VEB Robotron-Elektronik Radeberg. Aufgabe des Projektierungskollektivs in der Struktureinheit D (Datenverarbeitung) war es, Software zur Anwendung innerhalb des Kombinats zu entwickeln und zu einem stabilen Dauerbetrieb zu führen. Die Projektierungskollektive von D hatten nichts mit dem Zentrum für Forschung und Technik (ZFT) zu tun, dessen Zielstellung darin bestand, Systemunterlagen für die allgemeine Anwendung in der DDR zur Verfügung zu stellen. Lediglich im Rahmen der Stufe E1 (Pflichtenheft) wurde geprüft, ob domänenorientierte „Sachgebietsorientierte Programmiersysteme“, hier speziell der SOPS MAWI eine vorgefertigte Säule für die EDV-Anwendung in der Materialwirtschaft sein kann [GR06].

Die Situation in den Jahren 1973/74 war auf dem Gebiet der Informationsverarbeitung innerhalb des Kombinates durch folgende Faktoren gekennzeichnet. In der Softwareentwicklung wurden die ersten Methoden zur Modellbildung bekannt. Die IBM kreierte Anfang der 70-er Jahre ihre Methode „Hierarchy plus Input Process Output (HIPO)“, in der die Ideen der strukturierten Programmierung und des strukturierten Entwurfs umgesetzt wurden [IBM71]. Als wichtiges Darstellungsmittel für den funktionsorientierten Entwurf entstanden der Funktionsbaum sowie die HIPO-Diagramme für den Grob- und Feinentwurf. Ehe IBM die HIPO-Methode außer Haus frei gab, folgte als wertvolle Ergänzung die Struktogrammtechnik von Nassi-Schneiderman [NS73]. Außerdem kam als Entwurfsmittel die Entscheidungstabellentechnik zum Einsatz [STA74].

Vorrangige Programmiersprache war Assembler, teilweise wurden Programmabschnitte in PL1 programmiert. Die objektorientierten Entwurfsmethoden entstanden erst nach der Entwicklung der objektorientierten Programmiersprachen, wie Eiffel, C++ und später Java. Sie wurden aus dem Erfordernis heraus geschaffen, um analog den anzuwendenden Programmierparadigmen die entsprechenden Entwurfselemente bereitzustellen. Die damaligen Beschreibungsmittel unterstützten vorerst das strukturierte und funktionsorientierte Vorgehen bei der Problemanalyse. für die Datenmodellierung waren es die Datenstrukturierung nach Chen [CHE77] und das Codd’sche Relationenmodell [COD71], die beide für den Entwurf von Datenbanksysteme noch heute von großer Bedeutung sind.

Auf dem Gebiet der Gerätetechnik herrschte eine Übergangsituation vor zwischen dem für nationale Belange entwickelten Robotron 300 und dem erwarteten ersten ESER-Rechner EC 1040. Dazwischen lag der R21, der dem Projektierungskollektiv größtenteils als Testrechner diente. Die Ein-/Ausgabe-Peripherie bestand im Wesentlichen aus Lochkartenleser und Paralleldrucker. In diesem Umfeld wurde in der DDR das Bildschirmsystem BSS/R300 entwickelt, um erste Einsatzfälle schon mit dem Rechner R300 realisieren zu können [GAR72].

Das erste Projekt, das mit der Bildschirmtechnik BSS/R300 entwickelt wurde, war das „System automatisierter Leitungshilfen (SAL)“, welches für die Nutzung durch die Stabsorgane der Kombinatsleitung vorgesehen war. Leider war ihm aus folgenden Gründen kein stabiler Dauerbetrieb beschieden:

- keine kontinuierliche sondern überwiegend stochastische Nutzung durch die Mitglieder der Kombinatsleitung
- sporadische Datennutzung oftmals nicht vorhersehbar
- zu geringe im Direktzugriff liegende Speichertechnik (ein polnischer Magnettrommelspeicher mit 400000 Zeichen, ansonsten nur Magnetbandtechnik).

Die Wechselplattengeräte standen erst später mit der Einführung der ESER-Technik beginnend beim R21 zur Verfügung. Der Stamm des SAL-Projektierungskollektivs wurde beauftragt, ein weiterentwickeltes Bildschirmprojekt für die Nutzung im eigenen Kombinat zu schaffen, welches die vorgenannten Mängel von vornherein ausschließt.

2 Einsatz von Bildschirmtechnik im Arbeitsprozess eines Betriebes

Im betrachteten Zeitraum 1974-77 wurde in der DDR versucht, das **H**alb**a**utomatische **D**aten**e**rfassung**s**ystem **HADES daro 1600** einzusetzen. Es beinhaltete eine Abkehr von den herkömmlichen datenträgererzeugenden Sologeräten hin zu alphanumerischen Datenerfassungsplätzen verbunden mit dezentralen Abfrageeinheiten. Damit sollten nahe am Ursprungsort, z. B. in Fertigungsprozessen, Daten dezentral erfasst und für die Verarbeitung zentral an einem Steuerrechner ausgewertet werden [KRJ06].

Vorzugsweise um den Absatz der HADES-Technik nicht zu gefährden, wurde der Einsatz von Bildschirmgeräten, auch wenn sie im direkten Betrieb mit dem Rechner **online** zu betreiben waren, von staatlichen Stellen nicht angepriesen. Ein Grund mag gewesen sein, dass sie zunächst innerhalb des Hauses Robotron erprobt werden sollten, bevor sie nach außen vertrieben wurden. Auch bei den Sachgebietsorientierten Programmiersystemen, die für die allgemeine Anwendung im ZFT entwickelt wurden, spielte der mögliche Einsatz von Bildschirmtechnik keine Rolle [GR06]. Eine ausführliche Studie zur Anwendung von online-Technik bei der Produktionslenkung aus dem Jahre 1976 zeigt [GLS76], dass die HADES-Technik bei vorwiegender Datenerfassung für die Fortschrittskontrolle recht gut geeignet ist. Dagegen sollte die Bildschirmtechnik bei vorwiegender Dialogarbeit (Auskunftserteilung in Verbindung mit Verfügbarkeitskontrollen) eingesetzt werden. Dies ist eine Erkenntnis, die sich heute bis zu Handys (mobile und stationäre Endgeräte) durchgesetzt hat. Auch die ergonomische Gestaltung von Bildschirmoberflächen war damals noch ein Fremdwort.

Deshalb verließ sich das DEBAMO-Projektierungskollektiv vordergründig auf die Nutzung pragmatischer Grundsätze, die aus der Organisationsgestaltung bekannt waren. Es zeigte sich, dass das geschaffene Materialwirtschaftsprojekt mittels Einsatz der Bildschirmtechnik die ergonomischen Prinzipien nach [ISO06] bereits recht gut unterstützte. Es wurden Vorwissenstrukturen der Nutzer beim Aufbau der Bildschirmbilder genutzt, indem bekannte Strukturen bisher verwandter Belege nur wieder erkannt zu werden brauchten. Damit verbessern sich automatisch Erwartungskonformität und Selbstbeschreibungsfähigkeit des Dialogsystems. Generell wurde von den Vorzügen profitiert, die ein Einsatz der Bildschirmtechnik als Ein-/ Ausgabemedium mit sich bringt.

3 Entwurfsmethoden der Software

Das Projekt wurde nach den in der DDR üblichen Entwicklungsstufen E1 bis E6 (stabiler Dauerbetrieb) durchgeführt. In Auswertung des SAL-Projekts sollte die Bildschirmtechnik in einem Leistungsprozess innerhalb des Kombinates Robotron zum Einsatz kommen, also direkt vor Ort in einem betriebswirtschaftlichen Teilgebiet. Für die Werktätigen sollten ein „Zwang“ zur Nutzung als Arbeitsmittel bestehen und damit die sporadische Nutzung in einer Betriebsleitung ausgeschlossen werden.

Im Zuge der Anforderungsanalyse (Stufe E1) wurde als Teilgebiet die Materialwirtschaft ausgewählt, in der die Kommunikation mit dem System aufgrund der Warenbewegungen und der Disposition relativ hoch geschätzt wurde. Als Einsatzbetrieb wurde der damalige Stammbetrieb des Kombinats, der VEB Robotron-Elektronik Radeberg, ausgewählt. Damit wurde die Zielstellung postuliert, die Bildschirmtechnik BSS/ESER als ergonomisches Ein- und Ausgabemedium direkt in den dezentralen Fachabteilungen einzusetzen, womit in der DDR in den Jahren 1975-77 Neuland betreten wurde [LIS77].

Im Rahmen der Stufe E1 (Pflichtenheft) fand eine Verträglichkeitsprüfung statt, ob das domänenorientierte „Sachgebietsorientierte Programmiersystem“ SOPS MAWI innerhalb des online-Materialwirtschaftssystems verwendet werden kann [GR06]. Nach dem heutigen Sprachgebrauch waren lediglich andere Ein-/Ausgabegeräte zu verwenden und damit eine andere Benutzungsschnittstelle zu schaffen. Die Verträglichkeitsuntersuchungen verliefen vor allem auf Grund der starren Anpassungsmechanismen des SOPS negativ. Die enthaltenen Buchungsalgorithmen waren nicht variabel genug und es waren keine Schnittstellen für den Anschluss von Bildschirmgeräten vorgesehen. Damit musste die Aufgabe vom ORZ des Kombinats auf Basis einer Eigenentwicklung selbst übernommen werden.

Die Voruntersuchungen führten zu einer Aufgabenstellung für das Projekt DEBAMO unter Einsatz der Bildschirmtechnik BSS/ESER. Es wurde in der Stufe E1 im ORZ des Kombinates Robotron im Beisein des Direktors Materialwirtschaft und maßgebender Vertreter des Stammbetriebs VEB Robotron-Elektronik Radeberg verteidigt. Zu Beginn der Projektierung galt der Grundsatz „Erst Organisieren dann Automatisieren". D. h. die zugehörige Problemlösung wurde erst nach einer gründlichen Analyse des Altsystems vor Ort in den Fachabteilungen des VEB Robotron-Elektronik Radeberg entwickelt. Um eine möglichst geringe Gedächtnisbelastung der Nutzer zu realisieren, wurde Wert darauf gelegt, die verwendeten materialwirtschaftlichen Dispositionsformulare nur wenig geändert auf dem Bildschirm zu präsentieren. In den Dialogmasken wurde versucht, nicht mit Abkürzungen zu arbeiten, womit ebenfalls eine hohe Kompatibilität zu den Vorwissenstrukturen der Nutzer erreicht wurde. Die Beispiele zur Ergonomie des Systems ließen sich fortsetzen. Generell wurde versucht, möglichst Bekanntes auf dem Bildschirm wieder anzubieten. Darüber hinaus war eine sichere Nutzerrechtsprüfung zu realisieren, die gewährleistet, dass nur berechtigte Bearbeiter z.B. Materialbuchungen vornehmen konnten.

Für den Grobentwurf wurde zunächst eine Funktionsanalyse nach Teilgebieten, vergleichbar mit einem Funktionsbaum analog der HIPO-Methode, durchgeführt. Für die materialwirtschaftliche online Buchung im Stammbetrieb erwies es sich als zweckmäßig, die Teilgebiete Bestellwesen, Wareneingang, Wareneingangskontrolle, Materialdisposition (Lagerübernahme, Materialentnahme) sowie Dateirecherchen als sachgebietsorientierte Funktionskomplexe zu bearbeiten. Innerhalb jedes Teilgebiets wurden als Bestandteile der Stufe E3 (Problembeschreibung) mittels der damals modernen softwaretechnologischen Methoden folgende Diagramme erarbeitet:

- HIPO-Diagramme (EVA-Tabellen) für den Grobentwurf
- Entscheidungstabellen und
- Struktogramme für den Feinentwurf.

Sie dienten der funktionsorientierten Zerlegung und zum Auffinden der zu bearbeitenden Datenstrukturen. Sie bildeten die Vorlage für die anschließende Programmierung. Die Entscheidungstabellen wurden dazu benutzt, den Dialogablauf aus dem Sachproblem heraus zu spezifizieren, um damit die Programmvorlage weiter zu verfeinern [LIS83]. Abbildung 1 zeigt beispielhaft eine Entscheidungstabelle für den Dialogablauf im Teilgebiet Bestellwesen.

MBB	Eingangsgrößen	Ausgangsgrößen	Folgebild
	▷ (MBB, geheimname	BESTNR=00...0	MBB001/1
MBB001/1	BESTNR=*		ST1001
	BESTNR=99...9	MATNR, LNR, ME bestelldaten	 MBB001/2
	BESTNR=00...0	Fehlernachricht	MBB001/1
MBB001/2	bestelldaten(neu): - Bestellmenge/Monat - Ausweichmaterial - Bestelldatum - Bestellstatus	bestelldaten (neu) (Quittungs- ausgabe)	 MBB001/1

Abbildung 1: Entscheidungstabelle für den Dialogablauf im Teilgebiet Bestellwesen

Das Problemkennzeichen MBB oben links bezeichnet den Tabellen(=Programm)namen. Als Eingangs- bzw. Ausgangsgrößen gelten die im Dialogschritt veränderten BS-Ein- bzw. Ausgabewerte. Für das Teilgebiet Bestellwesen existiert nur ein Dialogprogramm MBB. Es führt die Änderungen von bestelldaten(neu) aufgrund von Bestellbestätigungen, Nachbestellungen (Mehrbedarf) oder Stornierungen durch. Das einzige Bild MBB001 wird sowohl für die Auswahl der Bestellungen(Dialogschritt MBB001/1) als auch zur eigentlichen Bestelländerung (Dialogschritt MBB001/2) benutzt. Die auf dieser Basis gewonnenen Entwurfsdokumente bildeten die Grundlage für eine Abstimmung aller Teilgebietsmodelle untereinander und für die anschließende Programmierung. Die Problemlösung der Software wurde in der Stufe E3 vor dem Anwender verteidigt.

Mit dem Entwurfsmittel Entscheidungstabelle (Kombinatorischer Automat) lassen sich also durch logische Kombination aller Bedingungen (hier Eingangsgrößen in den Dialogbildern) alle notwendigen Aktionen (hier zu verarbeitende Ausgangsgrößen) auffinden. Aus heutiger Sicht kann man die Entscheidungstabelle als eine erste Methode zur Dialogmodellierung ansehen [GOE94].

Entwurfsergebnisse waren HIPO-Diagramme, Entscheidungstabellen und Struktogramme, die als Programmiervorlage genutzt wurden. Diese (agile) Vorgehensweise ist auch heute noch vielfältig anzutreffen. Der Einsatz von CASE zur automatisierten Generierung von Code-Gerüsten aus Diagrammen ist leider noch kein Allgemeingut.

4 Schaffung eines Vielfachzugriffssystem zur Echtzeitverarbeitung

Das Projekt DEBAMO wurde damals Mitte der 70-er Jahre im Betriebssystem DOS/ES entwickelt. Dies ist ein reines Stapelbetriebssystem. Ein System, das den Teilhaber- oder Teilnehmerbetrieb für den Mehrfachzugriff in Echtzeitbetrieb unterstützt, stand nicht zur Verfügung.

Um die zeitgleiche Arbeit mehrerer online angeschlossener Bildschirmgeräte und Nutzer zu ermöglichen, musste in einem DOS-Programmbereich ein *Vielfachzugriffssystem (VZS)* geschaffen werden, das als problemunabhängiges Echtzeitsteuersystem den Teilhaberbetrieb einer endlichen Zahl von Nutzern a_i mit der Menge der anwendungsorientierten Problemprogamme (PP) c_k realisiert [GUH77]. Das Grobschema des VZS zeigt Abbildung 2.

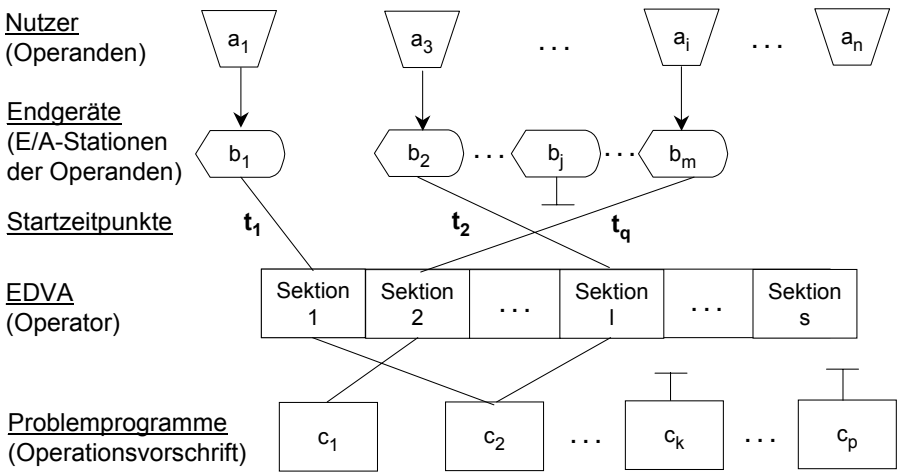


Abbildung 2: Schematische Darstellung des Vielfachzugriffssystems VZS

Die virtuelle Verwaltungseinheit, die in einer bestimmten Zeiteinheit einem Teilhaber zur BS-Arbeit mit einem Problemprogramm zur Verfügung steht, wird Sektion l genannt. Das VZS löst generell die folgenden Verwaltungsaufgaben

- die Speicherzuteilung für die unterschiedlichen gleichzeitig ablaufenden Problemprogramme c_k, indem die jeweils aktive Sektion einen zur Verarbeitung feststehenden HS-Bereich zugewiesen bekommt und alle anderen für das „core swapping“ ausgelagert und bei Bedarf wieder aktiviert werden
- die Zeitteilung zwischen den an den Stationen l arbeitenden Teilhabern und Problemen, indem die Problemprogramme unterbrechbar programmiert werden.

Das VZS wurde speziell für das Betriebssystem DOS/ES und die eingesetzte Bildschirmtechnik BSS/ESER und die betrieblichen Echtzeitdateien in der Materialwirtschaft entwickelt. Die Hauptbestandteile des realisierten VZS zeigt Abbildung 3.

Sämtliche Ein-/Ausgaben der Teilhaber werden vom BS-E/A-System gesteuert. Im Testbetrieb oder bei einer Havarie kann anstatt des E/A-Systems mit der BS-Simulation über Konsole und Paralleldrucker gearbeitet werden. Die Druckunterstützung steuert normalerweise mittels der Fernzugriffsmethode ROTAM die Druckausgabe auf den Seriendrucker SD 1156. Der Dialogmonitor organisiert den Verwaltungsdialog mit dem Teilhaber. Er arbeitet eng zusammen mit der Sektionenverwaltung, die die Zustände unterbrochener Sektionen sichert und deren Auslagerung in das VZS-Reservat veranlasst. Weitere Dienstleistungen der Sektionenverwaltung sind die Aufrechterhaltung der Kommunika-

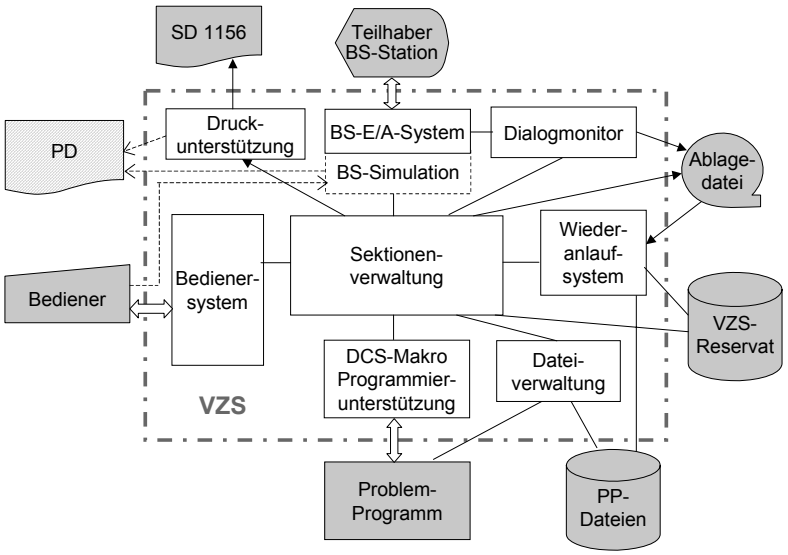


Abbildung 3: Hauptbestandteile des Vielfachzugriffssystem VZS

tion zwischen den Sektionen und das Realisieren der Ausgaben auf die Ablagedatei. Der DCS-Makro dient zusammen mit weiteren Makroanweisungen der Programmierunterstützung für die Problemprogramme. Der Dateiverwalter koordiniert die Arbeit mit den Problemprogrammdateien und sichert insbesondere, dass Daten durch den möglichen Vielfachzugriff nicht gleichzeitig in unkontrollierter Art verändert werden.

Die aufgeführten Komponenten führten im Laufe der weiteren Entwicklung zur Schaffung von Echtzeitbetriebssystemen und verteilten Systemen [REP99]. Einige der geschilderten Funktionen werden problemneutral vom Betriebssystem übernommen. Trotz allem müssen auch neuere Systeme dafür sorgen, dass Daten für den Vielfachzugriff koordiniert werden. Für den Fall, dass die zentrale Datenbank oder der gesamte Server ausfallen, muss über Replikate für den Datenabgleich und ebenfalls für den Wiederanlauf gesorgt werden.

Ein sehr naheliegender Weg zur Vereinfachung von DEBAMO wurde in Radeberg mit der Verfügbarkeit leistungsfähiger PC-Technik gegangen. Anstatt sämtliche Verarbeitungskapazität in einem zentralen Rechner zu konzentrieren, übernahmen die dezentralen E/A-Stationen nun selbst die Problembearbeitung. Dateiverwaltung und Wiederanlauf verblieben zunächst weiter auf dem Zentralrechner. Allgemein gesehen hat der Einsatz dezentraler Rechentechnik mit der Aufgabenverteilung sowie der Schichtenstruktur aus dem OSI-Referenzmodell viel zum Entwurf *Verteilter Systeme* und damit zu Schaffung verteilter Anwendungen beigetragen [REP99].

5 Konzeption der Datenbasis

Die Problemprogrammdateien wurden konsequent nach dem *Relationalem Konzept* aufgebaut [LIS83]. Ausgangspunkt bildete die bestehende betriebliche Organisation mit den vorhandenen Verknüpfungen zwischen den Informationsoperanden (Abb. 2) und den Abhängigkeiten zwischen den Attributmengen der materialwirtschaftlichen Daten.

Auf einem der Relationensynthese stark angenäherten Weg [WED74] wurde entschieden, welche Attributmengen zweckmäßig in einer normalisierten relationalen Datei zusammenzufassen sind. Ein Einstieg ins Dateisystem im Teilgebiet Wareneingang ist die Wareneingangsscheinnummer WESNR gekoppelt mit der Positionsnummer POSNR. Über die funktionale Abhängigkeit (WESNR, POSNR) gelangt man zur Lieferantennummer LNR, zur Bestellnummer BESTNR und zur Materialnummer MATNR, womit der Zugriff zu den wichtigsten Dateien der Materialwirtschaft hergestellt ist. Zwischen den Attributmengen MATNR und DIGRNR (Disponentengruppennummer) besteht keine funktionale Abhängigkeit. Erst durch die Verkettung beider Schlüssel zu (MATNR, DIGRNR) wird die volle funktionale Bindung zu den Dispositionsdaten hergestellt.

Ohne Verwendung eines relationalen Datenbanksystems wurden im Projekt DEBAMO 13 Wechselplatten- und 2 Magnetband-Dateien geschaffen, auf die 24 online Problemprogramme zugreifen. Mit der heutigen Verfügbarkeit leistungsfähiger Anwendungs-Server-Systeme würde man verteilte Technologien einsetzen, wobei die Nutzung des relationalen Ansatzes zur Gewinnung normalisierter Dateien auch weiterhin sehr nützlich sein sollte.

6 Nutzung des Makrokonzepts für Wiederverwendung

Zur Unterstützung der Problemprogrammierer wurde ein Makrosystem geschaffen, mit dem im materialwirtschaftlichem Problemprogramm Dienstleistungen sowohl des VZS (siehe DCS-Makro Abb. 5) als auch einheitlich Programmierhilfen bereitgestellt werden. Im einfachsten Fall wurden dadurch kleinere Programmbausteine geschaffen, die bestimmte Befehlsfolgen und –abläufe vereinheitlichten. Das Ziel des Makrokonzepts besteht darin, ständig wiederkehrende Sequenzen von Einzelanweisungen mit möglichen Modifizierungen in Form einer neuen (selbstdefinierten) Makro-Anweisung für die ganze Sequenz zur Verfügung zu stellen. Die Modifizierung der Makro-Anweisung wird in der Regel durch Makro-Parameter gesteuert.

Über die Parameteraufrufe des DCS-Makros werden dem VZS die Wünsche des jeweiligen Programms zur Durchführung der erforderlichen Funktionen im Steuersystem mitgeteilt. Die Übergabe der Parameter erfolgte sowohl in Form von Stellungsparametern als auch in Form variabler Namensparameter im Makrokommandoaufruf. Beispielsweise bedeutet der DCS-Makroaufruf

DCS ART=OPEN, INF=JA; Problem=MBB

Programmeröffnung im VZS für das Problem MBB (Abb. 2) mit Informationen=ja über den Sektionenzustand.

Es gab eine Vielzahl von Makros zur Erledigung der unterschiedlichsten Aufgaben. Beispiele sind Makros für Programmanfang und -ende, Makros für Formatumwandlungen der Datenbestände, Makros zur Übermittlung von „Wünschen“ an das VZS zur Bereitstellung und Rückspeicherung benötigter Datensätze sowie zum Sperren von „Fremd“-Zugriffen. Zusätzlich wurden die Anweisungen zur Ein- und Ausgabe von Bildschirmbildern, d.h. also der eigentlichen Dialogdurchführung in einem Makro mit Anweisungen zum Speichern des jeweiligen Bildinhaltes auf einem Kontrollausgabe-Magnetband festgehalten, Vor allem beim Start des eigentlichen Nutzbetriebes haben diese Maßnahmen wertvolle Dienste geleistet, konnten doch die Abläufe von einem bestimmten Bildschirmgerät aus sehr genau verfolgt und mögliche Fehlabläufe schnell erkannt und beseitigt werden.

Die entwickelten Makro waren zwar „Einzelanfertigungen“, sie waren zur effektiven Unterstützung der Programmierer jedoch sehr nützlich. Aus heutiger Sicht war es ein Vorgriff auf Komponententechnologien, die an den unterschiedlichsten Fronten (CORBA, JavaBeans, EJB, COM+, .NET) geschaffen wurde, um den Grad der Wiederverwendung von Bausteinen nach dem Vorbild des Maschinenbaus erheblich zu steigern. Mit dem durchgängig organisierten Makrokonzept innerhalb von DEBAMO wurde in diese Richtung bereits vorgedacht, allerdings mehr auf eine effektive Programmierunterstützung gerichtet als in Hinblick auf die Bereitstellung und Wiederverwendung des Bausteinsystems für andere Projekte.

7 Echtzeit-Wiederanlaufsystem

Eine unerlässliche Voraussetzung für ein im online Betrieb arbeitendes Buchungssystem ist die Gewährleistung der Datensicherheit. Zu diesem Zweck wurde als Bestandteil des VZS ein Wiederanlaufsystem geschaffen (Abb. 3).

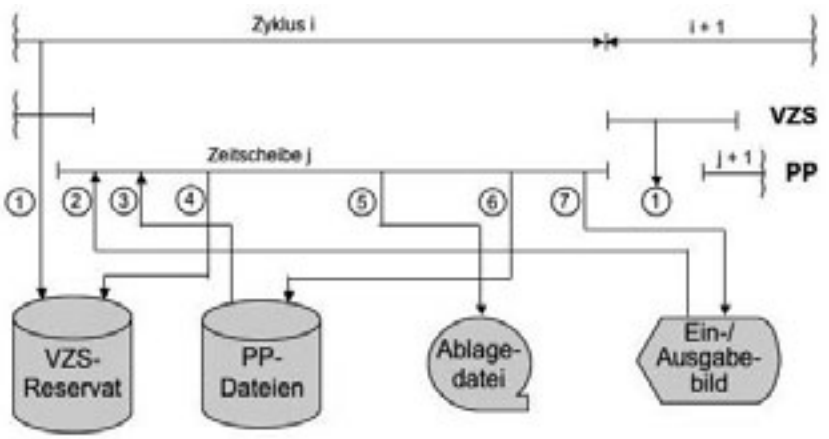


Abbildung 4: Zeitgetaktete Darstellung des Wiederanlaufsystems

Das Wiederanlaufsystem soll die Weiterführung der online Arbeit nach Programm- bzw. Gerätefehlerunterbrechungen garantieren. Die grobe Funktion des realisierten Wiederanlaufsystems mit den Takten eines PP in der Zeitscheibe j zeigt Abbildung 4 [LIN83].

Vor der Arbeit mit einem neuen PP ③, angefordert vom Teilhaber ②, rettet das VZS den Hauptspeicherzustand des zu unterbrechenden PP in das VZS-Reservat ①. Das PP beginnt seine Arbeit in der Zeitscheibe j mit der Auswertung des Eingabebildes ②. Danach werden normalerweise die zugehörigen PP-Dateien gelesen ③. Jede Änderung an den PP-Dateien ist zur Gewährleistung der Dialogschritt-Sicherheit in das VZS-Reservat zu hinterlegen ④. Nach der internen Verarbeitung durch das PP werden sämtliche geänderten PP-Sätze neben der Ablage in die normale PP-Datei ⑥ zusätzlich auf eine Ablagedatei ⑤ ausgegeben. Erst danach ist der Dialogschritt mit dem Teilhaber beendet ⑦.

Mit dieser Organisation werden die folgenden Aufgaben zur Datensicherheit erledigt:

- Problemreorganisation zwangsweise unterbrochener Sektionen durch Rückspeichern aus dem VZS-Reservat,
- Dateiwiederanlauf durch Aktualisierung der Dateiverzeichnisse nach dem Abbruch,
- Dateireorganisation durch Wiederherstellung der PP-Dateien mit duplizierten Dateiausgaben von der Ablagedatei.

In der Praxis haben sich diese Maßnahmen während des Nutzbetriebs sehr positiv ausgewirkt. Bei einem Datenanfall von ca. 1000 bis 1500 Buchungen pro Tag lief das Projekt über fast 5 Jahre stabil, bis es Mitte der 80-er Jahre durch den Einsatz dezentraler PC-Arbeitsplätze abgelöst und dezentralisiert wurde.

In heutigen verteilten Systemen wird ein Wiederanlaufsystem unter dem Begriff Verfügbarkeit von Kommunikation eingeordnet. Mit der Offenheit von WAN haben sich die Schutzbedürfnisse der Menschen natürlich wesentlich erweitert. Die Datensicherheit wird in der Regel durch Replikationsalgorithmen zwischen den teilnehmenden Stationen erreicht. Der Wunsch nach einer sicheren Echtzeitverarbeitung ist grundlegend und nach wie vor zu gewährleisten.

8 Schlussbetrachtungen

Das bildschirmorientierte Echtzeitsystem für die Materialwirtschaft im DOS/ES lief für den Anwenderbetrieb Robotron-Elektronik Radeberg sehr erfolgreich. Nach Meinung der Verfasser war es das erste online Bildschirmprojekt, welches in der DDR in einen stabilen Nutzbetrieb überführt wurde. Der Einsatz der Bildschirmtechnik direkt in den Fachabteilungen hatte sich trotz der notwendigen online Kopplung zu einem Zentralrechner bewährt.

Obwohl das Projekt bereits eine zukunftsweisende Entwicklung darstellte, waren schon damals weitere Verbesserungen in Hinblick auf eine verkaufsfähige Lösung erkennbar:

- Weiterentwicklung zu einer allgemeinen anpassungsfähigen Problemlösung für beliebige Einsatzfälle
- Unterstützung weiterer Betriebssysteme (damals zunächst das OS/ES) bis hin zur Plattformunabhängigkeit vorrangig bei Einsatz dezentraler Geräte- bzw. Bildschirmtechnik
- Verallgemeinerung und Weiterentwicklung geschaffener Einzelbestandteile, wie
  - des Dateisystems unter Nutzung vorgefertigter Datenbank-Managementsysteme,
  - des VZS durch Herstellen einer Zusammenarbeit mit neuen Betriebssystem-Teilhaberkomponenten,
  - Erhöhung der Wiederverwendung von Softwarebausteinen durch Verwendung verallgemeinerter Makrokonzepte bzw. von Komponententechnologien.

Diese Vorstellungen wurden dem Leiter der Struktureinheit D im ORZ des Kombinates und dem Betriebsdirektor Robotron-Elektronik Radeberg 1979 mitgeteilt. Nachdrücklich wurde die Notwendigkeit unterstrichen, weiter auf den genannten Gebieten arbeiten zu können, um wertvolles Wissen nicht zu verschenken. Leider wurde anders entschieden, denn das Projektierungskollektiv gehörte nicht zum ZFT, in dem allgemeingültige Systemunterlagen zu schaffen waren. Immerhin gelang es, einen Teil der Gruppe an ein kleinrechnergesteuertes Projekt für die Lagerwirtschaft im Echtzeitbetrieb zu binden. Mitte der 80-er Jahre entwickelte die Gruppe vernetzte PC-Systeme für das Hotelwesen (Hotelcomputersystem AURIS), die beispielsweise im Grand Hotel Berlin und in den damaligen Dresdner Interhotels Bellevue und Dresdener Hof sowie im Motel Dresden zum Einsatz kamen. Damit überbrückte man die Wende, konnte aber die Insolvenz der neuen Anstellungsfirma nicht verhindern.

Nach unserer Meinung deutet das Projekt aus dem Jahre 1977 auf zukunftsweisende Entwicklungen der Informatik hin. Erstmalig wurde in der DDR die Bildschirmtechnik online gekoppelt mit einem Zentralrechner in einem Leistungsprozess vor Ort eingesetzt. Für die Entwicklung wurden mehr oder weniger unbewusst ergonomische Prinzipien der Dialogführung genutzt, die damals noch weitgehend unbekannt waren. Den Nutzern wurde die Arbeit erheblich erleichtert, was der Grundpfeiler für die Akzeptanz der BSS-Technik war. Das Echtzeitsteuersystem VZS war als Teilnehmersystem für den Bildschirmbetrieb damals unbedingt notwendig, weist aber heute auf Teile hin, die bei anderen Betriebsystemen verwendet bzw. nach ähnlichen Prinzipien aufgebaut sind. Die Programmierunterstützung in Form von Makros war sicher nicht sehr weit ausgereift, zeigt aber in die Richtung der Schaffung wieder verwendbarer Komponenten. Schließlich stellt das projektierte Wiederanlaufsystem einen wichtigen Beitrag zur Erhöhung der Datensicherheit dar. Das geschaffene System diente in erster Linie der Aufrechterhaltung der Dialogverarbeitung. Bei Batchverarbeitung besteht der Wiederanlauf in dem Neuauflegen des Stapels, ist also nicht mit dem Echtzeitbetrieb zu vergleichen. Mit den aufgezeigten Bestandteilen von DEBAMO soll verdeutlicht werden, dass mit einem derartigen Einzelprojekt sehr wohl bedeutsame Entwicklungen der Informatik aufgezeigt bzw. angeregt werden können.

Literaturverzeichnis

[CHE77] Chen, P.: The Entity-Relationship-Approach, to Logical Data Base Design; Q.E.D. Information Sciences 1977

[COD71] Codd, E., F.: A date base sublanguage founded on the relational calculus; ACM SIGFIDET Workshop on Data Discription; Assec and Control (San Diego-Calif.) 11(1971) Nov. S. 35-68

[GAR72] Garbe, K.: SAL – bildschirmorientiertes Dialogsystem für die Entscheidungsvorbereitung; Schriftenreihe Datenverarbeitung, Verlag Die Wirtschaft Berlin 1972

[GLS76] Günther, M., Liskowsky, R., Steinberg, R.: Vergleich HADES–Bildschirmsysteme – Anwendung von online Technik bei der Produktionslenkung; Rechentechnik/Datenverarbeitung 13(1976) H. 2, S. 22-26

[GOE94] Götze, R.: Dialogmodellierung für multimediale Benutzerschnittstellen; Dissertation Fakultät Informatik der Uni Oldenburg 1993; erschienen im Teubner-Verlag 1994

[GR06] Gräßler, R.: Die Entwicklung von Sachgebietsorientierten Programmiersystemen „SOPS“ des VEB Kombinat Robotron; Lecture Notes in Informatics „Informatik in der DDR – eine Bilanz, Bonn 2006, S. 288ff.

[GUH77] Guhr, H.-D.: Steuersystem für bildschirmorientierten Teilhaber-Betrieb im DOS/ES; Rechentechnik/Datenverarbeitung 14(1977) H. 2, S. 12-14

[IBM71] IBM: Chief Programmer Teams Principles and Procedures; IBM Federal Systems Division, FSC 71-5108, Gaithersburg 1971

[ISO06] DIN EN ISO 9241-110: Ergonomie der Mensch-System-Interaktion; Grundsätze der Dialoggestaltung; Deutsche Fassung EN ISO 9241-110:2006

[KRJ06] Krause, Ch., Jacobs, D.: Von der Schreibmaschine zu Mikrorechnersystemen – Der Beitrag der Mercedes Büromaschinen-Werke/Robotron-Elektronik Zella-Mehlis zur Entwicklung der Rechentechnik in der DDR; Lecture Notes in Informatics „Informatik in der DDR – eine Bilanz, Bonn 2006, S. 166ff.

[LIN83] Liskowsky, R., Nitzsche, L.: Zur Projektierung von Wiederanlaufsystemen für den Echtzeit-Teihaberbetrieb (russ.); Rechentechnik sozialistischer Länder, Moskau Heft 13 1983

[LIS77] Liskowsky, R.: Bildschirmorientiertes Echtzeitsystem für die Materialwirtschaft im DOS/OS; Rechentechnik/Datenverarbeitung 14 (1977) H.11, S. 20-25

[LIS83] Liskowsky, R.: Nutzung elektronischer Datenverarbeitungsanlagen im Echtzeitbetrieb dargestellt am Beispiel der Materialversorgung für die diskrete Fertigung im elektronischen Gerätebau; Habilitation an der TU Dresden 1983

[NS73] Nassi, I., B., Shneiderman: Flowchart Techniques for Structured Programming; in SIGPLAN NOTICES 8 (1973)

[REP99] Rechenberger, P., Pomberger, G.: Informatik-Handbuch (2.Aufl.); Carl Hanser Verlag 1999

[STA74] Stahn, H.: Entscheidungstabellen als Darstellungsmittel abstrakter Automaten; Wiss. Zeitschrift der TU Dresden 23(1974) H. 1, S. 159-162

[WED74] Wedekind, H.: Datenbanksysteme (Teil I und II); B.I. Wissenschaftsverlag Mannheim, Wien, Zürich 1974

Adam Rieses Visionen oder: Der lange Weg zum ESER

Friedrich Naumann

Rilkestraße 33
09114 Chemnitz
friedrich.naumann@phil.tu-chemnitz.de

Abstract: Die Informatik als Wissenschaft entstand unzweifelhaft im Zusammenhang mit der Erfindung und Entwicklung des Computers. Die von Konrad Zuse in den 40-er Jahren des zurückliegenden Jahrhunderts aufgenommenen Arbeiten stehen dafür ebenso, wie jene in England und den USA. Nachdem schließlich erkannt wurde, dass derartige Instrumentarien zur Verarbeitung von Informationen nicht nur „schwachstromtechnischer Begeisterung“ oder „mathematischem Pflichtgefühl“ (Heinz Zemanek, 1955) zu gehorchen hatten, sondern ggf. hohen Nutzen versprachen, gewann die Entwicklung endlich an Fahrt. Retrospektiv wird oft außer Acht gelassen, dass diese initialen Ereignisse eine reiche Vorgeschichte aufzuweisen haben, die ihre Ausprägung insbesondere in der Entwicklung des Rechnens, mithin der Mathematik, und der Entwicklung und Handhabung von Rechenhilfsmitten und Rechenmaschinen erfuhr. Die Computerentwicklung zehrte von diesem Baumaterial, das quasi von vielen Disziplinen verfügbar gemacht wurde und schließlich zu vollkommen andersartigen Bedingungen zusammenfand. Für Sachsen lässt sich eine erstaunliche Vielfalt an derartigen Leistungen nachweisen, beginnend mit der Abfassung populärer Rechenbücher durch Adam Ries und Peter Apian sowie erfolgreichen Versuchen zur Konstruktion von Vierspeziesrechenmaschinen durch Gottfried Wilhelm Leibniz und Jacob Leupold bis hin zur Begründung der deutschen Rechenmaschinenindustrie durch Arthur Burkhardt im erzgebirgischen Glashütte. Nicht zuletzt lieferten die mit Beginn des 20. Jahrhunderts gegründeten sächsischen Rechenmaschinenfirmen das ideelle und technische Know-how für die in den 50-er und 60-er Jahren etablierte moderne Rechen- und Büromaschinenindustrie. Ein Großteil dieses „Erfahrungskapitals“ mündete schließlich in die Entwicklung von Systemen zur Daten- und Informationsverarbeitung, die durch das ESER ihren gesellschaftsbedingten Abschluss fanden. Der Beitrag nimmt die Leistungen der frühen Zeit in den Zirkel, um die historische Bedeutung der Gelehrten, Erfinder und Ingenieure als Voraussetzung für eine moderne, hauptsächlich in Sachsen etablierte, Computerindustrie zu würdigen.

1 Vorbemerkungen

Sachsen, in Carl Christian Schramms Reise-Lexikon von 1744 „die Schatz-Grube Deutschlandes und ein gesegneter Garten des Herrn“ genannt, glänzte in seiner langen Geschichte nicht nur mit kulturellen Spitzenleistungen, sondern brachte auch zahlreiche Gelehrte hervor, die mit „heiliger Neugier“ und glücklicher Hand das weite Feld von Wissenschaft und Technik befruchteten. Sich an diese zu erinnern und ihre Leistungen aus der Sicht der etablierten Informatik zu bewerten, scheint insofern sinnvoll, als keine

der Wissenschaften „vom Himmel fiel“, sondern in ihrer Genese stets auf eine umfangreiche Vorgeschichte zurückzuführen ist. Der Mensch als Akteur im Feld der Wissenschaften sammelt in dieser unbewusst oder bewusst und weiß im entscheidenden Moment über sein Wissen zu verfügen. Thomas Alva Edisons berühmter Spruch „Genius is one per cent inspiration and ninety-nine per cent perspiration” beleuchtet diesen Umstand aus anderer Sicht, trifft aber genau den Kern der Sache. Für den Urquell aller technischen Errungenschaften sind natürlich auch „göttliche Neugier und der Spieltrieb des bastelnden und grübelnden Forschers und nicht minder die konstruktive Phantasie des technischen Erfinders“ vorausgesetzt, wie Albert Einstein bei der Eröffnung einer Rundfunk- und Phono-Schau im Jahre 1930 in Berlin bemerkte. Und da ein Verständnis der gegenwärtigen Wege der Wissenschaft und ihrer Zukunft dringend geboten ist, scheint der Rückblick auf die Vergangenheit fast eine moralische Verpflichtung zu sein. „Savoir pour prévoir“ – also: wissen, um vorauszusehen, lautet das über die Jahrhunderte bestätigte Programm. Heute hat dieses Hochziel aller wissenschaftshistorischen Bestrebungen seine Ausprägung in verschiedenen Disziplinen der Wissenschaftsgeschichte gefunden, die sich nach Kräften mühen, nicht nur Einblicke in das Wechselspiel von Erkenntnis und Irrtum zu finden, sondern auch die Determinanten der Wissenschaftsentwicklung von ihren Anfängen bis zur Gegenwart zu bestimmen und zu dokumentieren.

Für die Informatik, die – nicht ohne Widerspruch verschiedener Geisteswissenschaftler, aber auch Ingenieure – ohne Einschränkung den Technikwissenschaften zugeordnet werden muss, spielen die Entwicklung des Rechnens, also der Mathematik in der Vielfalt ihrer Ausprägungen, und die Entwicklung und Handhabung von Rechenhilfsmitten und Rechen- und Büromaschinen eine entscheidende Rolle. Gleichermaßen muss man jene natur- und technikwissenschaftlichen Disziplinen in die Betrachtung einbeziehen, ohne die der Computer als komplexes und vor Hochtechnologie „strotzendes“ Gebilde nicht denkbar gewesen wäre. Insofern besteht zwischen Adam Ries und dem ESER zwar kein direkter, jedoch gewiss ein wissenschaftshistorischer Zusammenhang, zumal die Vorgeschichte – zumal auf sächsischem Boden – auf zahlreiche einmalige wissenschaftliche und technische Leistungen und deren Repräsentanten verweisen kann, die im Nachfolgenden eingehender beschrieben werden sollen.

2 Sächsische Rechenmeister, Gelehrte und *Mechanici*

Zu alten Zeiten war von Computern noch keine Rede, wenngleich die Mühsal des Rechnens der Entwicklung von Rechenhilfsmitteln beizeiten aufgeholfen hat, eingeschlossen die kluge Handhabung von Arithmetik und Geometrie. An Schulen und Universitäten gedieh gleichermaßen die mathematische Wissenschaft, obwohl noch im Rahmen der *Septem artes liberales*. Immerhin umfassten diese bereits Vorlesungen über die Sphäre, den Algorithmus, die Perspektive, die Latitudines und Arithmetik, zudem Disputationen über mathematische Gegenstände. Anders hingegen die fördernde Praxis, die sich als treibende Kraft für angewandte Rechenkunst und innovatives mathematisches Gedankengut erwies. Technische Lösungen waren demgegenüber noch selten; denn es gelang zunächst nur in bescheidenem Umfang, mathematische Strukturen durch artifizielle abzubilden. Einfache Speicher – der Strich auf der Tafel, die Kerbe im Holz, das abgelegte Steinchen, der Knoten im Faden, auch das schriftlich Festgehaltene – bildeten den

Anfang. Vorteilhafter hingegen zeigte sich der traditionelle Abakus, in seiner konstruktiven Form dem dezimalen, im Sächsischen auch dem staatlich sanktionierten römischen Zahlensystem angepasst. Einfachste Rechenoperationen, aber auch die Handhabung des Stellenwertsystems, waren damit leicht zu vermitteln; später beherrschte man auch Höheres. Für die Anwendung solcherart Rechenhilfsmittel empfahlen sich vor allem jene Regionen, die vielfältigen Bedarf aufzuweisen hatten, wo Handel und Gewerbe weit fortgeschritten waren, der davon beeinflusste Grad an Bildung bereits einen ansehnlichen Umfang angenommen hatte. Für Sachsen ist ein derartiges Niveau unverkennbar, wobei der seit dem 12. Jahrhundert umgehende Silbererzbergbau von größtem Einfluss war.

Inwieweit Albert von Sachsen (eigentlich: A. von Rickmersdorf; auch: A. von Helmstedt; *Albertus de Saxonia*, um 1316-1390), nominalistischer Philosoph und Naturforscher, bereits zur Frühgeschichte beigetragen hat, ist schwer nachvollziehbar. Immerhin avancierte er nach Studien in Prag und Paris zum Rektor der Sorbonne und übernahm 1365 das erste Rektorat der Universität Wien. Neben zahlreichen philosophischen Traktaten trat er auch mit mathematischen Arbeiten hervor, so z. B. über Proportionen, die Quadratur des Kreises, das Verhältnis der Diagonale eines Quadrates zu seiner Seite u. a. Seiner Mitwirkung ist es zu verdanken, dass sich die Wiener Universität bald zu einer „mathematischen" entwickeln konnte. Mehr als an anderen Einrichtungen wurden hier mathematische Vorlesungen angeboten – so z. B. über die Sphäre, den Algorithmus, die Perspektive, die Latitudines und über Arithmetik. Neben den *Disputationes* über philosophische Gegenstände waren auch solche über mathematische im Angebot.

Johannes Widmann (um 1469 - nach 1498) wurde zwar in Eger geboren, erwarb aber 1485 an der Leipziger Universität den Magister und lehrte hier bis etwa 1490. In dieser Zeit machte er Leipzig zu einem Zentrum der Mathematik. Wahrscheinlich verzog er dann nach Annaberg, um eine Lateinlehrerstelle anzunehmen. Bekannt wurden seine Algebravorlesungen – die ersten an einer deutschen Universität. Sie gingen als „Lateinische Algebra" in die Mathematikgeschichte ein und wurden zur Grundlage der „deutschen Coß". In diesem Zusammenhang festigte sich die Anwendung von Symbolen, um Potenzen der algebraischen Unbekannten und ihres Quadrats auszudrücken, dazu gehörte auch der Gebrauch der Zeichen + und -. Erstmals finden sich diese in Widmanns Rechenbuch „Mercantile Arithmetic oder Behēde vnd hubsche Rechenung auff allen kauffmanschafft" (Leipzig 1489): „was – ist dz ist minus" und „das + ist mer". Zu den zahlreichen, wenngleich undatierten und anonym erschienen Schriften zählen auch *Algorithmus Integrorum Cum Probis annexis, Algorithmus Minutiarum Vulgarium, Algorithmus Minutiarum Phisicarum, Tractatus Proportionum plusquam Aureus* und *Regula Falsi apud Philozophantes Augmenti et Decrementi appellata*.

Wenig später machte ein weiterer Mathematiker von sich reden: der aus Regensburg stammende Andreas Alexander. 1493 wurde er an der Leipziger Universität immatrikuliert. Als *Magister artium* vertrat er die Fächer *arithmetica communis, mathematica, perspectiva communis* sowie Euklidische Geometrie. Er gab auch zwei Schriften heraus: Johann Pechams *Perspectiva communis,* ein Standardlehrbuch über die geometrische Optik, und das *Mathemalogium*, in dem er mathematische Begriffe und Probleme der aristotelischen Schriften erläutert, z. B. die Inkommensurabilität und das Verhältnis zwischen geraden und krummen Gebilden.

Von beiden Gelehrten ging unzweifelhaft ein großer Einfluss auf jenen Mann aus, des-

sen Name noch heute in aller Munde ist: Adam Ries (1492-1559). Unzweifelhaft verkörperte der im fränkischen Staffelstein geborene „Zahlenhexer vom Römerberg“ – so sein ehrender Nimbus aus der Frankfurter Zeit, seit den 1520-er Jahren im prosperierenden erzgebirgischen Annaberg sesshaft – ein außergewöhnliches Maß an mathematischer Begabung. Das sicherte ihm einerseits das Wohlwollen seines Landesherren; denn nicht grundlos ernannte ihn Herzog Georg der Bärtige 1539 zum „Churfürstlich Sächsischen Hofarithmeticus“ und anerkannte damit gewissenhafte Diensterfüllung als „Bergmann von der Feder“. Andererseits erlangte Ries als Rechenmeister und Autor hervorragender Rechenbücher überregionale Bedeutung, so dass selbst Berufskollegen ihren Respekt nicht verhehlen konnten. Im Jahre 1518, Ries wohnte noch in Erfurt, verfasste er das erste Buch zum „Linienrechnen“[1], die größte Verbreitung erlangte jedoch das Buch zum „Ziffernrechnen“[2]; denn es erschien nach seiner Vorstellung 1522 in mindestens 118 Nachauflagen als mathematischer Bestseller jener Zeit. „Vom Leichten zum Schweren / Vom Einfachen zum Zusammengesetzten / Von der Anschauung zum Begriff, (um) alles auf das clerlichste und einfeltigste auszudrücken“, lautete die Devise, mit der Ries die Rechenkunst „dem gemeynen mann nutzlich zu machen“ wusste. Mit der „Practica“[3], durch ein Privileg Kaiser Karl V. gegen missbräuchlichen Nachdruck geschützt, setzte Ries nochmals Maßstäbe; und wer den Inhalt dieses Buches beherrschte, galt als Meister seines Faches. David Smith urteilt deshalb nicht grundlos: „The book represents the culmination of Riese’s work, and is the best exponent of the practical arithmetic of the middle of the century in Germany“.[4] Große Bedeutung erlangte auch das lange verschollene Lehrbuch der Algebra, die in Deutsch abgefasste und nach der Unbekannten (Variablen) bezeichnete „Coß“ – seit 1992 in transkribierter und gedruckter Form vorliegend.[5]

Ob Ries auch von einer Maschine zum Rechnen träumte, ist nicht zu belegen, wenngleich die hochentwickelte Mechanik der im Erzgebirge entwickelten Bergmaschinentechnik vielfältigste Lösungen mit Vorbildcharakter bereithielt und derartiges durchaus im Bereich des Möglichen gewesen wäre. So hielt er besser an der klassischen Technik des Abakus fest, passte sie allerdings an die Gegebenheiten des römischen Ziffernsystems an, die Existenz zusätzlicher Fünferwerte berücksichtigend. Rechentisch und Rechentuch bildeten auf diese Weise das Grundgerüst, die aufzulegenden Rechenpfennige bzw. Rechensteine die Variablen; ohne materielle Basis – da im Kopf des Rechnenden resistent – blieb hingegen der Lösungsweg, diesen musste noch jeder selbst bestimmen.

Neben Ries hatte Annaberg noch weitere Rechenmeister; nachzuweisen sind Heinrich Müller, Johann Dehm sowie Rieses Söhne Adam d. J., Abraham und Jacob. Über fast ein Jahrhundert prägten sie entscheidend die Entwicklung der Mathematik im ganzen Lande.

Frühe Belege zur Rechentechnik lassen sich auch bei Rieses Zeitgenossen Georgius

---

[1] Rechnung auff der linihen // gemacht durch Adam Riesen vonn Staffel- // steyn / in maßen man es pflegt tsu lern in allen // rechenschulen gruntlich begriffen anno 1518.

[2] Rechnung auff der linihen // vnd federn in zal / maß /vnd gewicht auff // allerley handierung / gemacht vnnd zu // samen gelesen durch Adam Riesen // vō Staffelstein Rechenmey- // ster zu Erffurdt im // 1522. Jar

[3] Rechnung nach der // lenge / auff den Linihen / vnd Feder. // Darzu forteil vnd behendigkeit durch die Proportio- // nes / Practica genant. / Mit grüntlichem // vnterricht des visierens. // Durch Adam Riese. // im 1550. Jar. //

[4] David Eugene Smith: Rara arithmetica. Boston/London 1908.Nachdruck New York 1970, S. 252.

[5] Vgl. Einblicke in die Coß von Adam Ries. Eine Auswahl aus dem Original mit aktuellen Anmerkungen und Kommentaren. Hrsg. v. Rainer Gebhardt, Leipzig, Zürich 1994.

Agricola (1494-1555) finden. Das berühmte Werk „De re metallica libri XII“[6], ein exzellentes Dokument des frühen Montanwesens und dessen wissenschaftliche Begründung markierend, beschreibt nicht nur den Stand der mathematisch orientierten Markscheidekunde, sondern zeigt bereits den Umgang mit damals gebräuchlichen Kerbhölzern – jenen Rechenhilfsmitteln also, die als einfache, aber fälschungssichere Speichermedien dienten und eine sichere Handhabung des Dezimalsystems belegen.

Der in Leisnig gebürtige Peter Apian (auch Bienewitz, 1495-1552) setzt den Reigen fort. Nach Absolvierung der Lateinschule in Rochlitz ging er nach Leipzig, um bei Stromer von Auerbach (Geometrie) und Petrus Mosellanus zu studieren. Danach wechselte er an die Wiener erste und zweite mathematische Schule, studierte Mathematik, Astronomie, Kartographie und Instrumentenbau und erwarb das Baccalaureat. Hier lernte er auch Conrad Celtis kennen und erlebte die Gründung des „Collegiums der Poeten und Mathematiker“. In besonderem Maße beeinflussten ihn die Gelehrten Stiborius, Stabius, Tannstetter und Aventin. 1520 erschien seine erste „Weltkarte“, 1524 das Buch *Cosmographicus liber*, das eine Weltbeschreibung lieferte und besonders für die Navigation Bedeutung erlangte. Apian befasst sich hierin auch mit dem Magnetismus und bestimmte als einer der ersten experimentell die Missweisung – für Ingolstadt betrug diese 10° 30’. Beiträge für die praktische Geometrie und die Astronomie leistete er durch die Konstruktion von Jakobsstäben, Sonnenuhren, Nachtuhren und Quadranten *(Quadrans Apiani).* Im *Astronomicum Caesareum* beschreibt er 50 von ihm entwickelte drehbare Papierscheibeninstrumente zur Bestimmung der Planetenstände im Tierkreis, der Wochentage und der Finsternisse von 7000 vor bis 7000 nach Chr.

1527 wurde er an die Universität Ingolstadt berufen, wo er 25 Jahre als Lehrer für Mathematik wirkte. Seine in Deutsch abgefasste „Kauffmanß Rechnung“[7] erfuhr sieben Auflagen und verlieh der Mathematik wesentliche Impulse. Auf der Titelseite ist die erste in Westeuropa gedruckte Darstellung eines binomischen Dreiecks zu sehen. Das Werk findet sich auch auf dem bekannten Gemälde „Die Gesandten“ von Hans Holbein d. J. Herzog Georg berief ihn schließlich nach Sachsen und ersuchte ihn, eine „Mappierung“ durchzuführen, womit Sachsen als erstes deutsches Land eine präzise Karte erhalten hätte; daraus wurde allerdings nichts. Apian, von seinem Zeitgenossen Sebastian Münster achtungsvoll „aus Meißen“ benannt, hielt sich jedoch zugute, bis zu seinem Lebensende der „Petrus Apianus von Leysnigk“, im Herzen also ein Sachse zu bleiben.

Wenngleich auch Gottfried Wilhelm Leibniz (1646-1716) Sachsen beizeiten den Rücken kehrte, blieb er doch zeitlebens seiner Heimat treu verbunden. Die Grundlagen der Bildung wurden in seinem Geburtsort Leipzig gelegt. Nicolaischule und Universität, wo er 1653 immatrikuliert wurde, jedoch aufgrund seines Alters von 18 Jahren noch nicht promoviert werden konnte, schufen das wissenschaftliche Grundgerüst. Nach mehr als 40 Jahren, auf dem Gipfel seiner Karriere stehend, äußerte er rückblickend in einem Brief an den Theologen Adam Rechenberg: „Ich liebe Leipzig, wie es sich für die Heimat geziemt, und habe nicht das Gefühl, dass sie gegen mich undankbar war. Ich habe keinen Grund zur Klage darüber, dass ich als junger Mann und fast noch ein Knabe unter

[6] Ediert beim Verlag Froben, Basel 1556. Vgl. dazu auch F. Naumann: Georgius Agricola – Berggelehrter, Naturforscher, Humanist. Sutton-Verlag Erfurt 2007.

[7] Eyn Newe Vnnd wolgegründte vnderweysung aller Kauffmanß Rechnung in dreyen büchern/mit schoenen Regeln und fragstucken begriffen. Ingolstadt 1527.

so vielen Männern, die an Alter und Gelehrsamkeit hervorragten, nicht auffiel. Dennoch reut mich meine Ungeduld nicht. Die Irrtümer der Menschen werden durch die göttliche Vorsehung gelenkt, so dass oft schlechte Entschlüsse zum Guten führen“.[8]

Aus informatikhistorischer Sicht beeindrucken vor allem zwei Abhandlungen, mit denen Leibniz zur Konstitution der mathematischen Grundlagen der Informatik beitrug, es sind dies *Dissertatio de arte combinatoria* und *De dyadices*.[9] In ersterem finden sich die ersten Andeutungen zur „Erfindungskunst“ *(ars inveniendi et dijudicandi),* die er nicht nur auf den Bereich der höheren Analysis beschränkte, sondern auch zu einem *Instrumentum panarithmicon* führen wollte. Mehrere Jahre arbeitete er deshalb an der Entwicklung einer Vierspezies-Rechenmaschine. Die ersten Entwürfe zu dieser „lebendigen Rechenbanck“ waren 1672 fertiggestellt und wurden in Paris und London vorgestellt. Sie wiesen zwei bedeutsame Erfindungen auf: die Staffelwalze und den Schaltwerksschlitten zur Dekadenverschiebung, Voraussetzung für eine mehrstellige Multiplikation. Eine grandiose Idee, wie sich zeigen sollte, den diese Maschinenelemente wurden von vielen nachfolgenden Erfindern übernommen und hielten sich über 250 Jahre im mechanischen Rechenmaschinenbau. Nach 1694 wurde schließlich eine fünfte, die „jüngere“ Maschine, fertiggestellt; diese blieb glücklicherweise erhalten und befindet sich heute in der Niedersächsischen Landesbibliothek in Hannover. *De dyadices* beschreibt erstmals in der Geschichte das binäre Rechnen, die *Dyadik* – heute elementare Grundlage der gesamten digitalen Informationstechnik. Weitere Überlegungen dazu finden sich in der Abhandlung *Explication de l'Aritmétique Binaire* (1703). Auf dieser Grundlage skizzierte Leibniz sogar eine binäre, also mit Dualzahlen arbeitende, Rechenmaschine, die zudem die logischen Verknüpfungen Und, Oder und Negation realisieren sollte. Sein Vorschlag, diese mittels Kügelchen und Rinnen zu fertigen, wurde offensichtlich nie realisiert.

Der binäre Ansatz führte gar zu einer rein philosophischen Betrachtung, nämlich dass die Welt aus zwei gegensätzlichen Kategorien, dem Sein und dem Nichts, besteht. Einer der biblischen Grundlehren entsprechend, „ist die erschaffung aller dinge aus nichts durch die allmacht Gottes: nun kann man wohl sagen daß nichts in der welt sie beßer vorstelle, ja gleichsam demonstrire, als der ursprung der zahlen wie er alhier vorgestellet, durch deren ausdrückung bloß und allein mit Eins und Null oder Nichts, und wird wohl schwehrlich in der Natur und Philosophi ein beßeres vorbild dieses geheimnißes zu finden seyn“. An gleicher Stelle heißt es, dass die *Dyadik* noch weitaus mehr zeigt: Anders als das Dezimalsystem ist sie auch Symbol der Vollkommenheit der Welt.[10] Die Überlegungen verdeutlichen Leibniz' Verständnis zum dualen Rechnen. Er war damit seiner Zeit weit voraus; denn konkrete Anwendung fand das Binärsystem erst in den 30-er Jahren des 20. Jahrhunderts. Schließlich kann man seine Prognose in der Abhandlung *De scientia universali seu calculo philosophico* (um 1680) fast als einen Antizipation des modernen Computers auffassen: „Es wird dann beim Auftreten von Streitfragen für zwei Philosophen nicht mehr Aufwand an wissenschaftlichem Gespräch erforderlich sein als für zwei Rechnerfachleute. Es wird genügen, Schreibzeug zu Hand zu nehmen, sich vor

[8] K. Müller/G. Krönert: Leben und Werk von G. W. Leibniz. Eine Chronik. Frankfurt a. M. 1969, S. 208.
[9] Vgl. Leibnizens mathematische Schriften. Hrsg. v. C. I. Gerhard. Halle, Bd. I (1858), S. 1-7, Bd. III (1863), S. 228-234.
[10] Aus dem sogenannten Neujahrsbrief vom 12. Januar 1697 an Herzog Rudolf August von Wolfenbüttel. In: G. W. Leibniz: Sämtliche Schriften und Briefe. Hrsg. v. d. Preußischen Akademie der Wissenschaften. Darmstadt 1923 f., Bd. I, 13, S. 117.

das Rechengerät zu setzen und zueinander (wenn es gefällt, in freundschaftlichem Ton) zu sagen: Laßt uns rechnen."[11] Unter diesen Umständen verwundert es nicht, dass Norbert Wiener, Begründung der Kybernetik, Leibniz ob seiner fundamentalen Arbeiten gar zum „Schutzpatron" dieser modernen Wissenschaft ernannte.

Sehr viel dauerhafter etablierte sich in Leipzigs Mauern ein Mann, der als erster Historiograph der Rechentechnik in die Geschichte eingegangen ist: der *Mathematico* und *Mechanico* Jacob Leupold (1674-1727). Geboren und aufgewachsen in Planitz bei Zwickau und akademisch gebildet in Jena, Wittenberg und Leipzig, begann er seine Karriere mit dem Unterrichten von Zimmerleuten und Maurern in Schreiben und Rechnen, Grundlagen der Geometrie (z. B. nach den *Elementis Euclidis*) und der Zivilbaukunst. Auf Anregung des Pastors Seeligmann, der meinte, „Leipzig habe Prediger genug, aber keine Künstler, deren Wissenschaft auf mathematischen und physikalischen Gründen beruht"[12], gründete er eine „Mechanische Fabrique" und begann mit der Fertigung von Globen, Quadranten, Sonnenuhren, Zirkeln, Messketten, Mensuren und anderen Gerätschaften. Zugleich versuchte er, die wichtigste Literatur zu allen „mechanischen Künsten" zusammenzutragen und aufzuarbeiten, wofür die Kommunikation mit bekannten Persönlichkeiten, wie dem Hallenser Mathematiker Christian Wolff, G. W. Leibniz und Friedrich August I. (August der Starke), hilfreich war.

Im Ergebnis entstand ein mehrbändiges Kompendium zum Instrumenten- und Maschinenwesen, das berühmte *Theatrum machinarum*. Es erschien in den Jahren 1724 bis 1727 im stattlichen Umfang von acht großen Foliobänden, enthaltend 1528 Textseiten und 436 Kupfertafeln. Der Ruf dieses Werkes, das Peter I. (der Leupold gern nach St. Petersburg geholt hätte, wegen dessen Krankheit jedoch verzichten musste) sofort ins Russische übersetzen ließ, war so ausgezeichnet, dass James Watt noch im hohen Alter die deutsche Sprache erlernte, um die Werke im Original lesen zu können. Für die Informatikgeschichte ist vor allem das *Theatrum arithmetico-geometricum*[13] von Bedeutung, da es erstmals die Entwicklung verschiedener Rechenmethoden, Rechenhilfsmittel und -maschinen sowie von Zeichengeräten, geodätischen Instrumenten und gebräuchlichen Maßen aus der Perspektive von Herstellung und Anwendung dokumentiert .

Dementsprechend finden sich hier auch die Arbeiten all jener Erfinder, die vor ihm „Maschinen zum Rechnen" entwickelt hatten: John Napier, Blaise Pascal, René Grillet, Johannes Poleni und Leibniz; Wilhelm Schickards Erfindung aus dem Jahre 1623 blieb ihm allerdings verborgen. Wahrscheinlich sah sich Leupold davon inspiriert; denn er entwickelte zunächst eine auf den Napierschen Rechenstäbchen basierende, verbesserte Maschine, genannt „Die Leupoldische Rechen-Machine mit Abtheilung der Neperianischen Staebgen". Sehr viel anspruchsvoller war jedoch „Die Leupoldische oder des Autoris curieuse und gantz neue Rechen-Machine" – eine innovative Konstruktion, bei der sich erstmals alle Maschinenelemente konzentrisch zueinander angeordnet finden. Die Zahl der Reibung verursachenden Zahnräder bzw. Antriebe war damit deutlich geringer

[11] Zit in: F. L. Bauer et al.: Informatik. Eine einführende Übersicht. Berlin, Heidelberg, New York 1973, S. 179.

[12] M. G. Tettelbach: Die letzte schuldige Freundschaft – und Ehrenbezeugung wolte dem Weiland Hochedlen, Vesten und Hochwohlgelahrten Herrn, Herr Jacob Leupolden […] Harpeter, Dresden 1727.

[13] Theatrum arithmetico-geometricum. Das ist: Schau-Platz der Rechen- und Meß-Kunst, Darinnen enthalten Dieser beyden Wissenschafften nöthige Grund-Regeln und Handgriffe so wohl, als auch die unterschiedene Instrumente und Machinen [„„] Leipzig 1727.

und das Problem des Zehnerübertrags besser gelöst. Bereits im Jahr 1720 erscheint sie in einem „CATALOGUS macherley Machinen und Instrumenten“ unter der Rubrik „Neu inventirte und verbesserte Machinen und Instrumenta“; neben Waagen, Zirkeln, Maßstäben, Linealen steht sie hier zum Preis 200 Talern als „Machina Arithmetica, oder Machine zum Rechnen, da ein jeder, der nur zu numeriren weiß, ohne einmahl eins alle Species mit grosser Behendigkeit und Accuratesse tractiren, und die größten Summen behende ausrechnen kann“ – ein deutlicher Hinweis auf die wirkliche Existenz.

Leupolds Werk zur Rechen- und Messkunst fand nach seinem Tode mit dem *Supplementum*[14] die angekündigte Ergänzung. Beide Werke gaben nicht nur zeitgenössischen Vertretern der *praxis geometriae*, sondern auch späteren Rechenmaschinenerfinder, wie Antonius Braun, Philipp Matthäus Hahn, Johann H. Müller, Christel Hamann und Curt Herzstark, Anregung und Unterstützung. Ihre Maschinen folgten nicht nur in der äußeren Form, sondern auch in mechanischer Hinsicht der Idee des konzentrischen Aufbaus.

Genannte Erfindungen blieben allerdings allesamt Unikate, selbst Leupolds überzeugende Offerte blieb in der Angebotsvielfalt praktischer Instrumente und Gerätschaften zunächst unbeachtet, weckte also keinerlei gesellschaftlichen Bedarf. Dies änderte sich erst zu Beginn des 19. Jahrhunderts – zunächst im hochentwickelten Frankreich.

3 Aufbruch in die industrielle Fertigung

In dieser „Sturm- und Drangperiode der Produktion“ erfand Charles-Xavier Thomas aus Colmar im Elsaß eine Rechenmaschine, die ob ihrer sicheren Funktion nicht nur zu serienmäßiger Herstellung taugte, sondern auch der Einführung neuer dezimaler Maß- und Gewichtssysteme entgegenkam. Das Patent wurde bereits 1920 erteilt.[15] Staffelwalze und verschiebbarer Schlitten, ähnlich der Leibnizschen und Hahnschen Konstruktionen, bildeten die technische Besonderheit der *Arithmomètre*, die rasch bekannt wurde und Thomas zu hohen Auszeichnungen verhalf.[16] Anfangs wurden davon lediglich 15, später schon 100 Maschinen pro Jahr gefertigt. Bis 1871 verließen 1500 Maschinen das Pariser Werk. Zu den Abnehmern zählten englische und amerikanische Firmen, vor allem jedoch Versicherungsgesellschaften, Rentenanstalten und Behörden, aber auch Wissenschaftler – Ingenieure, Physiker, Astronomen und „Arithmetiker“. Aus heutiger Sicht war die Erfindung ein Meilenstein der Technikgeschichte; denn formalisierbare geistige Arbeit konnte erstmals durch eine Maschine substituiert werden.

In Deutschland fand diese Maschine zunächst nur zögernd Verbreitung, was den Berliner Kinematiker Franz Reuleaux veranlasste, 1862 entsprechende Einzelheiten in den Zeitschriften „Civilingenieur“ (Bd. 8) und „Dinglers Polytechnisches Journal“ (Bd. 165) zu veröffentlichen. Noch im gleichen Jahr erschien sein Buch „Die sogenannte Tho-

[14] Theatri Machinarum Supplementum. Das ist Zusatz zum Schau-Platz der Machinen und Instrumenten […] Leipzig 1739.

[15] „(Patent) Hr. C. X. Thomas zu Paris, d. 18. November (1820) für 5 Jahre auf eine Maschine zu allen arithmetischen Operationen, die er Arithmomètre nennt.“ In: Bulletin de la Societe d’ Encouragement pour l’Industrie nationale. Nov. 1822, S. 355-365.

[16] Die „Aufmunterungsgesellschaft“ verlieh ihm die Goldmedaille, er wurde Mitglied der Ehrenlegion, selbst der Pabst anerkannte Thomas’ Leistungen.

mas'sche Rechenmaschine"[17], mit dem er die „fast gänzliche Unbekanntschaft mit diesem geistreichen mathematischen Werkzeug" beenden wollte: „Fast keine geistige Erschlaffung ist so groß wie diejenige nach tagelang fortgesetzter Beschäftigung mit dem Abstraktum der Zahlen, namentlich in höheren Rechnungsarten, und bekannt ist es, dass es bis jetzt nicht menschenmöglich gewesen ist, Logarithmentafeln fehlerfrei zu rechnen, indem der menschliche Geist sich entschieden zu sträuben scheint, in die unbeugsame Maschinenmäßigkeit einzutreten, welche dafür gefordert wird. Es musste deshalb eine Erfindung mit Freuden begrüßt werden, welche die Sklaverei des Rechnens zu brechen gekommen ist".[18] Dieser Appell eines einsamen Propheten, an Maschinen-, Bau-, Berg- und Militäringenieure, Physiker, Astronomen, Zahlenmathematiker und Statistiker in Versicherungsgesellschaften, Rentenanstalten sowie an Steuer- und Finanzbeamte gerichtet, sollte nicht allzu lange ungehört bleiben: In den folgenden Jahrzehnten begann man auch in Deutschland mit der serienmäßigen Fertigung von Rechenmaschinen.

Die Initiative dazu kam aus dem osterzgebirgischen Glashütte, bekannt insbesondere durch den Namen des Uhrmachers und späteren Unternehmers Ferdinand Alfred Lange. Lange brachte, nachdem er 1845 in seiner dortigen Werkstatt mit der Ausbildung von 15 Lehrlingen begonnen hatte, innerhalb nur einer Generation die feinmechanische Präzisionsarbeit auf ein hohes Niveau, so dass bald versierte Facharbeiter zur Verfügung standen, die sich hervorragend auf die Fertigung von Rädern, Schrauben, Trieben, Zeigern, Gehäusen usw. verstanden. Der Ingenieur Carl Dietzschold (1852-1922), Absolvent des Karlsruher Polytechnikums, wusste dieses Potential zu nutzen und errichtete 1877 eine mechanische Werkstätte in der Absicht, die Fertigung von Rechenmaschinen aufzunehmen. Die ersten Resultate wurden – wenngleich wenig überzeugend – dem Preußischen Statistischen Amt vorgestellt.[19] Dietzscholds Wirken währte jedoch nur kurz, da er zum Direktor der k. k. Fachschule für Uhrenindustrie nach Karlstein/Niederösterreich berufen wurde. Er übergab deshalb die Werkstatt an seinen Studienfreund Arthur Burkhardt (1857-1918), der die Arbeiten weiterführen sollte. In einem Brief schrieb er: „Nun, offen gestanden, ich dachte Dir die konstruktive Durchbildung der Rechenmaschine zu überlassen, und würde es sich dabei zeigen, ob Du für diesen Mechanismus gebaut bist. Es gibt nichts Schwereres und Raffinierteres, als diesen Apparat, aber ein Bildungsmittel der Getriebelehre wie kein zweites. Ich glaube aber, dass Du Dich leicht hineinfinden würdest, da Du ja in den Wirkereimaschinen aufgewachsen bist."[20]

Burkhardt nahm die gebotene Chance wahr und gründete 1878 die „Erste Deutsche Rechenmaschinenfabrik". Da er kein Risiko eingehen wollte, entschloss er sich zunächst, die relativ ausgereifte Thomas'sche Maschine nachzubauen, die er am Preußischen Statistischen Amt kennengelernt hatte. Dies war für ihn insofern unproblematisch, als es in Sachsen noch nach Einführung des Reichspatentgesetzes von 1877 die Möglichkeit gab, mittels eines sogenannten Einführungspatentes ausländische Erfindungen gegen Zahlung einer Gebühr zu übernehmen. Auch erbat er sich dafür das Wohlwollen des sächsischen Königs Albert. Der zunächst unveränderte Nachbau hinderte ihn nicht daran, seine

[17] Franz Reuleaux: Die sogenannte Thomas'sche Rechenmaschine. Zürich 1862.
[18] Ib., S. 3.
[19] Ein Exemplar der Dietzschold'schen Maschine ist noch erhalten geblieben; es befindet sich im Mathematisch-Physikalischen Salon des Dresdener Zwingers.
[20] Geschichte der Gründung der deutschen Rechenmaschinen-Industrie, im besonderen die Geschichte der Ersten Glashütter Rechenmaschinenfabrik Arthur Burkhardt, Ing. in Glashütte. O. O., 1913, S. 6.

„Burkhardt-Arithmometer“ genannte Maschine als „anerkannt bestens arbeitende Rechenmaschine“ anzupreisen, obwohl spezielle Erfahrungen zur Fertigung kaum vorlagen und die Beherrschung unterschiedlichster Materialien wie auch diskontinuierlich arbeitender Getriebeelemente noch recht unvollkommen war. Doch das französische Vorbild war robust und funktionssicher, so dass sich die Anfangsprobleme in Grenzen hielten. Gigantische Absätze waren zunächst nicht zu verzeichnen, so dass sich Burkhardt genötigt sah, auch anderes herzustellen: Bleistiftspitzer, Patentgeigen, Cellowirbel, Hubzähler, „Karthographen“ sowie elektrische Artikel. 1884 verlegte er kurzerhand für zwei Jahre den Wohnsitz nach Braunschweig, um sich im Bau von Harfen zu versuchen. Da auch dies wenig erfolgreich war, kehrte er nach Glashütte zurück, um die Rechenmaschinenfertigung fortzusetzen. Mittlerweile hatte sich durch die neue Reichsversicherungsgesetzgebung und die damit geschaffenen Rechenämter ein vielfältiges Absatzfeld aufgetan, so dass Burkhardt einer gesicherten Menge von Aufträgen entgegensehen konnte. Die Kalkulation für seine Maschinen lag – je nach Ausführung (sechs-, acht- oder zehnstellig) – zwischen 350 und 600 Reichsmark. Da konkurrierende Unternehmen in Deutschland noch nicht existierten, außerdem auch Reparaturen von Thomas-Maschinen durchgeführt wurden, genügte ein Jahrzehnt, um sich erfolgreich zu etablieren. Reuleaux, der die Entwicklung mit Aufmerksamkeit verfolgte, lobte deshalb 1892 die „Tüchtigkeit, mit der die Burkhardtsche Werkstätte die Maschine ausführt“, die „Vorzüglichkeit der Herstellung (wie es bei der französischen Maschine nicht erreicht worden war)“, nicht zuletzt auch die „Güte der Erzeugnisse seiner Werkstätte“.[21]

In technischer Hinsicht wurde das „Thomas-Burkhardt-System“ zunächst nur geringfügig verändert, allerdings trug es in zunehmendem Maße unterschiedlichen Kundenwünschen Rechnung und wurde deshalb mit „gruppenweiser Färbung der Einstellhebel, Antriebshebeln mit Postenzählern, doppelten Zählwerken für Fakturenrechnung, Rabatt- und Zinsrechnung“ usw. ausgestattet. Im Gegensatz zu anderen Produzenten, die sich nach und nach etablierten und Erfahrungen der amerikanischen Massenproduktion zu nutzen wussten, blieb Burkhardt dem Konventionellen verhaftet. Seine Maschinen wurden nur auf Bestellung in Kleinserie gefertigt, und die Vertriebsorganisation behielt ihren Sitz in Glashütte. Trotzdem verzeichnet die 1913 verfasste Bilanz einen „Absatz in alle Kulturstaaten der Welt“ sowie eine Vielzahl nationaler und internationaler Auszeichnungen. Das Ritterkreuz I. Klasse des Königlich Sächsischen Albrechtsordens bedeutete für Burkhardt eine ganz besondere Ehre, anerkannte es doch auch seine Verdienste um die Hebung der sächsischen Wirtschaft. Bald etablierten sich in Glashütte, vielfach durch Ausgründungen, weitere Firmen, deren Maschinen unter den Namen „Saxonia“, „Archimedes“ und „Cordt“ den traditionsreichen Standort europaweit bekannt machten. Auch nach Ende des Zweiten Weltkrieges konnte die Produktion fortgesetzt werden, und bei Produktionseinstellung der nun volkseigenen Werke im Jahr 1960 verzeichnete die Bilanz der Glashütter Rechenmaschinenproduktion die Entwicklung von 42 Maschinentypen und die Lieferung von 85.000 Maschinen in 27 verschiedene Staaten**.** Die noch junge Computer-Industrie der DDR griff, die reichen Erfahrungen des Standortes nutzend, das Potential auf und fertigte in Glashütte sowohl Analogrechner (Typ „endim 2000“), Schreibmaschinen (Typ „Erika“), seit 1987 dann auch Achtfarben-Flachbettplotter (Typ „robotron K 6411“).

[21] Franz Reuleaux: Die Thomas’sche Rechenmaschine. Leipzig 1892 (2. Auflage), Vorwort.

Glashütte blieb zunächst alleiniger Produktionsstandort, erst ein Vierteljahrhundert später etablierten sich in anderen Städten weitere Produzenten. Hauptsächlich waren dies Dresden (1906 Seidel & Naumann, 1912 Hans Sabielny, 1914 Schubert & Richter, 1920 Clemens Müller, 1926 Franke & Richter), Leipzig (1903 Triumphator, 1910 Bordt & Behrens, 1914 Lispsia, 1923 Kurth) und Chemnitz (1904 Schubert & Salzer, 1916 Wanderer-Werke Continental, 1921 Astra-Werke).

Die Chemnitzer Entwicklung ähnelt insofern der Glashütter, als sie in ihrem Anfangsstadium gleichermaßen durch den Transfer einer – in diesem Falle amerikanischen – Innovation beeinflusst wurde, die sich als äußerst folgenreich erweisen sollte. Nachdem die Wanderer-Werke in Schönau bei Chemnitz, zunächst hauptsächlich auf Fahrzeug- und Fräsmaschinenbau orientiert, 1902 die Entwicklung von Schreibmaschinen (Warenzeichen „Continental“) aufgenommen hatten, entschloss man sich auch zum zukunftsträchtigen Rechenmaschinenbau und engagierte dafür einen gewissen John E. Greve (1880-1967). Der in Hamburg-Altona Gebürtige hatte das Handwerk des Addier- und Buchungsmaschinenbaus durch mehrjährige Arbeit bei der Dalton Adding Machine Company in Poplar Bluff (Missouri, USA) erlernt und dabei auch Einblick in die hochentwickelte Massenproduktion erhalten. Die Firma Dalton war nach 1900 mit der von Hubert Hopkins konstruierten ersten druckenden Addiermaschine auf den Markt getreten, sie verfügte über lediglich 10 Tasten zur Eingabe der Werte und arbeitete sehr sicher. Gegenüber der bislang üblichen und bis zu 100 Tasten umfassenden Volltastatur (z. B. der Burkhardtschen Maschine) versprach solcherart Einfachtastatur großen Erfolg. Greve wollte sie deshalb in Deutschland weiterentwickeln und auf den Markt bringen.

Die hochentwickelte feinmechanische Industrie in der Industriemetropole Chemnitz war für ihn Veranlassung, in den dortigen Wanderer-Werken 1910 eine Arbeit als Konstrukteur aufzunehmen. Zunächst konzentrierte er seine Arbeit auf die Entwicklung einer Addiermaschine, wobei ihm vorschwebte, diese mit genannter Einfachtastatur nach amerikanischem Vorbild auszustatten. Diese Idee fand jedoch nicht das Wohlwollen der Direktion, denn noch war man voll und ganz von den Vorzügen der bereits in vielen Maschinen verwendeten Volltastatur überzeugt. So entstand in den Jahren bis 1914 zunächst eine schreibende Addier- und Subtrahiermaschine mit Volltastatur. Aus technischer Sicht war dies zwar ein Novum auf dem europäischen Mark. Greve befürchtete jedoch, diese Maschine ließe sich nicht zweckmäßig bauen, da es hauptsächlich an entsprechenden Fabrikationseinrichtungen, Präzisionsmaschinen und Messgeräten fehlte. Die Zweifel waren nicht berechtigt, denn man vermochte durchaus, die Probleme zu lösen, so dass am 16. November 1916 die erste Continental-Addiermaschine ausgeliefert werden konnte. 1919 verließ Greve verärgert das Werk, um die seit langem gehegte Idee einer druckenden Addiermaschine mit Einfachtastatur umzusetzen. Mit Unterstützung einiger Mechaniker gelang es ihm, die bislang mit nur einer Null-Taste versehene Einfachtastatur insofern zu erweitern, als er zusätzlich eine Zwei-Null- und eine Drei-Null-Taste entwickelte. Diese sinnvolle Erfindung vereinfachte die Zahleneingabe ganz erheblich.

1921 gründete Greve die „Astra-Werke AG“ – Senecas *Per aspera ad astra* schien hierfür den rechten Weg zu weisen. Bankdirektoren, Rechtsanwälte und Fabrikbesitzer unterstützten ihn und brachten das erforderliche Kapital in Höhe von 3,5 Millionen Mark auf, um so den Aufstieg dieses *Newcomers* auf dem Rechenmaschinenmarkt zu befördern. Die einfach zu bedienenden und funktionssicheren Maschinen entsprachen genau

dem stetig steigenden Rechenaufwand in Sparkassen und Banken, Finanz-, Reichs-, Staats- und Landesbehörden, so dass der geschäftliche Erfolg nicht ausblieb. In nur wenigen Jahren erlangte die Firma Weltruf, zumal die Produktion nach internationalen Bedürfnissen ausgerichtet wurde und bald auch ein umfangreiches Spektrum von Buchungsmaschinen umfasste. Schon in den zwanziger Jahren gab es Firmenvertretungen in Dänemark, Frankreich, Italien, Norwegen, Österreich, Polen, Rumänien, der Schweiz, Spanien sowie in Südamerika. Kontinuierlich wurde an vielfältigen Verbesserungen gearbeitet, so dass man bereits 1933 auf der Internationalen Büromaschinenausstellung (IBA) in Berlin erstmals eine Buchungs-Maschine mit 16 Registern (Speicherwerken) vorstellen konnte. Zum Sortiment gehörten auch Buchungsmaschinen mit eingebauter Schreibmaschine, Kleinaddiermaschinen sowie Vierspezies-Rechenmaschinen.

Der über Jahrzehnte gewachsene Nimbus blieb auch nach dem Zweiten Weltkrieg erhalten, wenngleich die unumgänglichen Reparationszahlungen erhebliche Einschnitte in der Produktion brachten. Aber bereits 1951 konnten wieder Exportverträge in größerem Umfang abgeschlossen werden. 1953 vereinigte man die Produktion der Werke „Astra“ und „Wanderer-Continental“ zum VEB Büromaschinenwerk Chemnitz, dem späteren VEB Buchungsmaschinenwerk Karl-Marx-Stadt (1954). 1955 begann man mit der Entwicklung der Buchungsmaschine Klasse 170, die ob ihrer Leistungsfähigkeit nicht nur zur „Königin der Buchungsautomaten“ (Paris 1964) gekürt, sondern deren Produktion fast unverändert über 28 Jahre aufrechterhalten werden konnte. In der Abschlussbilanz standen 332.742 Maschinen, die, in 102 Länder der Erde exportiert, dafür sorgten, die Chemnitzer Wertarbeit in aller Welt bekannt gemacht zu haben. 1956 musste nach einem langwierigen und angestrengten Prozess, den der Werksgründer Greve von der Bundesrepublik aus angestrengt hatte, das Warenzeichen „Astra“ dem neuen Reservewarenzeichen „Ascota“ weichen, das dann in allen Vertriebsländern eingetragen und ab 1959 für alle Produkte des Buchungsmaschinenwerkes verwendet wurde. Unbeachtet dessen ging die Produktion auf vollen Touren weiter und stellte sich auch den schärferen Wettbewerbsbedingungen des internationalen Marktes.

In dem erfolgreich arbeitenden Betrieb, der zeitweise über 10.000 Beschäftigte hatte, wurde neben den klassischen Buchungsautomaten der verschiedenen Leistungs-Klassen noch eine Vielzahl weiterer Geräte produziert: Kleindatenverarbeitungsanlagen und Zusatzgeräte, Bürocomputer, Diskettenlaufwerke, Schreibmaschinen etc. Deren Marktwert erhöhte sich vor allem im Zuge der Beteiligung am nationalen Datenverarbeitungsprogramm, das in den 1960-er Jahren forciert wurde. Der Anschluss an den internationalen – vor allem westlichen – Markt ging jedoch zunehmend verloren, da der Zugriff auf die Mikroelektronik-Technologie nicht in gewünschtem Maße möglich war und Walter Ulbrichts Parole „Überholen ohne einzuholen“ damit zur Farce verkam. Im Oktober 1991 erfolgte deshalb, nicht zuletzt ob des Verlustes aller Partner des Weltmarktes, die Liquidation der Robotron-Ascota AG – das schmerzhafte Ende einer sächsischen Legende. Der Standort Chemnitz/Karl-Marx-Stadt, der sich in Sachen Rechenmaschinen- und Bürotechnik in dieser Weise fast ein Jahrhundert exponiert hatte, wurde allerdings auch Ausgangspunkt einer modernen Computerentwicklung, nachdem ein Regierungsbeschluss 1957 zur Gründung des VEB Elektronische Rechenanlagen (ELREMA) geführt hatte. Den weiteren Weg zum ESER zu erhellen, möge – soweit noch nicht in Chemnitz und Erfurt behandelt - den anderen Beiträgen dieses Heftes vorbehalten bleiben.

Zur Kooperation zwischen dem NIZEWT (UdSSR) und Robotron (DDR) bei der Entwicklung der ESER- Betriebssysteme - ein historischer und technologischer Rückblick T[1]

H.-Georg Jungnickel

Landbergstrasse 3E
01737 Tharandt
Hgeorg_jungnickel@web.de

Abstract: Nach einem Rückblick auf den Wirtschaftsfaktor „ESER-EDVA“ wird ein außerordentliches Software- Entwicklungsprojekt beschrieben – die zweiseitige Entwicklung aller Hauptspeicher- residenten Betriebssysteme des ESER zwischen NIZEWT und Robotron. Ausgehend von Fakten zur strategischen Rolle der ESER-Systemfamilie in der Informatik des RGW-Wirtschaftsraumes werden die Hauptinhalte der Entwicklungsarbeit bei ESER- Betriebssystemen skizziert, dabei auch eine Übersicht zu den Produkt- Äquivalenten der ESER- und IBM-Betriebssysteme gegeben, sowie Hauptelemente der SW- Entwicklungs-Technologie und die gängige Projekttechnologie bei der Kooperation umrissen.

Nachfolgend erfolgen Kurz-Betrachtungen zu Aspekten des EDVA-Systemüberganges nach 1990 in den neuen Bundesländern und in der UdSSR/Russland unter Nutzung des „System Know-hows“ der Entwickler für „Professional IT- Services“, die von der hohen Kompetenz der ESER- Entwickler zeugen.

1. Einleitende Bemerkungen

- Zum *Autor* des Beitrages :
  - 1981 bis 1990 Direktor des ESER- Entwicklungszentrums Karl-Marx-Stadt (Fachgebiet Geräte „E2“), dessen Vorgänger bereits 1957 als „VEB Elektronische Rechenmaschinen Karl Marx-Stadt“ (ELREMA) gegründet wurde.
  - 1980 bis 1990 „Chefkonstrukteur der DDR im ESER“. Im Rahmen der Arbeit des „Rates der Chefkonstrukteure“ der MRK Rechentechnik auch international für die DDR – Belange dieser Erzeugnislinie verantwortlich.
  - 1990 bis 1991 Vorstand „Entwicklung und Technologie“ der ASCOTA-AG, u.a. weiter verantwortlich für die Geschäftsbereiche des Entwicklungszentrums.
  - 1992 bis 2007 tätig im Projektgeschäft von „Siemens Business Services“ in Osteuropa
- In verschiedenen Vorträgen vorangegangener DDR-Informatik- Symposien wurden Informationen zu den ESER- EDVA der DDR vordergründig im nationalen Umfeld dargestellt. Ohne die Betrachtung und Bewertung der *Beziehungen UdSSR-DDR* sind aber viele innere Aspekte und historische Zusammenhänge der DDR Geschichte „rund um die Rechentechnik“ nicht schlüssig darstellbar.

- Das ESER- System war wohl das erfolgreichste internationale Projekt der wissenschaftlich-technischen Zusammenarbeit im RGW. Es liegt daher nahe, dieses Projekt in seiner Spezifik und Komplexität gesondert, detaillierter und in den vielfältigen Zusammenhängen der Beziehungen zur UdSSR und den dort abgelaufenen Prozessen umfassend zu betrachten (siehe[2]**)**. Im Rahmen des aktuellen Symposiums bietet es sich an, die Geschichte der Schaffung der *ESER-Betriebssysteme* als relativ selbständigen Teil davon zu betrachten.
- Der Vortrag konzentriert sich auf die **ESER- Mainframe-Linie**. Auch die **ESER-PC** (bis EC 1835) begannen ihren erfolgreichen, aber kurzen Weg bei „E2" in Chemnitz. Bei **Softwareentwicklung** bestehen aber hier keine berichtenswerten Besonderheiten, besonders nicht zur UdSSR- Kooperation.

1.1. Zum Wirtschaftfaktor ESER- EDVA

Einleitend soll hier zunächst kurz der enge Zusammenhang der DDR-Arbeiten an der ESER- Mainframe- (EDVA)-Linie und der Arbeiten an den ESER- Betriebssystemen umrissen werden:

- Die Produktlinie ESER- EDVA war wichtigstes Spezialisierungsobjekt der DDR-Rechnerindustrie im RGW und ein Geschäft mit hohen Gewinnmargen. ROBOTRON entwickelte **5 Mainframe- Modelle** in den Architektur-Niveaus ESER- Reihe I bis Reihe III und produzierte **1587 Stck. ESER-EDVA**[4] bzw. –Zentraleinheiten, die zum überwiegenden Teil exportiert wurden (siehe Tabelle Grobübersicht ESER-EDVA der DDR). Das war von der Zahl etwa **10%** des Gesamtaufkommens der MRK- Staaten bei Zentraleinheiten, der Performance- Anteil lag deutlich höher.
- ROBOTRON leistete für die Entwicklung der ESER- Betriebssysteme über 40% der eingesetzten Manpower[5], insgesamt wurden ca. **6.000 Mannjahre** Leistungen erbracht und ca. **6 Mio. "Lines of code"** (ESER- Programmbefehle) incl. Dokumentation erarbeitet. Ihre Finanzierung erfolgte als Umlage auf den Geräte- Preis.
- Zwischen der hohen technischen Qualität und absoluten(!) Systemkompatibilität der DDR- Produkte mit den ESER- Operationsprinzipien einerseits und dem Entwicklungsanteil der DDR an den ESER- Betriebssystemen andererseits bestand eine **totale** wechselseitige Abhängigkeit: ohne Betriebssystementwicklung keine Hardware-Systemkompatibilität und ohne System-Kompatibilität (und höchster Qualität) kein Export -Erfolg. Nur die **Einheit** der Arbeiten an der Hardware der Zentraleinheiten und den Betriebssystemen sicherte der DDR ihre profitable Position und der UdSSR einen unverzichtbaren Entwicklungs-Partner und Lieferanten eines wichtigen Teiles ihres EDVA- Parks.
- alle DDR-Anwendungen von ESER- EDVA auf OS-ES Niveau (1989 ca. 350 Anlagen) wurden mit DDR- Betriebssystemen betrieben. Alle rund 1300 DDR-Exportanlagen wurden an den Anwender mit einem ESER- Betriebssystem ausgeliefert[6], das zwischen UdSSR und DDR auf Vertragsbasis entwickelt wurde.

1.2. Das historische Umfeld

Zunächst ein kurzer historischer Rückblick zum damaligen „strategischen Umfeld":

- 1957 wurde in der DDR zur Konzentration der Kräfte „ELREMA“ gegründet.
- Nach der Kuba-Krise 1962 lebten die Länder des RGW im tiefsten Winter des kalten Krieges und des Handelsembargos. Im RGW ging es um schnellstmögliche Verringerung des Technologie- und Anwendungs-Rückstandes der IT/DV gegenüber den westlichen Staaten.
- In der UdSSR waren mehrere bedeutende Architektur- Schulen und Entwicklungs-Institute in einem höchst uneffektiven, inkompatiblen Parallelismus verfangen. Das Niveau der Anwendungs- Software war besorgniserregend. In der Plankommission, **AdW** und einigen Ministerien der UdSSR wurde intensiv nach **Vereinheitlichung** der DV-Architektur und deren Betriebssoftware gesucht.
- Anfang der 60-er änderte sich in der UdSSR auch das Klima in Wissenschaft und Technik und die allgegenwärtigen Traumata der Folgen des Stalinismus vernarbten langsam. Im mittleren Management der IT- Bereiche kam eine neue Generation Spezialisten zum Einsatz, eine kompetente internationale Zusammenarbeit begann.
- In der DDR wurde nach dem System „Robotron 300“ (analog IBM/1400) verstärkt am Konzept „Robotron 400“ (analog IBM/360) gearbeitet. Einige wenige sowjetische Kollegen kannten diese Arbeiten.
- 1968: Die UdSSR- Regierung entscheidet über die Nutzung von IBM/360 als ESER-Vorbildsystem im Lande. Das „Vorbild“ („Analog“, „Prototyp“) war eine zukunftssichere, mit hoher Autorität behaftete Architektur, die als quasi-Industriestandard (Amdahl, CDC, Siemens, Fujitsu...) außerdem eine Fülle von Anwendungs- Lösungen und Peripherie verfügbar machte[7]. Diese Entscheidung bewährte sich 20 Jahre.
- das Abkommen zur Mehrseitigen Regierungskommission Rechentechnik (MRK) sammelte (Dezember 1968) auch die Kräfte der IT- Branche der europäischen sozialistischen Länder unter einem Dach. Diese Bündelung auf eine Architektur überwand auch im RGW das sich anbahnende Architektur- Chaos.
- Sowohl die Amtszeit Chruschtschows, als auch die DDR-Zentrale unter W. Ulbricht (bis 1971) [8,] gaben einen starken Impulse zum Anstieg der **IT/DV- Investitionen** und verwandter Basistechnologien. Dieser Start vermittelte dem ESER bis Mitte der 80-er Jahre im volkswirtschaftlichen Rahmen ausreichende Impulse für eine gute bis zufriedenstellende Unterstützung bei der Schaffung neuer Technologieplattformen und der breiten Einführung der EDVA. Das erforderliche Technologieniveau war zur damaligen Zeit noch relativ niedrig (SSI, MSI –Schaltkreise, MLLP, Maschinentakt mit HS-Zyklus 1,35 µs), der technologische Abstand zum Prototyp betrug nur wenige Jahre.
- Bereits die ersten Entwicklungs-Ergebnisse unterstrichen die erfolgreiche praktische Umsetzbarkeit der /360-System-Entscheidungen. Die DDR brachte 1973/74 mit EC 1040 das zunächst längere Zeit leistungsfähigste Serien- Modell auf den Markt. In der UdSSR lief die *stabile* Serienproduktion mehrerer Modelle an. Das ESER erlebte in im Verlaufe eines Jahrzehnt- dann schon mit /370- vergleichsweise eine **Blütezeit**.
- Ab ca. 1984/85 verstärkten sich die **Entwicklungs-Schwierigkeiten** in der UdSSR und DDR infolge der Auswirkungen verfehlter Wirtschafts- und Sozialpolitik und der enormen Belastungen aus dem Wettrüsten massiv! In beiden Ländern waren die politische Führung und das wirtschaftliche Umfeld nicht mehr in der Lage, die nötige Unterstützung im Technologie- und Investitionsbereich aller Zweige der Wirtschaft zum Erhalt eines minimalen Tempos zu finanzieren und Kooperationen zu organisieren.
- Trotz der in der zweiten Hälfte der 80-er Jahre zunehmenden politischen Differenzen

zwischen den Reform- Plänen (!) der UdSSR- Führung und dem Konservatismus der DDR-Oberen, und **trotz** der bekannten DDR-Aktivitäten zur „32-Bit-Architektur“, blieb das **wirtschaftliche Gewicht** der ESER- EDVA in den Beziehungen UdSSR-DDR auf hohem Niveau erhalten. DDR- Qualität, nachhaltige „ESER- Systempolitik“ und das massive Beharrungsvermögen des "UdSSR-Hochsee-Liners ESER-EDVA"(ca. 10.000 EDVA im Einsatz) trugen weiterhin Früchte.

Unter den **umrissenen Rahmenbedingungen** realisierten viele leistungsfähige Teams in Forschung, Entwicklung und Produktion **Außerordentliches!**

Wichtigste und zum ESER- Ende 1990 verwertbare Ergebnisse waren
Verfügbarkeit einer Vielzahl von **TOP-Fachleuten** im Mainframe- und PC- Bereich
ein hervorragend vorbereiteter **EDVA -Markt**!

2. Historischer und technologischer Rückblick auf ein außerordentliches Software- Entwicklungsprojekt

2.1. Zur ESER- Architektur und dem Betriebssystemkonzept

CISC- Rechner (**C**omplex **I**nstruction **S**et **C**omputer) waren bis weit in die 80-er Jahre wegen der technischen Realisierungsbedingungen international typisch. Die Befehlsstruktur musste wegen extremer Bit- Kosten *speicheroptimal* sein, eine starke technische Modularität mit Parallelabläufen war für Großrechner zwingend und komplexe Maschinenbefehle benötigten hohen Logik- und Technikaufwand.
Die IBM/360 - Architektur war sowohl im Ankündigungsjahr 1964 als auch zum Start-Zeitpunkt der ESER- Arbeiten 1968 eine höchst innovative und weitreichende „360 Grad“- Universal-Konzeption und für die Belange der geplanten „REIHE“ des ESER ein ideales System-Design. Sie zeichnete sich durch folgende Haupt-Aspekte aus[9]:

- Orientierung auf Geschäftsdatenverarbeitung, Vorbereitung auf Dialog- Datenverarbeitung und Fernverarbeitung.
- Implementierung umfangreicher neuer Möglichkeiten der Systemorganisation, wie max. unabhängige E/A – und Peripherie-Subsysteme
- leistungsfähige Speicherverwaltung unterschiedlichster Speichermedien in mehreren Hierarchie- Ebenen; „Großer“ Adressraum , anfangs begrenzt auf 24 Bit (16 M Byte)
- universelle Logikstruktur für verschiedene Programmarten, Betriebsmodi bzw. Steuerprogramm- Konfigurationen durch ein leistungsfähiges Interruptsystem, Speicherschutz- Mechanismen, geschützte Supervisor- Programm- Verwaltung u.a.,
- 6 Befehlsklassen mit einem sehr breiten Spektrum von leistungsfähigen und flexiblen Befehls- und Datenformaten, Einsatz von 32 Bit- Universalregistern mit 24 Bit Adressfeld.
- erstmalige Einführung des 8-Bit Byte und der 32-Bit Word- Struktur (incl. 32- oder 64-Bit Gleitkommaworte) mit hexadezimaler Basis.

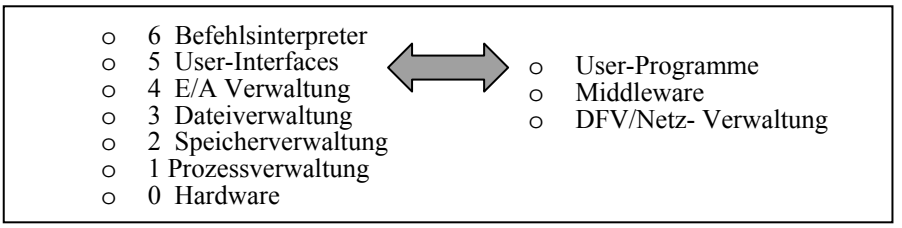
Die echte Auf- und Abwärtskompatibilität der Maschinensprache auf Bit- Niveau,

hinweg über eine Familie von 6 Modellen und einer Performance- Spanne Faktor 50 (!) war eine Sensation. Allein das machte das /360-**Konzept** zum **idealen ESER-Prototyp (Vorbild)**. Seine Entwicklung zum weltweiten Industriestandard gab auch den RGW-Ländern in der EDV-Politik Sicherheit und Kontinuität und sorgte für einen weitgehend einheitlichen Mainframe-Markt incl. Peripherie.

Der **Grundsatz** der gesamten ESER- Arbeiten bestand in einer ***„Prototyp"- nahen Systementwicklung***. Als technische und technologische **Herausforderung** stand im ESER vorrangig die Implementierung des Systemkonzeptes und der Operations-Prinzipien der IBM/360 bzw. /370 Architektur durch einem ***eigenen Logikentwurf*** und deren technischer Realisierung mit RGW- Schaltkreisen/Material gemäß dem UdSSR-Standard GOST 25122-82 „Basiskonstruktionen der technischer Mittel des ESER (TM ESER)" und anderer ESER- Grundsatzdokumente (Bauelementeeinsatz, Prüfbedingungen, Transport- und Lagerbedingungen).

Zu einigen Aspekten der Architektur der Betriebssysteme:

- Wesentliche Schlüsseltechniken heutiger Betriebssysteme wurden durch die IBM / ESER- Betriebssystemkonzepte erstmalig realisiert.
- Die Komplexität der IBM/ESER- Betriebssysteme bildet sich, was deren Architektur betrifft, in einem Schichtenmodell ab. Eine bestimmte Schicht ist jeweils zuständig für die Verwaltung einer Betriebsmittel-Klasse, sie benutzt andere Schichten (Aufrufe) und ordnet sich in eine **funktionale Hierarchie** ein (siehe Abbildung).



- Die universellen Eigenschaften der Logikstruktur - ausgebautes Interruptsystem, leistungsfähige hardware- basierte Schutzmechanismen und Sicherheits-Features, indizierte Adressierung mit virtueller Adressverwaltung zwischen logischer und physischer Adresse- ermöglichten über ca. 20 Jahre die permanente Erweiterung der Betriebsmodi und Programmfunktionalität, Nutzung moderner Speicher, Entwicklung von Fernverarbeitungs- und Dialogsystemen, Mehrprozessorkomplexen u.a..
- Es war wichtig, sowohl die Stapelverarbeitung von Programmen, als auch den Betrieb vieler Nutzer durch „Teilnehmerbetrieb" und Time sharing der Ressourcen effektiv zu gestalten. Umfangreiche Fernverarbeitungssysteme, wie etwa Bildschirmsysteme (EC 7920) mit lokalen und fernaufgestellten Steuereinheiten und vielen (unintelligenten) Terminals und auch alle „Stapel- Tasks" nutzten gemeinsam alle Ressourcen der zentralen EDVA.
- Die Zuverlässigkeit der Datenverarbeitung (Sicherung der Verfügbarkeit und Schutz vor Datenverlust) und die Informationssicherheit (Schutz vor unbefugtem Zugriff oder Angriffen) spielten eine außerordentliche Rolle. Die IBM/ ESER- Architektur und Software war gängigen Kleinrechnersystemen stark überlegen.

2.2. Hauptinhalte der Zusammenarbeit UdSSR-DDR bei ESER-Betriebssystemen [10]

DDR- Softwarespezialisten hatten bereits im Vorfeld der Gründung der MRK gegenüber UdSSR- Fachleuten die Eigenschaften der /360- Architektur überzeugend darstellen können. Ab Gründung der MRK (12/1968) erfolgten alle ESER- Betriebssystem-Arbeiten in der DDR in enger Zusammenarbeit mit der UdSSR, ab ca.1973 dann **arbeitsteilig** auf **kommerzieller Basis** mittels paritätischer Lizenzverträge. Ein einheitliches Betriebssystem war die Voraussetzung der geplanten DDR-Exporte in die UdSSR und für die UdSSR ergab sich die dringende Lösung ihres Manpower- Problems.

Zunächst zum Inhalt:

Unter Federführung zweiseitiger Spezialistengruppen UdSSR/ DDR wurden Konzeptionen (Systemkonzepte) für einzelne *Entwicklungsetappen der Betriebssysteme des ESER- Systems* (ESER I, ESER II, ESER III) erarbeitet, deren Inhalte wurden durch den zuständigen mehrseitigen Spezialistenrat №1 bzw. dessen ständige Arbeitsgruppe OC- EC „Operationssysteme“ (später “Sektion“) des RCK ESER beschlossen und in den internationalen „Entwicklungsplan“ (ESER- Chiffre) aufgenommen. Entwickler der Operationssysteme[11] waren die UdSSR /DDR.

Den Arbeitsinhalt der einzelnen Entwicklungsabschnitte der Betriebssysteme bestimmten:

- die relevanten ESER- Operationsprinzipien (Reihe I bis Reihe III) ,
- die **aktuellen Software- Produkte des Vorbildes**, bezogen auf analoges Funktions- und Technikniveau
- die nationalen Interessen der UdSSR und DDR, abgeleitet von den technischen Parametern ihrer Hauptmodelle und den Forderungen der Haupt-Anwender.

Die nachfolgende Tabelle (Grobübersicht ESER-EDVA der DDR) zeigt aus DDR-Sicht die Komplexität der langjährigen Entwicklung der ESER- Operationsprinzipien und der Dynamik der Betriebssystem- Entwicklung. Relevant dafür war neben den Operationsprinzipien der aktuellen „REIHE“ die Entwicklung der Hauptspeicherkapazität, speziell der DDR- und UdSSR- Modelle, sowie der Plattenspeicher- Technologie und DFV- Komplexe.

Mit den technologischen Entwicklungen ergaben sich Möglichkeiten und Erfordernisse, immer *leistungsfähigere Steuerprogramm-Konfigurationen* zu unterstützen und umfangreiche weitere *Softwarekomponenten* (Datenfernverarbeitung, Teilnehmerbetrieb, Sprachen und Compiler, Dienst- und Serviceprogramme u. a.) bereitzustellen.

Deutlich war diese Entwicklung am Beispiel der Zahl der im *Multiprogrammbetrieb* verwalteten Tasks (**MFT** oder **MVT)** durch Nutzung der virtuellen Adressverwaltung und der Aufgabensteuerung (Job/Task- Control) zu beobachten, sowie später der Zahl der durch eine *Steuerprogrammkonfiguration* verwalteten virtuellen Adressräume, d.h. **SVS** (Single Virtual Storage) oder im ESER sehr viel später die aufwendige **MVS** – (Multiple Virtual Storage)- Verwaltung.

Modell	Befehlszahl	HS-Ausbau	Haupt-Betriebssyst.
EC 1040 1974-1980	**143 Befehle** (ESER I)	1 MByte (256KByte pro Schrank)	DOS EC ; OC- EC/MFT, OC- EC/MVT
EC 1055 1979-1984	**173 Befehle** (ESER II) + 2 Emulationsbefehle+ 37 MaMo- Befehle	2 MByte (1MByte/ Schrank)	OC-6.1. (SVS) /EC
EC 1055M 1983-1986	**182 Befehle** (ESER II-erweitert) (37 MaMo- Befehle)	4 MByte (max.2 Schränke)	OC-6.1.(SVS) /EC SVM /EC
EC 1056 1985-1989	**182 Befehle** (ESER II-erweitert) + interne SVM Mikroprogramm- Makros	4 MByte (max. 1 Schrank)	OC- 7.1./EC mit SVS 7.1./EC,SVM .0./EC BPS 7.1./EC
EC 1057 auch als 2-Prozessor-Modell 1989-1990	**203 Befehle** (ESER III);	16 MByte (mit Coprozessor/ 2 Schränken)	OC -7.2. EC mit SVS- 7.2./ EC, SVM-3.5 /EC, BPS 7.2./EC MVS.2(SP)/EC (ab 1989)

Tabelle 1: Grobübersicht ESER-EDVA der DDR

Eine besondere Bedeutung erlangte in den 80-er Jahren das Konzept der virtuellen Maschine (SVM). Die Spezifik des SVM bestand in der Weiterentwicklung der Virtualisierung einzelner Mittel des Rechners (Hauptspeicher, Peripheriegeräte...) hin zur Virtualisierung einer ganzen EDVA. SVM gestattete die Einrichtung und Verwaltung von mehreren virtuellen Maschinen und die parallele und *völlig unabhängige* Arbeit mehrerer Nutzer auf einer Anlage. Der autorisierte Nutzer musste dafür jeweils ein ESER-Betriebssystem oder ein system-unabhängiges Dienstprogramm nutzen. Insbesondere im Entwicklungsbetrieb durch mehrere Test- und Diagnosesysteme oder bei der Arbeit mit stark vertraulichen Inhalten verschiedener Nutzer ergab sich eine *hohe Informations-Sicherheit*. Ab dem OC 7.1./EC wurden in der UdSSR- DDR-Kooperation mehrere SVM- bezogene Teilsysteme bearbeitet, die Systeme **BPS** 7.1./EC.und **SVM** 3. /EC im SVS 7.1./EC und nachfolgend SVS 7.2/EC. Obwohl SVS und BPS vorrangig für den Stapelbetrieb konzipiert waren, arbeitete das sowjetische „Basisbetriebssystem" BPS *ausschließlich* unter einer virtuellen Maschine.

Beginnend mit OC 7 /EC (1983) war eine gewisse *Trennung der Produktpolitik* zwischen der UdSSR- Linie des OC 7 und in den DDR-Kunden-Versionen nicht vermeidbar[12]. Die UdSSR- Partner erarbeiteten eine **Originalstruktur** des OC 7.1. /EC ohne ausländisches Vorbild. Auf der UdSSR- Seite begann eine *teilweise* Abwendung von IBM- Vorbild- Lösungen. Die in der DDR zur Auslieferung gekommenen Systeme OC 7.1/ EC und nachfolgend OC 7.2./ EC sind in den entsprechenden Artikeln der DDR-Entwickler genau beschrieben.[13] Spezifische Eigenschaften, die das SVM bietet (hochgradige Nutzertrennung, Informationsschutz u.a. für wichtige Großanwender in der UdSSR) waren dominant dafür, dass das OC 7.2./EC spezifisch strukturiert wurde. Bei der breiten Masse der DDR- Anwender blieben die IBM- nahen Konzepten weiterhin favorisiert und Robotron machte Kundenberatung für OC 7.2./EC mit SVM.

Die zunehmende *SVM-Dominanz* in der Systempolitik des NIZEWT und im Hauptexportland UdSSR war auch der Hauptgrund der Ausstattung der EC 1056 mit einem leistungsfähigen mikroprogrammresidenten Satz von *SVM- Makros*, die die SVM-Performance bis zu 50% erhöhen konnten.

2.3. Aufgabenstellung und Technologie der ESER- Kooperation und des Projektmanagement

Die ESER- Betriebssystem- Arbeiten in der DDR nach Gründung der MRK (12/1968) erfolgten, wie bereits ausgeführt, **arbeitsteilig** mit der UdSSR auf **kommerzieller Basis** mittels zweiseitiger Lizenzverträge. Das jährliche vertragliche Volumen betrug im Höhepunkt der Arbeiten ca. 600 Mannjahre, jede Seite erbrachte strikt 50% der Leistung und übergab diese dem Partner als Lizenz. Die Finanzierung war also ein *0- Summen-Clearing,* beide Seiten wurden Eigentümer des Gesamtsystems.

Die Methodik der inhaltlichen Definition der Entwicklungsabschnitte basierte auf einer ausführlichen Analyse entsprechender Unterlagen zur Strategie der IBM und anderen IBM- Publikationen. Da die ESER- Entwicklung mit ca. 4 Jahren Versatz zum Vorbildsystem startete und im Verlaufe der Arbeiten etwa 8 Jahre Abstand hinnehmen musste, bestand die Erarbeitung der *„Aufgabenstellung"* für die nächste Etappe des ESER- Betriebssystems vorrangig in der sorgfältigen Analyse der Produktpolitik und von Publikationen der „Prototyp- Firma". Es wurden auch deren ***Original- Produkte*** analysiert bzw. ausgewertet, denn sowohl die UdSSR, als auch die DDR konnten für bestimmte Unternehmen (z.B. für die Zentralverwaltung für Statistik oder das Kombinat Datenverarbeitung) auch gemäß Exportbestimmungen der USA- Regierung ***Original-EDVA*** importieren[14]. Die Auswertung der mitgelieferten Unterlagen war dann auch unseren ESER- Spezialisten möglich. Auch aktuelle Maschinenzeit für Testarbeiten für ESER- Betriebssysteme stand so zur Verfügung.

Ziel der Arbeiten war die Entwicklung ***vertriebsfähiger ESER- Betriebssysteme*** gemäß dem damals international gültigen Stand des Software- ***Urheberrechtsschutzes***. Die Prototyp- nahe Systementwicklung erforderte keine umfangreichen *eigenen konzeptionellen und systemtechnischen* Vorlaufarbeiten in der mehrseitigen oder zweiseitigen Kooperation, wohl aber gute Systemkenntnisse und umfangreiche „handwerkliche Arbeit". Schließlich mussten die Anwender sich darauf verlassen können, dass sie *kurzfristig* zu jeglichen Funktionsabweichungen eine Lösung erhielten (batches, Umgehungslösungen) sowie längerfristig das Problem grundsätzlich behoben wurde (Ausgaben, Versionen).

Am Start der Kooperation wurden die Prinzipien der ***Entwicklungstechnologie*** definiert und später laufend ergänzt. Folgende Haupt-Elemente waren relevant:

- Erstellung eines Ausgangssystems ***)**
- Erarbeitung des Quellcodes des Ausgangssystems (für vertiefte Analysen)
- Definition aller Moduln und ihrer Schnittstellen (Makronamen, Aufrufe, andere Software- Schlüssel-Elemente)
- Grundsätze der Anpassung der Moduln an die ESER- Forderungen: Analyse und Änderung von Befehlsfolgen nach fixierten Kriterien, Anpassung der Programmlogik bei Erhalt der Makros, Calls usw., ESER- Kommentare, Anpassung an das E/A-Gerätespektrum, Erarbeitung von Dokumentation u. a. Makros, Aufrufe und andere Schnittstellen behielten Original-Namen;
- ***)** Das Vorbildmaterial waren kompilierte, d.h. kundenkonkrete ablauffähige Programmsysteme. Zur Gewinnung des Quellcode der Ausgangssysteme war jeweils eine programmunterstützte Re-Assemblierung erforderlich, die durch dialogorientierte Bearbeitung des Prototyp- Produktes erfolgte. Das war eine Schlüsseltechnologie, ohne deren Existenz am Start der Zusammenarbeit das ESER wahrscheinlich so nicht bestehen würde [15]

- Methodik der Zusammenstellung von Zwischenständen und Integrationstests
- Fehlerdienst und Nachbetreuung
- Methodik der Fehlerbehandlung und des Entwicklungsabschlusses des Vertragsteil**)
- die Entwicklung eines vertriebsfähigen ESER- Produktes implizierte die Sicherheit des Lieferanten, dem Anwender eine vertretbare Zusicherung geben zu können, dass durch „Fremdcode“ oder die Existenz von Registriernummern oder Kennzeichnungen (etwa Vorstufen heutiger Trojaner oder Systemnummern) kein Informationsabfluss vorbereitet wird oder andere Sicherheitsaspekte berührt werden.
- ****)** Das ESER- Betriebssystem musste zum Original kompatibel sein und wurde daher zunächst auf Prototyp- EDVA getestet und korrigiert. Funktionsabweichungen derart getesteter Systeme auf einer ESER- EDVA zeigten mit höchster Wahrscheinlichkeit dann auf Logikfehler des Entwurfs der ESER-ZE.

Zu Beginn einer neuen ***Entwicklungsetappe*** erfolgte die genaue Auswahl eines Prototyp- Produktes, dessen Einzelkomponenten (d.h. einzelne Funktionsbereiche wurden ggfls. zeitlich vertagt) und die Definition des Rahmenzeitplanes der Entwicklung bestimmter Teilsysteme, sowie der Arbeitsteilung. Es wurde sowohl arbeitsteilig nach Funktionsmoduln gearbeitet, aber später auch Versions- bezogen. Daher bestand keine durchgängige Spezialisierung der Seiten auf bestimmte typische Moduln, jedoch mehr Flexibilität in der operativen Arbeit.

UdSSR- und DDR- Beauftragte und die Softwareteams beider Seiten erarbeiteten zur Vorbereitung des nachfolgenden ***Vertragsabschnittes*** detaillierte Aufgabenstellungen mit Funktions-Beschreibung einzelner Moduln, Terminen, Schnittstellen und Zwischenetappen. In späteren Jahren wurden so auch vom Prototyp ***abweichende*** Systemkonzeptionen (z.B. BPS der UdSSR) erarbeitet und vertraglich vereinbart.

2.4. Zum Projektmanagement und Arbeitsklima

Die Entwicklungsarbeiten wurden, gemäß heutiger Terminologie, durch ein paritätisch besetztes ***Projektboard*** („Entwicklungsleiter“) mit 3-4 Meetings im Jahr geführt. Für einzelne Themengebiete bestanden Teilprojekte mit Teilprojektleitungen, deren Verantwortliche i.d.R. jeweils die Leiter ganzer Bereiche, Abteilungen oder Gruppen waren. Diese verantworteten die Entwicklungsarbeiten ihrer Seite und berichteten im Vertragsrahmen an das Projektboard. Vertrags- Abschlüsse oder Grundsatzfragen zum Vertrag wurden auf Ebene der Direktoren der Betriebe (NIZEWT/ Robotron-E2) geregelt. Differenzen auf Ebene der Direktoren bestanden sehr selten.

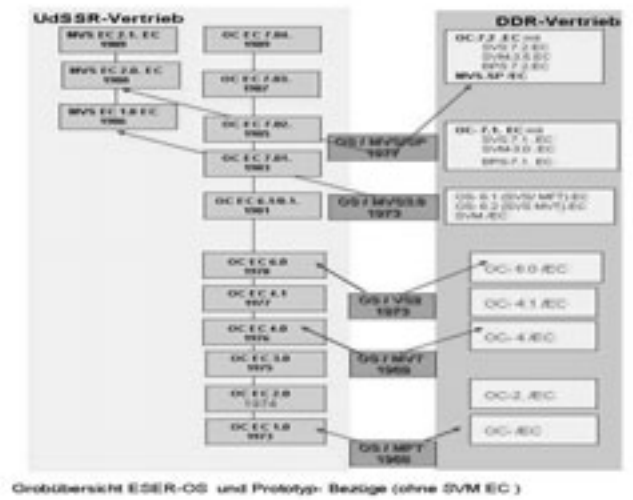
Während der jahrelangen fruchtbaren Kooperation entstanden viele persönliche Kontakte. Kameradschaftliches Miteinander und gegenseitige Unterstützung waren die Regel. Es gab nicht wenige persönliche Freundschaften zwischen den Mitarbeitern. Trotz bestimmter Regelungen der Sicherheitsdoktrin der UdSSR entstanden zu keiner Zeit Spannungen, „persönlich“ und „dienstlich“ waren zwei paar Stiefel. Die übergroße Mehrzahl der DDR- Mitarbeiter war froh und motiviert, in einem derartigen Team zu arbeiten.

*Ein Detail**:*** Übergaben der Zwischenstände/ Endergebnisse erfolgten auf ½ Zoll Magnetband–Cartridge, später auch auf 14“ Plattenstapeln. Das war in der grenzüberschreitenden Kooperation der einzig praktikable Weg für die umfangreichen Softwarekomponenten und Systeme, die oftmals mehrere MByte groß waren. Anfangs wurden aus Vertragsgründen zusätzlich noch Papierausdrucke übergeben. Dafür mussten ganze Zugabteile mit Papier beladen werden. Der

technische Aufwand derartiger Übergaben, aber auch die Einschränkungen in der Kommunikation mit dem Partner sind heute nicht mehr vorstellbar. Auch nicht die KGB Restriktionen. Anfang der 80-er Jahre wurden Sonder- Genehmigungen für online- Verbindungen erteilt. Bei der Kooperation Robotron/ NIZEWT wurden erstmals im UdSSR- DDR- Datenverkehr 1,2 Kbit/s Wahlleitungen/Modems eingesetzt, die teilweise nächtelang in Betrieb waren. Eine spürbare Erleichterung vor allem der operativen Kommunikation trat ein.

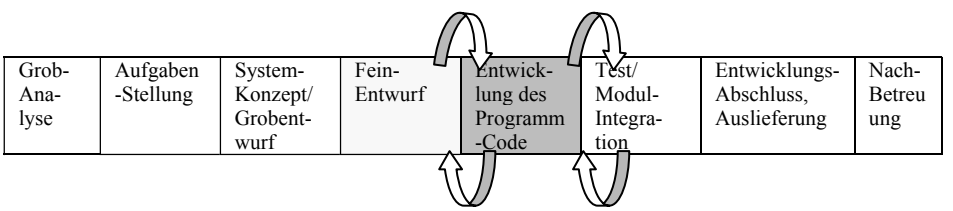
3. ESER- und IBM Produkt-Äquivalente der Hauptbetriebssysteme

Dieser Aspekt kann im Rahmen dieses Vortrages wohl am besten durch die Graphik „Grobübersicht ESER-OS und Prototyp- Bezüge“ umrissen werden.



Grobübersicht ESER-OS und Prototyp- Bezüge (ohne SVM EC)

Details im Bereich SVM siehe auch [16] .
Zur Frage, in welcher Weise seitens der ESER- Software-Entwickler ***„nach“- gedacht*** wurde, sind oben bereits Ausführungen erfolgt. Ergänzend dazu:
In einem heute typischen Software- Projekt- Ablauf sind i.d.R. die folgenden Stufen zu durchlaufen:



Gemäß der fixierten ESER-Entwicklungstechnologie wurden alle Programmier- und Dokumentationsarbeiten etwa ab der Stufe „Feinentwurf“ in hohem Maße **eigenständig** erbracht, mit definiertem eigenem Code-Anteil, eigenen ESER- EDVA und eigener technologischer Software. Abweichungen von dieser Entwicklungs-

Technologie erfolgten *ab ca. 1987* bei den MVS- Arbeiten mit / 370 Architektur.

- Das erfolgreiche *Projektmanagement* nutzte Managementsysteme (Projektplanung u.a.), die heute üblichen Verfahren sehr ähnlich waren. Infolge der heute primitiv zu nennenden Kommunikations- und technischen Bedingungen (z.B. kaum Telefonate, „Textverarbeitung" mit EDVA u.a.) des Zusammenwirkens über Grenzen war die Leitung (V.: Leonid Raikow/ Walter Münch) eine anspruchsvolle Leistung. Der Abstimmungs- und Integrationsaufwand zwischen den Entwicklungspartnern war nur durch strikte Disziplin, Einhaltung der technologischen Vorgaben und hohe Kompetenz beherrschbar.
- Der „Programm-Start" ab dem „Feinentwurf" erforderte hohen **Aufwand** zur **Einarbeitung** in die Logikwelt des Prototyp-Produktes plus Aufwand zur Anpassung an eigene Gerätekomplexe und deren Unterstützung und Diagnose, wie E/A- Geräte, Displaysysteme, Fernverarbeitungsgeräte, Mehrrechner-Kopplungen u.a., die eine ***vollständige inhaltliche*** Beherrschung des Produktes erforderten, vor allem auch im Fehlerfalle (!) beim Anwender.

Zusammenfassend wäre festzustellen, dass die Prototyp-Orientierung überhaupt erst die praktizierte Form der ESER- Arbeitsteilung ermöglichte. Sie *erhöhte die Sicherheit von Systementscheidungen und logischen Teillösungen außerordentlich* und senkte den Konzeptions- und Entwicklungsaufwand stark. Richtig ist aber auch, dass der für eine wirtschaftlich und technologisch stabile Entwicklung der Produktlinie e*rforderliche und auch erreichte* Grad der ***unbedingten logischen Beherrschung*** aller Programm-Elemente (incl. Bearbeitung von Funktionsabweichungen, Fehlern und Umgehungslösungen) einen hohen personellen Aufwand erforderten und durch ein großes Team von ESER- Entwicklern bei E2 zweifellos gut beherrscht wurde.

4. Systemübergänge 1990 in Ostdeutschland und Russland/GUS

Die politischen und damit einhergehenden wirtschaftlichen „Paradigmenwechsel" des Gesellschaftssystems vollzogen sich in Ostdeutschland und in der ehem. Sowjetunion etwa zur gleichen Zeit, aber wie bekannt unter *extrem unterschiedlichen* Rahmen-Bedingungen. In beiden Ländern waren 1990 die ESER- EDVA die dominierende Basis der Informations- und Datenverarbeitung in Behörden und Verwaltung, Finanzwirtschaft, Verkehr und allen anderen lebenswichtigen Bereichen des Landes. In der DDR waren 1990 ca. 350 EDVA Anlagen im Einsatz. In der UdSSR waren im Jahre 1990 noch ca. 10.000 ESER-EDVA im Einsatz.

Die mit dem Wechsel verbundene Erneuerung der Informations- und Datenverarbeitung war in Ostdeutschland dominiert von einer Eingliederung in das bestehende Wirtschafts-Verwaltungs- und Rechtssystem der BRD in extrem kurzen Übergangsphasen. In der UdSSR erfolgte der „Wechsel" bekanntermaßen in einem mehrjährigen Prozess bei offenen Landes-Grenzen und freier Valutawirtschaft und einem widersprüchlichen nationalen Kräfteringen.

4.1. Wenige Aspekte des DV- Systemüberganges in den neuen Bundesländern

Neuinvestment oder Austausch von ESER-EDVA war dank der IBM- Kompatibilität des ESER beinahe ein Selbstläufer. IBM oder Siemens (Siemens trotz nicht kompatibler Plattensektoren) „erhielten" in den Regionen Ostdeutschlands jeweils ihren Marktanteil für Ersatzinvestitionen im Regierungs- und Verwaltungssektor, sowie bei Banken. Jeder Leiter brachte „sein System" mit. Der Wandel bei privaten Unternehmen bestimmte sich vielfach durch wirtschaftlichere Client-Server- Systeme mit INTEL- bzw. IBM/PC-Architektur. Diese neuen ***Mainframe- und PC- Architekturen*** waren den zahlreichen guten IT/DV-Fachleuten der neuen Länder aus ihrer DDR-Tätigkeit bestens bekannt,

Die beiden bereits genannten **Key- Player** des deutschen IT/DV – Marktes engagierten sich in den neuen Ländern in deutlich unterschiedlicher Weise. *Siemens* praktizierte die Übernahme von Kennern des Ostmarktes und baute damit *Vertriebsstrukturen* auf. Und es erfolgte eine Verlagerung von IT-Auslaufprodukten nach Ostdeutschland, eine Art Arbeitsbeschaffung und Vorstufe des aktuellen Produktions- Outsourcings. *IBM Deutschland* orientierte neben dem zielstrebigen Aufbau eines Vertriebsnetzes (ausschließlich für die BRD!) bereits auf den Aufbau ihrer „Professional IT-Services" und verfolgte zielstrebig die Einbindung der besten *Robotron- Systementwickler* in diese Strukturen. Durch Gründung der „IT Solutions and Services GmbH" (zunächst als Joint Venture, später als 100%-ge Tochter, heute „ITSAS") fanden 1990 zunächst 220 der erfahrensten ESER- Entwickler in Chemnitz eine Perspektive. Weitere IBM-Aktivitäten erfolgten bei Robotron Vertrieb Berlin und Leipzig. Heute hat die ITSAS-GmbH[17] 10 Standorte in der BRD und ca. 1200 Mitarbeiter. Im „Global Services Verbund" der IBM leisten die relativ wenigen noch aktiven ehemaligen Mitarbeiter hochwertige IT-Beratung und –Umsetzung. Die konsequente Orientierung des Teams von E2 auf die ESER-Architektur erfährt so in Chemnitz eine nachhaltige erfolgreiche Weiterführung.

Auch vielen **anderen Firmen in Sachsen, Thüringen und Brandenburg** half das gute **ESER- Know –How** ihrer Gründer und Mitarbeiter in der Startphase.

4.2. Kurz zum Systemübergang nach 1990 in Russland

Mit dem Zerfall der UdSSR mussten ESER- Produktion und -Services 1990/91 eingestellt werden. Wartung und Modernisierung der Maschinen (neue Plattensysteme!) wurden extrem schwierig. Bis **1999** blieben dennoch ca. 5000 ESER- EDVA vorrangig deshalb im Einsatz, weil deren Anwender-Programmsysteme lebenswichtig waren.

In den neugegründeten „MBO"- Firmen „Restart" und „EC-Leasing" realisierten NIZEWT- Mitarbeiter „Professional Services", die äußerst schnelle Migration der Anwender- Software der ESER- Nutzer auf modernere IBM- Plattformen , ohne großen Aufwand, kompatibel ist kompatibel! Ein größerer Teil der Mitarbeiter des NIZEWT ist auch heute noch mit „IT Solutions und Services" für die große Zahl der ehemaligen ESER-Anwender im Staats- und Regierungsapparat und in Staatsbetrieben tätig, die bis heute als IBM-Anwender (!) vertrauliche Anwendungssysteme, Daten-Netze und Hochleistungsrechner-Komplexe nutzen. Diese wurden teilweise selbst entwickelt und werden mit höchster Kompetenz und Vertraulichkeit im IT-Service betreut, wie auch Rechnerkomplexe und Spezialsysteme -Derivate aus den ESER- Betriebssystemen.

5. Résumé

- Das ESER war integriert in das Feld der Systemkonfrontation zwischen Ost und West.
- Unter den **umrissenen Rahmenbedingungen** der ESER- Jahre realisierten viele leistungsfähige Teams in Forschung und Entwicklung in der DDR und UdSSR **Außerordentliches!** Das ESER erlebte ca. ab 1975 bis 1985 eine **Blütezeit.**
- Es ist uns nicht bekannt, dass die Firma IBM nach 1990 juristische Schritte zu Schutzrechtsfragen eingeleitet hätte, weder in Russland, noch in Ostdeutschland. Es gab dazu offenbar weder Interesse, noch existierte eine Rechts- Basis.
- Das „System-Wissen“ und die Arbeitsmethoden der ESER- Entwickler und Service-Spezialisten waren 1990 durchaus vergleichbar mit Teams aus der BRD.
- Es ist höchst erfreulich, dass auf der Basis von exzellentem Wissen, hervorragendem Können aus der Praxis und hoher Motivation viele mit dem ESER aufgewachsene ehemalige Kollegen noch heute einen stabilen Arbeitsplatz ausfüllen, viele davon ***als selbständige Unternehmer oder leitende Mitarbeiter*** größerer Firmen. Das gibt mir auch eine nachträgliche Bestätigung der umfangreichen Aktivitäten und des hohen persönlichen Einsatzes zum Erhalt der ESER- Mainframe- Linie bis 1990 in Chemnitz.
- ghChemnitzChemnitz.

Anmerkungen und Verweise

[1] „**NIZEWT**“: Leitinstitut des ESER und Sitz des Generalkonstrukteurs des ESER ,
„VEB Kombinat **Robotron**“ war das verantwortliche Kombinat der DDR für Entwicklung, Produktion und Kundenservices von Mitteln der Rechentechnik/ Datenverarbeitung

[2] Gesamtkomplex siehe z.B. http://www.eser-ddr.de/Rechentechnik der DDR im ESER.htm

[4] EDVA- Elektronische Datenverarbeitungsanlage = nutzungsfähige Gesamtkonfiguration incl. Peripherie und Betriebssystem und ggfls. Basissoftware

[5] DOS , Netzbetriebssysteme u.a. anderer Entwickler wurden eingerechnet;

[6] Komplettierung in UdSSR durch den Systemdienst VO SojusEVM Komplex

[7] Entscheidung: Kommission für RT der ADW der UdSSR am 27.01.1968 unter AM A.A. Dorodnizyn

[7] PB- und MR – Beschluss vom 23.06.1964 / 03.07.64: "Programm von Maßnahmen zur Entwicklung, Einführung und Durchsetzung der maschinellen Datenverarbeitung in der DDR in den Jahren 1964 bis 1970“

[9] IBM JOURNAL OF RESEARCH AND DEVELOPMENT VOL. 8 NO. 2 1964
http://www.research.ibm.com/journal/rd/441/amdahl.pdf

[10] Ausführlich: http://www.eser-ddr.de/BetriebssystemeEDVAdesESER_2.htm

[11] In einer späteren Phase kamen die CSSR und die UVR als DOS3- Entwickler hinzu.

[12] Details: http://www.eser-ddr.de/BetriebssystemedesESER-UdSSR-Ueberblick_000.htm#_Toc163133835

[13] Übersichtsartikel dazu siehe http://www.eser-ddr.de/Artikelsammlungundsitemap.htm

[14] In der DDR erfolgten Import nach Genehmigung durch ein „Bilanzorgan“ – für IT/DV das MEE

[15] Siehe http://www.eser-ddr.de/BetriebssystemedesESER-UdSSR-Ueberblick_000.htm

[16] siehe hierzu [(14)]

[17] http://www.itsas.de/

DEKK – das DDR-Konzept zum Zusammenführen von Büromaschine und Computer und dessen Umsetzung

Siegfried Junge

Lugturmstraße 34
01809 Heidenau
siegfried.junge@web.de

Abstract: Als technische Grundlage für die Zusammenführung der Kapazitäten auf dem Gebiet der Dezentralen Datentechnik seitens der Kombinate Robotron und Zentronik wurde in Arbeitsgruppentätigkeit eine Konzeption unter der Kurzbezeichnung DEKK erarbeitet. Der Beitrag zeigt die dabei entworfenen und die einbezogenen Komponenten sowie die daraus abgeleiteten Finalerzeugnisse auf und bewertet die Nutzung der Ergebnisse in den 1980er Jahren.

1 Einführung

Die Entwicklung auf dem Gebiet der Rechentechnik/Datenverarbeitung Ende der 70er Jahre war durch die explosive Entwicklung der Mikroprozessortechnik in der Welt geprägt. Der Beitrag der DDR-Bauelementeindustrie in Form der zu Intel 8008 und Z 80 kompatiblen Mikroprozessoren U808 und U880 hatte demzufolge starke Auswirkungen auf das Erzeugnisprogramm von Robotron.

Das bewirkte aber auch eine immer stärkere Parallelität der Bereitstellung rechentechnischer Erzeugnisse der Kombinate Robotron und Zentronik. Dies war eine natürliche Folgeerscheinung, denn die mechanische Büromaschinentechnik, einst das Markenzeichen von Zentronik, wurde immer stärker zurückgedrängt. Das Kombinat Zentronik musste sich also stärker zur elektronischen Verarbeitung hinwenden. In einer kapitalistischen Umgebung hätte man das den Markt klären lassen, bei zentralistischem Herangehen konnte man das nicht zulassen und so musste eine geregelte Sortimentsbereinigung erfolgen.

Wenngleich auch bereits bei der Gründung von Robotron auf eine Zusammenführung orientiert wurde, schlug sich das nur sehr zögernd in praktische Schritte um. So war im Jahre 1977 eine Eingliederung des Betriebes in Zella-Mehlis erfolgt, da dort seit Jahren für Robotron Kleinrechentechnik und in der letzten Zeit auch das bei Robotron entwickelte Mikrorechnerbaugruppen-System K1510 produziert wurde.

Im Weltmaßstab war weiterhin charakteristisch, dass die Gerätelinien der Bürotechnik und der Rechentechnik stärker zusammenflossen, was letztendlich auch mit der wachsenden Vernetzung der Geräte am Arbeitsplatz zusammenhing. Erfassungsgeräte waren dann plötzlich Terminals für zentrale Rechner und Fakturiermaschinen migrierten zu Bürocomputern.

In Auswertung dieser Tendenzen startete das zugehörige Fachministerium bei Robotron die Arbeit einer Arbeitsgruppe mit dem Ziel, die wesentlichen Gerätelinien zu definieren, die in beiden Kombinaten auf einer gemeinsamen technologischen Basis das Fundament für eine effektive Produktion sein könnten. Das betraf ein Spektrum von Erzeugnissen, begonnen bei den elektronischen Bürogeräten, über die notwendige rechentechnische Basis und Konstruktionstechnologie bis hin zu den Terminals für Großrechner .Natürlich leiteten sich hier auch Forderungen an periphere Geräte und Baugruppen ab. Das gesamte Ergebnis wurde unter die Überschrift **„Konzeption der Geräte für Datenerfassung, Kleinrechentechnik und Kommunikation (DEKK)“** gestellt und war die fachliche Grundlage für die dann folgende Eingliederung des Kombinates Zentronik in das Kombinat Robotron. In der Werbung trat dieses Konzept dann später als „Programm der Dezentralen Datentechnik“ in Erscheinung

Diese Strukturierung gab dann auch wichtige Impulse für den weiteren Ausbau solcher Produktionsstätten wie den Betrieb Auerbach, der ganz speziell auf Tastaturen ausgerichtet war.

Es wurde aber auch schnell sichtbar, dass bestimmte Peripheriegeräte, wie z.B. Diskettenspeicher, nicht in dem erforderlichen Umfange über die gängigen Wege der Spezialisierung im RGW erhältlich waren, was letztlich zur neuen Aufnahme solcher Erzeugnislinien, im konkreten Falle für externe Speicher in Zella-Mehlis führte.

2 Die Zielrichtung

Für die Arbeitsgruppentätigkeit zur Konzeption DEKK, die im Jahre 1976 begann, galt diese Zielstellung:

- Die herausgearbeiteten Erzeugnisse müssen ein Maximum an Anforderungen der DDR-Anwender befriedigen können. Insbesondere muss den Forderungen nach dezentraler Verarbeitung, Ablösung der Rechnerfamilie R 4000, Technik zur Produktionsvorbereitung (später als CAD/CAM propagiert) Rechnung getragen werden. Gleichermaßen muss die Grundlage für den Export ins Ausland gegeben sein.

- Die Geräte und Module, die bei Robotron und Zentronik bereits in Produktion und Entwicklung sind, müssen maximal weiter genutzt werden können. Das gilt insbesondere für das Mikrorechnermodulsystem K 1520 und das Mikrorechnersystem K 1600.

3 Die geräteseitigen Basisbestandteile des DEKK

Die als Finalerzeugnisse herausgearbeiteten Lösungen wie z.B. die verschiedenen Bürocomputer, werden im Punkt 4 näher betrachtet. Nachfolgend wird auf die genutzten gerätetechnischen Komponenten eingegangen.

3.1 Die rechentechnische Basis

Das Mikrorechnerbaugruppensystem K 1520

Da bereits mit dem Baugruppenkonzept K 1510 gute Erfahrungen vorlagen, war mit der Ankündigung des zum Z 80 kompatiblen Mikroprozessors U 880 die Vorbereitung des Nachfolge-Baugruppensystems K 1520 in vollem Gange, als mit dem Konzept DEKK begonnen wurde. Dieses Baugruppensystem stellte fortan eine tragende Säule dar.

Das System K 1520 [Un78] bestand aus Steckeinheiten im Doppel-Europaformat mit indirektem Steckverbinder, die in Einheitskassetten mit 5, 7 oder 11 Steckplätzen gesteckt werden konnten. Diese Basiskonstruktion war DDR-weit standardisiert worden. Als einheitlicher Nenner wurde ein Systembus geboten. Aus den Zentralen Verarbeitungseinheiten (ZRE), Speichersteckeinheiten, Anschlusssteuerungen für vielfältige Peripheriebaugruppen und –geräte, Adapter für standardisierte Interfaces sowie einem Sortiment an vereinheitlichten Stromversorgungen ließen sich mannigfaltige Mikrorechner konfigurieren.- Das System K 1520 stellte ein offenes System dar, das laufend erweitert wurde.

Es existierten im Wesentlichen 3 Typen von ZRE. Auf die damit verbundene Problematik der unterschiedlichen Struktur der Rechner soll bei den Finalerzeugnissen eingegangen werden.

Das Mikrorechnersystem K 1600

Auf der anderen Seite entstand bereits das Mikrorechnersystem K 1600, das von der Struktur her ein Minicomputersystem war, aber aus Gründen der staatlichen Förderung auf der Mikroelektronikwelle mit schwimmen musste.

Als im Jahre 1974 im RGW-Bereich das „System der Kleinrechner“ gegründet wurde, gab es als wesentliche Orientierung, die Rechner der Reihe PDP 11 von DEC mit eigenen Bauelementen funktionell nachzubilden. Robotron stellte damals Rechner der Familie 4000 her, die sich an einem Honeywell-Rechner orientierten. Deshalb gab es bis zur Ablösung zu Beginn der 1980er Jahre keinen Robotron-Kleinrechner im System der Kleinrechner (SKR). Als nun die Frage der Ablösung stand, kam nur ein Modell mit PDP 11-Orientierung in Frage. Das bedeutete eine vollkommen neue konstruktive Orientierung, die damit auch in das Erzeugnisprogramm DEKK getragen wurde.

Das System K 1600 kann grob wie folgt umrissen werden [Gi80]:

- Rechnersystem in Einschubbauweise, untergebracht in 19“-Schränken
- Leistungsabstufung durch die 2 Modelle K 1620 ohne und K 1630 mit Arithmetikprozessor
- Realisierung der Hauptfunktionen über das Sortiment hoch integrierter Schaltkreise U 830 und der Befehlsliste mittels Mikroprogrammierung
- Verarbeitung von 16-Bit-Worten oder Bytes, Abarbeitungszeit bei Festkommaaddition 8,5 µs und bei Multplikation 23 µs beim Rechner K 1630
- Kopplung der Anschlusssteuerungen und Geräte über einen K 1600-Bus oder über den SKR-Einheitsbus (letzterer für den Anschluss von direkt PDP 11-kompatiblen Geräten)
- Abarbeitbarkeit von PDP 11-kompatiblen Programmen

3.2 Die Basiskonstruktion (Grundgestaltung)

Die neue Erzeugnisgeneration, die bei den arbeitsplatzorientierten Geräten besonders auf die Bildschirmtechnik orientiert war, erforderte deshalb eine komplette neue Design-Linie bei Robotron. Bei den kleinrechnerorientierten Erzeugnissen war die Gestaltung durch die Forderung der international eingeführten 19“-Schranktechnik geprägt.

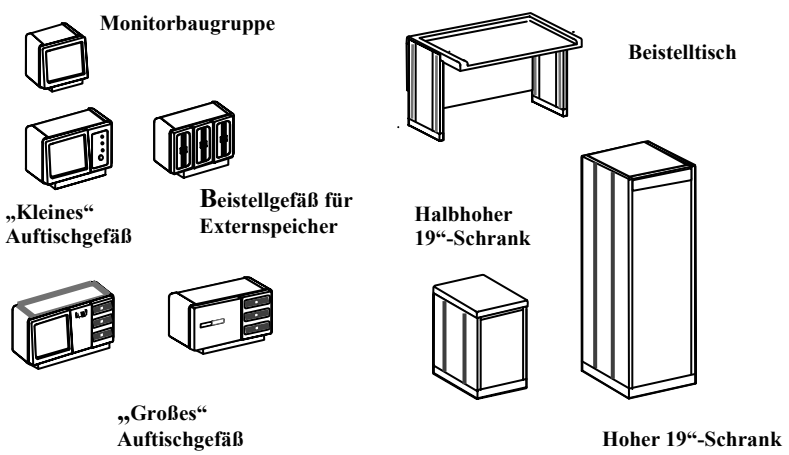


Abbildung 1: Module der DEKK Basiskonstruktion

Im Rahmen einer von allen Kombinatsbetrieben getragenen Arbeitsgruppentätigkeit entstand unter der Leitung des für die Basiskonstruktion zuständigen Fachgebietes E 2 des Zentrum für Forschung und Technik (ZFT) bei Mitwirkung der Gestalter von Burg Giebichenstein in einer kurzen Zeitspanne ein durchgängiges Projekt. Die wesentlichen Merkmale wurden für alle Beteiligten verbindlich und die konstruktive Lösung und Produktion der einzelnen Elemente erfolgte in verschiedenen Kombinatsbetrieben.

Mit diesem in Abbildung 1 zusammengefassten Modulsortiment kam dann auch die gesamte K 1520- und K 1600-Technik bis Ende der 1980er Jahre zur Auslieferung.

Die Umsetzung der Basiskonstruktion des DEKK bedeutete gewaltige Anstrengungen in der Phase des Überganges.. Bei den Auftisch- und Standgeräten war insbesondere den Anforderungen eines modernen Designs zu entsprechen, denn diese Technik sollte letztendlich auch die Kunden im westlichen Ausland erreichen. Andererseits mussten auch die materiellen Gegebenheiten der DDR berücksichtigt werden. So entstanden z.B. die typischen Verkleidungen auf Polyurethanbasis, die sicherlich gewöhnungsbedürftig waren aber dennoch Anklang fanden.

Bei den kleinrechnerorientierten Lösungen traten generell neue Anforderungen dergestalt auf, dass plötzlich die Zolltechnik des westlichen Auslands zu beachten war. Ausgangspunkt dafür waren die Festlegungen des SKR. Galt bislang für alle Betriebe der Elektrotechnik/Elektronik das auf metrischen Abmessungen gegründete „Einheitliche Gefäßsystem“ (EGS), so musste nun, begonnen bei den 19“-Schränken bis hin zu den Rastern auf den Leiterplatten, alles auf Zoll-Unterteilungen ausgerichtet sein.

3.3 Das Peripherie-Spektrum

Ausgehend davon, dass periphere Geräte im RGW-Bereich generell Mangelware darstellten, kamen bei allen anwendungsorientierten Finalerzeugnissen vordergründig die im eigenen Kombinat gefertigten Geräte zum Einsatz. Das waren insbesondere

- die seriellen Drucker des Büromaschinenwerkes Sömmerda (BWS),
- die Kassettenmagnetband- und Lochbandgeräte von Rechenelektronik Zella-Mehlis (REZ),
- die Minidiskettenspeicher des Büromaschinenwerkes Karl-Marx-Stadt (BWK) und
- die Tastaturen des Werkes Auerbach.

Bei den Kleinrechnerkonfigurationen waren die Importbeistellungen von Plattenspeichern und Magnetbandgeräten unumgänglich. Da jedoch auf diesem Sektor ständig Lieferengpässe und Qualitätsprobleme auftraten, wurde in den 80er Jahren hierfür im Kombinat eine eigene Entwicklungs- und Produktionslinie aufgebaut.

4 Die wesentlichen Finalerzeugnisse des Projektes DEKK[1]

4.1 Finalerzeugnisse auf Basis von K 1520-Baugruppen

Bürocomputer
Dem Erscheinungsbild des Weltmarktes entsprechend, war es plausibel, 3 körperlich verschiedene Ausprägungen zu entwickeln. Das waren:

Das **Auftischgerät A 5110**, das als Kern einen Drucker benutzte und um einen Bildschirmmonitor ergänzt werden konnte. Die Produktion lag bei BWS.

Das **Auftischgerät A 5120**, das im Grundgefäß einen Monitor anbot und um einen Drucker ergänzt werden konnte und

Das **Standgerät A 5130**, das von vornherein einen Monitor und einen Drucker aufwies und um weitere Geräte ergänzt werden konnte. Beide Ausführungen entstanden bei BWK.
Diese Ausprägungen bildeten die Grundlage für vielfältige Konfigurationen. Abb. 2 zeigt jeweils die Basiskonfiguration und eine typische ausgebaute Ausführung - Über die Jahre hinweg fand dann das Gerät A 5120 den meisten Absatz.

Datenerfassungssysteme

Zur Realisierung der im Betrieb REZ angesiedelten Erzeugnisgruppe Datenerfassungssysteme entstanden mehrere Ausprägungen:

Beim Komplex A 5220 waren unter Bezugnahme auf eine Systemsteuereinheit mit der ZRE K 2521 maximal 8 Dateneingabeplätze K 8913 zu koppeln, die dann eine Datensammlung auf Diskette oder Magnetband ermöglichten.

Durch Kopplung mit speziellen Produktionsdatenerfassungsterminals entstanden Subsysteme A 5230 zur Produktionssteuerung.

Elektronische Schreibsysteme

Äußerlich identisch zum Bürocomputer A 5120 kam eine Konfiguration als Schreibsystem A 5310 auf den Markt. Mit der Einführung universeller Schreibprogramme, abarbeitbar auf Bürocomputern, verlor eine solche Ausführung gar bald die Berechtigung.

[1] Detaillierte Auflistung siehe [Ju06]. Einen äußerlichen Eindruck vermittelt Abbildung 2

Mikrorechnerentwicklungssysteme

Das Mikrorechnerentwicklungssystem A 5601 war das erste Finalerzeugnis des DEKK, das den Anwendern in der DDR angeboten wurde. Es bestand sofort eine große Nachfrage, da die Entwicklung von Mikroprozessorlösungen mit der zugehörigen Programmtechnik ein Gebot der Stunde war. Es wurde als Standgerät mit dem Betriebssystem EIEX realisiert. Konstruktiv stellte es eine Mischung aus Schrank- und K 1520-Technik dar. (siehe Abb.2)

Terminals

***Die einfachen Bildschirmgeräte und Bildschirmterminals*:** Der große Bedarf nach Bildschirmgeräten bzw. Terminals unterschiedlicher Ausprägung führte zu der Erzeugnisreihe K 891X. Diese konnte auf Basis des kleinem Auftischgerätes und der ZRE K 2521 realisiert werden. Als Bediengerät für die Rechner K 1600, als Datenerfassungsterminal, fern aufgestelltes ESER[2]-Terminal und SKR-Terminal war es immer nur eine Programmvariante, die den funktionellen Unterschied erzeugte.

Ein größerer Funktionsumfang konnte auf Basis des großen Auftischgerätes zustande gebracht werden.
Besonders sind diese Lösungen zu nennen:

Universelles Bildschirmterminal (UBT) K 8931: Mit unterschiedlicher Peripherie ausgestattet konnten so z.B. Stapelterminals und Bildschirmterminals für das ESER realisiert werden.

Sparkassenterminal K 8924: Ausgerüstet mit spezieller Tastatur und Sparbuchdrucker und eingebunden in das ESER-Programmsystem Geldwirtschaft stellte dieses Gerät viele Jahre die Basis für die Automatisierung in Geldinstituten dar.

Platzreservierungsterminal K 8927: Bei der Reichsbahn, im Hotelwesen und in den Reisebüros war dieses Reservierungsterminal mit Trennung des Monitors vom Grundgerät nicht mehr wegzudenken. Als Sololösung oder eingebettet in eine ESER-Lösung wurde die Automatisierung vorangetrieben.

[2] ESER = Einheitliches System elektronischer Rechentechnik

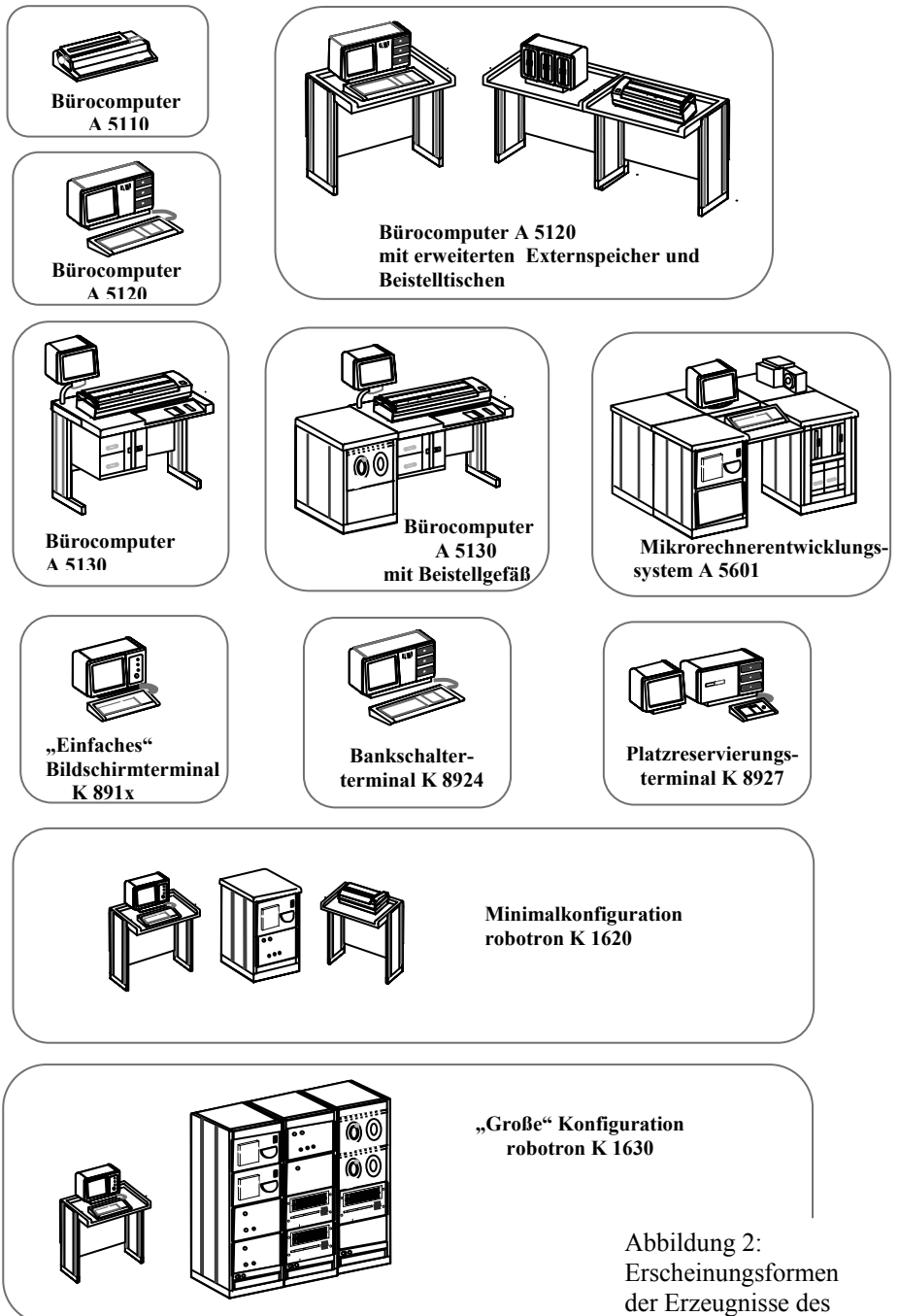


Abbildung 2:
Erscheinungsformen
der Erzeugnisse des
Konzeptes DEKK

Kommunikationsprozessoren [3]

Multiplexor und Konzentrator für Kommunikationslösungen: Die in der Ära des Rechners R 4201 entstandenen Lösungen für das ESER wurden nunmehr mit K 1520-Lösungen abgelöst und derartige Lösungen auch für das K 1600-Zeitalter bereitgestellt.

Die Unterbringung der Baugruppen erfolgte in Schränken, angepasst an die jeweiligen Systeme ESER und SKR.

4.2 Finalerzeugnisse auf Basis von K 1600-Rechnern

Das Konzept der Rechnerfamilie K 1600 gestattete vielfältige Anwendungslösungen. Die Unterscheidungsmerkmale dieser Lösungen waren dabei durch eine Spezialperipherie, mehrere zweckorientierte Betriebssysteme und die jeweiligen Anwendungsprogramme gegeben.- Insofern, wie komplette Lösungen ausgeliefert wurden, kamen diese unter einer Anwendungschiffre mit dem Kennbuchstaben „A“ zur Auslieferung.

Als Anwendungslösungen wurden entwickelt und angeboten:

Das Kommerzielle Basisrechnersystem A 6401/A 6402 mit Haupteinsatz für wissenschaftlich-technische - Berechnungen und kommerzielle Aufgaben,

das **Daten- und Informationssystem A 6220** für umfangreiche Datensammlung und -aufbereitung, geprägt durch Datenstationen und, Subsysteme zur Produktionsdaten-erfassung,

der **Arbeitsplatz für Konstruktion und Technolgie A 6452/A 6454** für Rechner-gestützte Entwicklung, Projektierung und Produktionsvorbereitung, ausgerüstet mit Digitalisiergeräten, Plottern und Zeichentischen,

das **Bildverarbeitungssystem A 647x** zur Auswertung bildhafter Informationen in der Fernerkundung, Medizin, Biologie usw. auf Basis von Bildspeicher, Filmein-/-ausgabe und Grafikterminals sowie die **Prozessrechnersysteme A 6491/A 6492**, gekennzeichnet durch die Prozesseingabe-/Prozessausgabeeinrichtung.

---

[3]Im Rahmen des Erzeugnisspektrums DEKK wurden einerseits Geräte projektiert und realisiert, die den Bedarf an Kommunikationskomponenten, wie Terminals und Kommunikationsprozessoren a priori befriedigen konnten und andererseits entstand durch die entsprechenden Interfaces und Kommunikationssoftware aus allen Anwendungslösungen, wie z.B. den Bürocomputern eine terminalfähige Lösung. Das ermöglichte in Breite die Einbringung in die Nomenklatur des ESER und des SKR und stellte damit eine wichtige Grundlage für die Exportfähigkeit dar.

5 Das Softwarespektrum

5.1 Software für Geräte mit K 1520-Baugruppen

Betriebssysteme

Es war Fakt, dass zum Zeitpunkt der Ableitung des Erzeugnisprogrammes DEKK bereits in 3 verschiedenen Betrieben an 3 unterschiedlichen Zentralen Recheneinheiten, die den Mikroprozessor U 880 enthielten, gearbeitet wurde. Gleichermaßen lief zu diesem Zeitpunkt die Entwicklung an 3 auf den jeweiligen Mikroprozessor bezogenen Betriebssystemen. Trotz enger Arbeitsgruppentätigkeit gelang es nicht, eine einheitliche Linie durchzusetzen, weil jeder dem anderen beweisen wollte, dass sein Ansatz der einzig richtige war. So resultierten daraus auf Dauer 3 verschiedene Betriebssysteme aus Eigenentwicklung ohne Vorbilder. Für die ZRE 2526 kamen dann wegen des breitesten Einsatzes sogar noch weitere hinzu.

EIEX 1521: Für die im ZFT entwickelte ZRE K 2521, die als Basis der Rechner für alle Echtzeitanwendungen dienen sollte, wurde das Echtzeitsteuerprogrammsystem EIEX 2521 entwickelt. Es wurde z.B. im Mikrorechnerentwicklungssystem und den Terminals bei Robotron aber auch bei vielen Anwendern der DDR für unterschiedliche Anwendungen genutzt.

BROS 1525: Zugeschnitten auf die ZRE K 2525 des druckerorientierten Bürocomputers A 5110 des Büromaschinenwerkes Sömmerda entstand dort das Betriebssystem BROS 2525. Wegen der eingeschränkten Anwendung kam es am wenigsten zum Einsatz.

SIOS 1526: Die Doppelprozessorlösung der ZRE 2526 machte ebenfalls eine eigene Ausführung in Form des Betriebssystems SIOS 2526 erforderlich, die dann bei den Bürocomputern A 5120 und A 5130 Anwendung fand.

UDOS 1526: Für die Programmentwicklung auf den Bürocomputern A 5120/A 5130 entstand darüber hinaus das Betriebssystem UDOS 1526. Dadurch wurden die Geräte A 5120/A 5130 zu einem effektiven Programmentwicklungsplatz für die in der DDR produzierten oder in Entwicklung befindlichen Mikroprozessoren (U880, U881, U8000) aufgewertet. Daneben konnte die Abarbeitung von Anwendungsprogrammen stattfinden. - Strukturell stellte UDOS 1526 ein ausschließlich diskettenorientiertes Betriebssystem dar.

SCP 1526: Um auf dem NSW-Markt eine Absatzchance zu besitzen, wurde das weit verbreitete CP/M als SCP (Single User Control Program) für die ZRE 2526 adaptiert.- Es wurde für die Entwicklung, Testung und Abarbeitung von Programmen genutzt, wie sie für die Klasse der Bürocomputer typisch sind.
Als Kern dient die Executive SCPX 1526. Sie ist ein diskettenorientiertes Steuerprogramm und gestattet die Abarbeitung von Programmen im Einzelnutzer- und Einzelaufgabenbetrieb.

Programmiersprachen und Assembler

Es kamen die zur jeweiligen ZRE bereitgestellten Assembler zum Einsatz. Für die genannten Betriebssysteme wurden die zur damaligen Zeit üblichen Programmiersprachen entwickelt oder adaptiert. Es handelte sich um

BASIC; COBOL; FORTRAN; PLZ (nur für UDOS), C (nur für SCP)

Zur durchgängigen Nutzung bestimmter Programme kam die Sprache PASCAL bei der Programmentwicklung zum Einsatz. Jedes Finalerzeugnis wurde mit einem Paket von Anwendungssoftware ausgeliefert.

5.2 Software für Finalerzeugnisse mit K 1600-Rechnern

Um der Funktion als Mittler zwischen Gerätetechnik und den Anwendungsprogrammen bei den stark unterschiedlichen Anwendungssystemen gerecht werden zu können, wurden mehrere Betriebssysteme mit unterschiedlichen Ausrichtungen angeboten.. Im Einzelnen waren das:

MOOS 1600: Als Modulares plattenspeicherorientiertes Betriebssystem war es zum Einsatz in Anwendungssystemen der Datensammlung, Datenverarbeitung in wissenschaftlichen und technischen Bereichen und zur Prozessüberwachung und –steuerung vorgesehen..- Es unterstützte die höheren Programmiersprachen BASIC und FORTRAN IV.

LAOS 1600: Das Betriebssystem für Laborautomatisierung war ein kleines plattenspeicherorientiertes Betriebssystem. - Es unterstützte die höheren Programmiersprachen BASIC; PASCAL; COBOL; FORTRAN IV und CDL.

MUTOS 1630: Hierbei handelte es sich um ein plattenspeicherorientiertes Timesharing-System, also ein Universelles interaktives Betriebssystem als Summe von Programmmoduln, kompatibel zu UNIX V zur Programmentwicklung, Lösung wissenschaftlich-technischer Probleme, Textverarbeitung und zur Lösung kommerzieller und ökonomischer Probleme.

OMOS 2.0: Dieses Optimierte Modulare Betriebssystem als Weiterentwicklung des MOOS 1.2 brachte ein verbessertes Laufzeitverhalten durch ein verbessertes Dateizugriffssystem FCS und stellte weitere ergänzende Merkmale, wie z.B. einen Gleitkomma-Emulator bereit.

6 Entwicklungs- und Produktionszeitpunkte der Haupterzeugnisse der Linie DDT

Die Erzeugnisse des DEKK wurden beginnend im Jahre 1980 bis zur Auflösung des Kombinates im Jahre 1990 weiter entwickelt und in mittleren bis großen Stückzahlen produziert.[4] Daneben fand aber eine Ablösung insbesondere bei den Bürocomputern durch die effektiveren Personalcomputer statt, deren konstruktive Grundlage, das „Motherboard“ noch heute Kern dieser Erzeugnisse darstellt.

7 Bewertung

Man kann heute ohne Übertreibung feststellen, dass mit dem Erzeugnisprogramm DEKK bis Mitte der 80er Jahre eine wesentliche Entwicklungsrichtung des Kombinates Robotron gewiesen wurde, die sich breit in der DDR niederschlug, und tief in den Bereich des RGW einwirkte, allerdings auf dem westlichen Markt aus Kostengründen nur zaghaft zu Ergebnissen führte.

Mit dem Konzept DEKK wurde eine effektive Basis für das Zusammenlegen der Kombinate Robotron und Zentronik geschaffen. Für die Anwender in der DDR war die Orientierung auf den Mikroprozessor U 880 ein Glücksgriff, der natürlich in der Hauptsache von der Bauelemente-Industrie ausging. Die darauf aufbauenden Anwendungslösungen erfüllten vielfältig die Anforderungen der 80er Jahre und brachten auch Forschung und Lehre voran.

Die Orientierung auf die Bauelemente-Serie U 830 als Basis der Minicomputer K 1600 warf die Linie Minicomputer leistungsmäßig zurück, sodass Mitte der 80er Jahre für CAD/CAM-Lösungen in Hektik auf den Nachbau der VAX-Linie orientiert wurde.

Literaturverzeichnis

[Un78] Unbenannt: Mikrorechnersystem K 1520, NTB (1978) H. 5, S. 136

[Gi80] Giebler, H. :Mikrorechnersystem K 1600, NTB (1980) H. 2, S. 39

[Ju06] Junge, S.: Erzeugnislinie Dezentrale Datentechnik: http://robotron.foerderverein-tsd.de/35/robotron35a.pdf

[We06] Weise, K.-D.: Zusammenstellung der im Kombinat Robotron produzierten Erzeugnisse: http://robotron.foerderverein-tsd.de/3/robotron3a.pdf

[4] Beispiele für Stückzahlen über den Gesamtzeitraum der Produktion (siehe [We06]):
K 1520-Sätze: 200.000
A 5110: 4225
A 5120: 17.000
A 5130: 5.640
K 1600-Konfigurationen : 2.125

Entwicklung der Software für Heim-, Klein- und Bildungscomputer als Basis für die Ausbildung in Schulen und Berufsschulen der DDR

Gert Keller
Dresden
g.keller@kelleworld.de

Gunter Kleinmichel
Dresden
gunter.kleinmichel@arcor.de

Abstract: Der Vortrag berichtet über die Entwicklung der Systemsoftware und der Programmiersprachen für die Heim-/ Kleincomputer der Serie KC 85/ KC 87 und den Bildungscomputer BIC A5105. Die Computer KC 85/ KC 87 wurden zunächst als Heimcomputer konzipiert, später aber auf staatlichen Beschluss vor allem in der Ausbildung und der Industrie eingesetzt, während die Bildungscomputer vorn vornherein ausschließlich für unterschiedliche Anforderungen des Bildungswesen ausgelegt wurden.

Unter diesen Gesichtspunkten ist auch die bei Robotron Meßelektronik in Dresden durchgeführte Softwareentwicklung zu sehen, die vor allem Basislösungen und Demonstrationsbeispiele für die genannten Anwendungen bereitstellen sollte. Unter Beachtung der durchaus beschränkten Hardware-Möglichkeiten (verfügbare Bauelemente, externe Programmspeichermedien, Monitore) dieser Computer entstand eine Reihe eigenständiger Software, die sich an internationalen Entwicklungen orientierte (Sprachsyntax, Schnittstellen). Vielfältige Kooperationen mit Firmen und dem Bildungswesen der DDR erweiterten die Anwendungsmöglichkeiten wesentlich.

1. Heim-, Klein- und Bildungscomputer – Entwicklungsziele und Voraussetzungen

Ausgangspunkt für die Entwicklung der Heimcomputer (Z9001) im Kombinat Robotron im Sommer 1983 war allein die politische Auflage des X. Parteitages der SED im Jahre 1981 nach Bereitstellung von hochwertigen Konsumgütern für die Bevölkerung. Damit sollte der wachsenden Unzufriedenheit mit der allgemeinen Versorgungslage und dem kritiklosen Begehren nach westlichen Erzeugnissen begegnet werden. Der VEB Robotron-Meßelektronik Dresden stellte sich damals das Ziel, Heimcomputer nach westlichem Vorbild für die private Nutzung im Heimbereich bereitzustellen. Diese Idee basierte auf bereits seit Ende 1982 von einem Jugendkollektiv im Kombinat Robotron entwickelten Prototypen eine 8-bit Heimcomputers, die aber ohne die oben beschriebene politische Auflage wegen der äußerst angespannten Versorgungssituation mit elektronischen Bauelementen und fehlender Entwicklungs- und Produktionskapazität keine Realisierungschance gehabt hätten. Dazu kam die Tatsache, daß gleichzeitig im damaligen Kombinat Mikroelektronik, und zwar im VEB Mikroelektronik Mühlhausen, ebenfalls ein Erzeugnis Heimcomputer entwickelt wurde, das unter der gleichen politischen Auflage geplant war und unter gleichen materiellen Behinderungen litt. Ein damals durchaus sinnvolles Zusammengehen beider Betriebe oder eine Vereinheitlichung der Entwicklung scheiterte an betriebsegoistischen Interessen der beiden Kombinate. Lediglich nach energischem Einspruch der "Aufsichtsbehörde" für elektronische Konsumgüter, dem damaligen Amt für Wirtschaftsprüfung und Standardisierung (ASMW) und einer entsprechenden Weisung durch das übergeordnete Ministerium für Elektrotechnik/Elektronik der DDR , wurde ein Teil der Basis-Software, die von Meßelektronik Dresden für den Heimcomputer bereitgestellt wurde, auch für das Mühlhausener Erzeugnis adaptiert und übernommen. Erst nach Bereitstellung der ersten Entwicklungsmuster und umfangreicher öffentlicher Vorstellung dieser Geräte entwickelte sich eine gewisse Euphorie für den plötzlich real vorstellbaren Einsatz im Heimbereich, der trotz deutlicher Leistungsunterschiede zu westlichen Heimcomputern, insbesondere hinsichtlich der Bereitstellung von angepaßten Monitoren und geeigneten Massenspeichern und einem relativ geringen Einzelhandelspreis zu einer enormen Nachfrage ab Verkaufsbeginn Dezember 1984 führte.

Wegen der mangelnden Verfügbarkeit von Computertechnik in der Industrie und in offensichtlicher Abkehr von der oben genannten politischen Zielstellung wurde jedoch zur Frühjahrsmesse 1985 durch das ZK der SED beschlossen, die Computer von diesem Zeitpunkt an nur noch an Schulen bzw. in die Industrie (Mess- und Automatisierungsaufgaben) zu liefern. Unter dieser neuen Zielstellung erfolgten eine formale Umbenennung in Kleincomputer KC85/1, eine völlig neue Preisgestaltung, die unter DDR-Bedingungen praktisch einen Verkauf an die Bevölkerung unmöglich machte und später die Weiterentwicklung zum KC 87. Der Computer wurde bis Mai 1989 mit insgesamt etwa 25 000 Exemplaren gefertigt. Eine technische Weiterentwicklung für den Heim- oder Industriebereich wurde bei Meßelektronik zwar angedacht, aber von staatlichen Stellen unterbunden und dem Kombinat Mikroelektronik zugewiesen.

Stattdessen wurde, beginnend 1988, ein Bildungscomputer BIC A5105 nach Vorgaben es Bildungswesens der DDR und gemeinsam mit der Akademie der Pädagogischen Wissenschaften konzipiert, wobei die Erfahrungen aus der Nutzung des KC85 den Forderungskatalog maßgeblich beeinflussten. Dieser Computer sollte ausschließlich in Schulen und Berufsschulen eingesetzt werden und dort in Computerkabinetten die Informatik- und die ESP-Ausbildung unterstützen. Es wurden wenige Tausend Stück des BIC produziert, die vorwiegend an Erweiterte Oberschulen (Gymnasien) und einige Berufsschulen geliefert wurden. Zum Termin der Währungsunion 1990 wurde die Produktion eingestellt, da diese dann kostenseitig nicht mehr vertretbar war. Trotzdem wurden die ausgelieferten Computer in vielen Schulen einige Jahre erfolgreich genutzt.

2. Heim- und Kleincomputer KC 85/1 und KC 87

2.1 Hardwarekonzeption und Basissoftware

Technischer Ausgangspunkt der Entwicklung des Heimcomputers Z9001 war das Ziel, einen Computer für die private Nutzung entsprechend der Möglichkeiten der Bauelementeindustrie der DDR (und des RGW) zu entwickeln, der in größerer Stückzahl gefertigt werden sollte. Im Vergleich zum westlichen Stand (Sinclair Spectrum, C64, Atari) waren die Hardwaremöglichkeiten vor allem bezüglich folgender Eigenschaften begrenzt:

Speichervolumen:
adressierbarer Speicherbereich von 64 KByte, davon intern 6 KByte ROM für Betriebssystem und Zeichengenerator,
16 KByte RAM, erweiterbar durch Zusatzmodule,
verfügbare Schaltkreise waren zunächst 2 KByte EPROM (U556) und 16 Kbit RAM (U256), später standen 64 Kbit RAM-Schaltkreise zur Verfügung.

Grafikausgabe:
keine Grafikcontroller für Pixelgrafik und Farbausgabe und für die Spielgestaltung (keine Hardware-Sprites) verfügbar, deswegen nur „Quasigrafik“ realisiert**.**

Tastatur:
Nutzung einer ergonomisch ungünstigen Tastatur,
aus Kostengründen keine übliche Schreibmaschinentastatur einsetzbar

Massenspeicher:.
handelsübliche Heimmagnetbandgeräte als Massenspeicher, da keine Floppy Disc für Heimanwendung verfügbar

Diese Nachteile wurden durch einige konzeptionelle Ideen und eine ausgereifte Konstruktion, die aus den Erfahrungen der Messgeräteentwicklung mit dem System K1520 gewonnen wurden, teilweise ausgeglichen,

- Erweiterbarkeit des Basisgerätes durch einfaches und robustes Modulkonzept (getriebener K1520-Bus),

- hohe Robustheit in Hard- und Software, auch durch sehr gutes Qualitätsmanagement

Die Basissoftware der Computerserie bestand, wie bei Heimcomputern damals üblich, aus einem BASIC-Interpreter, der aber (im Gegensatz zu einigen Vorbildsystemen), auf einem an CP/M orientiert modulares Betriebssystem aufsetzte, was später für einige Erweiterungen von großem Vorteil war. Dadurch konnte auch der Übergang von privaten Anwendungen als Heimcomputer zu Anwendungen in der Industrie bzw. der Ausbildung als „Kleincomputer“ besser unterstützt werden. (Dieser BASIC-Interpreter war zunächst nicht im Grundgerät implementiert, sondern mußte als Zusatzmodul gesteckt oder vom Massenspeicher Tonband nachgeladen werden.)

Durch die Änderungen der Anwendungsschwerpunkte der Computer kam es natürlich auch zu Änderungen der Strategie in der Modul- und Software-Entwicklung.

2.2 Das BASIC der Kleincomputer

Die Entwicklung des BASIC-Interpreters im VEB Robotron Meßelektronik „Otto Schön“ Dresden (MKD) erfolgte auf Basis eines von einem Institut der Landwirtschaft in Rostock-Dummersdorf bereitgestellten BASIC-Interpreters, der sich sehr am Kern des damaligen Microsoft- BASIC orientierte.

Da bei MKD seinerzeit bereits umfangreiche Erfahrungen in der BASIC-Nutzung auf höherwertigen PC (hp9845) vorlagen, konnten einige sinnvolle Spracherweiterungen durchgeführt werden. Umfangreiche Tests zeigten außerdem Fehler in den Grundroutinen wie sie bereits bei Microsoft auftraten, die vor Auslieferung der Geräte bei MKD beseitigt wurden.

Gemäß einer bereits 1983 erfolgten Weisung des Ministeriums für Elektrotechnik/Elektronik wurde der 8k-Kern des BASIC so gestaltet, das er auch für den gleichzeitig in Entwicklung befindlichen KC85/3 des VEB Mikroelektronik Mühlhausen verwendbar war. Das BASIC wurde vereinbarungsgemäß von den Kollegen in Mühlhausen übernommen und dort um Pixelgrafik-Befehle ergänzt. In einer 2K-Erweiterung des KC85/1 wurden schrittweise Zusatzfunktionen eingebaut.

Die gemeinsame Nutzung des BASIC-Interpreters und weiterer Basissoftware (auch die Kassettentreiber für die Programmspeicherung waren kompatibel) sollte die Übertragung von Software zwischen beiden Systemen ermöglichen. Praktisch hatte das für die Nutzer später wahrscheinlich wenig Bedeutung, da damals viele Programme noch mit direkten Zugriffen in den Bildspeicher (PEEK- und POKE- Befehle) geschrieben wurden, die wegen der unterschiedlichen Adressbereiche der Rechner nicht kompatibel waren.

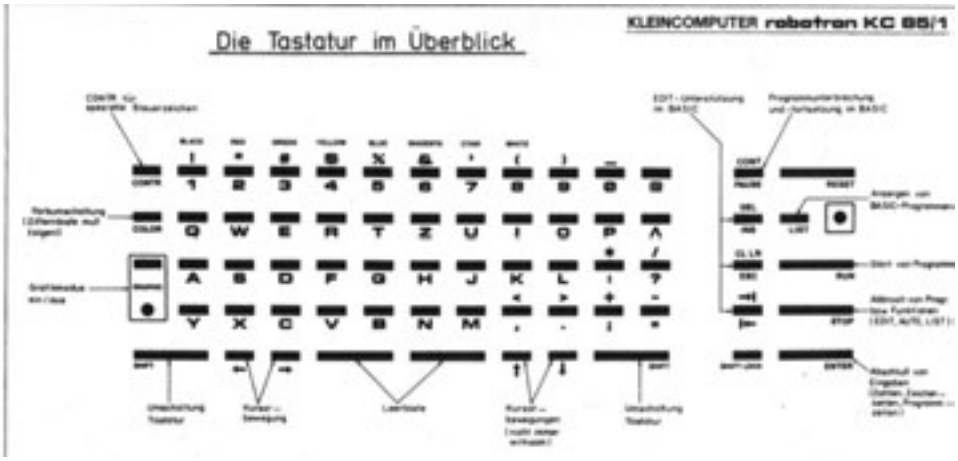


Abbildung 1: Tastatur der Kleincomputer KC85/1 und KC87.

2.3 Betriebssystem und Treiber

Als Besonderheit für Heimcomputer wurde das Betriebssystem für die Heim- und Kleincomputer modular gestaltet (BIOS- und BDOS-Ebenen) und die Schnittstellen dabei an Microsoft CP/M (DDR-Bezeichnung SCP) orientiert, was für die Übernahme verschiedener Treiber und unterschiedlicher Anwendersoftware auf die KC später von großem Vorteil war. Der Umfang des Betriebssystems (ohne BASIC.Interpreter) mußte wegen der begrenzten ROM-Kapazität auf 3 KByte beschränkt werden.

Bezüglich der Treiber sei besonders erwähnt, dass wegen der unterschiedlichen Drucker und Schreibmaschinen, die in den Kombinatsbetrieben in Sömmerda bzw. Karl-Marx-Stadt hergestellt wurden, insgesamt mehr als 10 verschiedenen Drucker (Nadeldrucker, Thermodrucker, Schreibmaschinen) angeschlossen werden konnten, deren Treiber für die KC-Serie alle bei MKD erstellt wurden (heute ist der Druckerhersteller genötigt, kompatible Treiber für seine Geräte zu liefern, damals war das umgekehrt!).

2.4 Ergänzungen für Heim- und Kleincomputer und Zusatzsoftware für Module

Durch die für die Softwareentwicklung bei MKD verantwortliche Abteilung wurde weitere Zusatzsoftware für die Computer erarbeitet, wie z.B.:

- Demoprogramme, die die Möglichkeiten des Computers und Applikationsbeispiele zeigen,

- Treiber für einige weitere Zusatzmodule, wie z.B. Speichererweiterungen RAM/ROM, Drucker-, Schreibmaschinen- und Plotteranschluss, digitale Ein-/Ausgabe, Spracheingabe, Analog-Digital-Umsetzer, EPROM-Programmierung.

Hervorzuheben ist auch der Anschluss eines A4-Plotters aus der CSSR (Typ XY 4131), für den der Treiber an den KC angepasst und eine Demosoftware bereitgestellt wurde.

In Kooperation mit anderen Robotron-Betrieben, einigen Hochschulen und der Industrie wurde weiter Software bereitgestellt, wie z.B. ein damals häufig benutzter Assembler, die Programmiersprache Pascal, Kassettentreiber für höhere Übertragungsraten, Beispiele für Musiksoftware und anderes.



Abbildung 2: Kleincomputer KC 87 (mit Baugruppen, Zubehör und Peripherie)

Der Vertrieb der Software durch Robotron erfolgte zum großen Teil durch Kassetten, die bei der Schallplattenfirma Amiga vervielfältigt wurden. Es ist klar, dass unter DDR-Bedingungen der Handel bzw. der Austausch von Software innerhalb der Nutzerkreise eine hohe Bedeutung erlangte.

2.5 Einsatzgebiete der Kleincomputer

Wie bereits erwähnt, wurde der Computer zunächst als Heimcomputer (Z9001) entwickelt, war aber für den Heimbereich nur kurzzeitig (Dezember 1984 bis März 1985) verfügbar. Danach erfolgte die Auslieferung der Computer nur noch für das Bildungswesen und die Nutzung als Automatisierungsmittel in der Industrie.

Im **Bildungswesen** wurde zunächst begonnen, sich mit den Möglichkeiten der Computer und deren Einsatz in der Ausbildung zu beschäftigen. In Zusammenarbeit zwischen dem Hersteller MKD und dem Ministerium für Volksbildung (vertreten vor allem durch APW Berlin, PH Güstrow, PH Dresden, TU Dresden) wurde die Einsatzmöglichkeiten in den Schulen teilweise kontrovers diskutiert und erste Erfahrungen mit Computern in unterschiedlichen Schultypen und Klassenstufen gesammelt. Man muss aus heutiger Sicht feststellen, dass die Computertechnik auch für die Verantwortlichen für Lehrpläne seinerzeit völliges Neuland war.

Dies zeigte sich auch im Hochschulbereich, in dem bis dahin die Programmierung über Lochkarten und Lochstreifen überwog und mit den Kleincomputern erstmals in größerem Umfang zur Programmierung von Bedienoberflächen übergegangen werden musste, was völlig neue Aspekte der Computernutzung, manche Überraschung und auch manche Irrtümer mit sich brachte. Insgesamt wurden die Kleincomputer aber auch im Hochschulbereich umfangreich genutzt. Es entstanden dort Lösungen (Treiber, Speziallösungen für Datenspeicherung, Programmiersprachen u.a.), die dann auch bei Industrieanwendungen nützlich eingesetzt wurden.

Die Nutzung des Computers für die Ausbildung wurde auch dadurch unterstützt, dass Bedienungsanleitung und Programmierhandbuch sehr sorgfältig erstellt wurden (alle Beispiele wurden mehrfach am Computer geprüft), so dass sie teilweise auch als Lehrbuch eingesetzt werden konnten (Das Programmierhandbuch wurde für die Berufsausbildung und Weiterbildung mit 20000 Exemplaren nachgedruckt). Das Buch wurde damals als Beispiel für den Einstieg in die Programmierung auch im Ausland sehr gelobt.

In der Industrie selbst wurden erstaunlich viele Anwendungsgebiete erschlossen. Vor allen entstanden viele kleinere Mess- und Prüfplätze, die teilweise noch langjährig zur Prüfung von Geräten oder Bauteilen eingesetzt wurden. Auch in der Industrie wurde der Lerneffekt (Was kann ein Computer?) deutlich unterstützt und so weitere Anwendungen auf höherem Niveau vorbereitet.

Die Wirkung der Kleincomputer aus heutiger Sicht lässt sich vielleicht in folgender Weise kurz zusammenfassen:

- Es entstand ein Ausbildungseffekt von nicht zu unterschätzender Wirkung für Schüler und Studenten,
- In der Industrie wurden zahlreiche interessante Lösungen geschaffen, die in Prüflabors teilweise über mehr als 5 Jahre genutzt wurden,
- Die Nachteile der fehlenden (Pixel-)Grafik und der unhandlichen Tastatur konnten natürlich nie beseitigt werden,

- Die Kleincomputer bildeten eine wesentliche Basis für die Informatik-Ausbildung in den Schulen und Berufsschulen der DDR. Auch für den speziellen „produktionsvorbereitenden“ Unterricht (ESP) wurden interessante Grundkonzepte geschaffen.

Bemerkenswert war die erstmals seit langem in der DDR zugelassene parallele Entwicklung von 2 Geräten (Kleincomputern) mit ähnlichen Eigenschaften in 2 verschiedenen Betrieben bzw. Kombinaten. Diese Konkurrenzsituation hat zwar die Qualität beider Computersysteme (Dresden und Mühlhausen) positiv beeinflusst, führte aber zu Doppelentwicklungen (z.B. Grafikprozessor und PROM für BASIC-Interpreter) und damit zu einer Zersplitterung der ohnehin knappen Ressourcen.

3. Bildungscomputer A 5105

3.1 Grundkonzeption, Betriebssystem und Basissoftware

Das Grundkonzept des Bildungscomputers robotron A5105 entstand 1986/87 aus

- einem von der Akademie der Pädagogischen Wissenschaften ausgearbeitetem Forderungskatalog des Bildungswesens der DDR,

- den für das Kombinat Robotron (bzw. den VEB Robotron Meßelektronik) erreichbaren technischen Möglichkeiten,

- nach Abstimmung des zulässigen/möglichen Bauelementeeinsatzes mit dem Ministerium für Elektrotechnik/Elektronik

Ziel war es, einen im gesamten Bildungswesen einsetzbaren universellen Computer zu entwickeln, der die Anforderungen unterschiedlicher Ausbildungsziele (Informatik, Büroanwendungen, Einführung in die Automatisierungstechnik, ESP-Unterricht, Vernetzung im Computerkabinett) gleichzeitig erfüllt. Der von Anfang an geplante Produktionsbeginn im September 1989 erfolgte termingerecht. Die Fertigung wurde etwa bis Sommer 1990 fortgesetzt und insgesamt ca. 5000 BIC produziert.

Der BIC bestand aus 3 Hauptbaugruppen, nämlich dem Computergrundgerät (CGG), der Diskettenspeichereinheit (DSE) und dem Monitor (MON). Dabei war das Computergrundgerät so konzipiert, dass es auch als eigenständiger Computer (mit separatem Netzteil, Kassettenmagnetbandgerät und TV) betrieben werden konnte und somit eine Nutzung als „Heimcomputer“ denkbar gewesen wäre.



Abbildung 3: Bildungscomputer robotron A 5105

Bezüglich der Hardware-Komponenten waren damals folgende Entwicklungen bemerkenswert:

- die Entwicklung von 2 ASICs für Grafik und Speicheransteuerung / Sound (die ASICs wurden durch Meßelektronik und das ZMD (Zentrum für Mikroelektronik Dresden) gemeinsam entwickelt und dann im ZMD gefertigt,

- Grafik-Bildspeicher und zugehörige Ansteuerung waren praktisch als 16-Bit-Lösung aufgebaut und in die 8-Bit-Computerarchitektur integriert. Der Bildspeicher umfasste 128 KByte, zusammengesetzt aus 4x64-kBit Schaltkreisen, die jedoch zunächst aus NSW-Importen abgedeckt werden mussten,

- der Computer hatte eine Vielzahl von fest eingebauten Schnittstellen und bot ein kostengünstiges und zuverlässiges Vernetzungsinterface für Computerkabinette.

Ebenfalls zur Unterstützung unterschiedlicher Ausbildungsziele wurde der BIC mit zwei Betriebssystemen ausgeliefert. Für die Einführung in die Computernutzung (Basic-Programmierung wurde das Betriebssystem RBASIC entwickelt. Für weitergehende (Büro-) Anwendungen wurde SCP 5105 bereitgestellt (eine Anpassung des Betriebssystems SCP 1715 des Sömmerdaer PC 1715)

3.2 RBASIC und Grafikeigenschaften

Die Entwicklung des RBASIC orientierte sich am damals in Asien vorherrschenden MSX-BASIC von Microsoft. Besonderheiten für den BIC waren dabei:

- Die Anpassung der Grafikbefehle an die (für 8-Bit-Computer guten) Hardware-möglichkeiten der Grafik (ASIC, 128kByte) verbunden mit einer fehlerfreien Grafikdarstellung auch für Sonderfälle (bei Microsoft seinerzeit nicht üblich).

- die Übernahme einiger bewährter und in der DDR eingeführter Anweisungen aus dem KC-BASIC (Sprachsyntax, Funktionen).

3.3 SCP5105 und Anwendersoftware

Das CP/M-kompatible Betriebssystem SCP5105 entstand aus dem Betriebssystem des PC1715 (Sömmerda) durch Anpassung der Treiber (BIOS) an die Hardware des BIC. Da der BIC aus Kostengründen nur ein Diskettenlaufwerk besaß, musste auch die Software zum Kopieren von Dateien angepasst werden, wobei die 128 KByte Bildspeicher effektiv als RAM-Disk genutzt werden konnten.

Die Lösung gestattete die Nutzung der gesamten für den PC 1715 vorhandenen Software, die sowohl durch Robotron bereitgestellt wurde, als auch in der Industrie vorhanden war. Das Betriebssystem SCP sollte in der Ausbildung z.B. für Textverarbeitung und Tabellenkalkulation benutzt werden, da mittlerweile erkannt worden war, dass die Computernutzung für viele Anwender nicht nur in der Programmierung liegen wird.

3.4 BIC A5105 - Entwicklung und Fertigung mit Partnern

Entwicklung und Fertigung des BIC erfolgte in enger Kooperation mit verschiedenen Betrieben der Kombinate Robotron, KEAB und Mikroelektronik:

Betrieb	**Anteil an FuE-Kooperation**	**Anteil an Fertigung**
Robotron Sömmerda	Bereitstellung Software SCP715	Monitor, Drucker
Robotron Karl-Marx-Stadt	Basistreiber für Disketten	Disketten-Laufwerk

Betrieb	Anteil an FuE-Kooperation	Anteil an Fertigung
Robotron Auerbach	Tastaturentwicklung nach Vorgabe	Tastatur
Robotron Riesa		Leiterplattenbestückung und -prüfung
Robotron Radeberg		Gehäuseteile für CGG
Robotron Hoyerswerda		Netzteil
KEAB Berlin		Diskettenspeichereinheit
ZFM Dresden	ASICs	ASICs
TU Dresden	Basislösung für Computernetz	

3.5 Nutzungskonzepte des Bildungscomputers für Schulen und Berufsschulen

Nachdem die Konzipierung des BIC bereits 1986 begann und nach langen Diskussionen mit der APW 1987 abgeschlossen wurde, erfolgte eine frühzeitige Information wichtiger Partner des Bildungswesens über die Leistungsmerkmale des geplanten Computers. Da auch Details der Software-Gestaltung schon 1988 übergeben wurden, waren vor allem die Pädagogischen Hochschulen in der Lage, rechtzeitig mit der Ausarbeitung neuer Lehrpläne zu beginnen. An der PH Güstrow entstanden sogar Versuchsmodelle, die später im ESP-Unterricht zur Demonstration von Automatisierungsabläufen eingesetzt werden sollten.

Erste Muster des BIC wurden bereits ab Mai/Juni 89 dem Bildungswesen übergeben. Neben dem Test der Computer, der wichtige Hinweise für Entwicklung und Fertigung ergab, dienten diese Muster zur weiteren Ausarbeitung der Lehrpläne für Schulen und Berufsschulen. In dieser Phase waren an der Vorbereitung der Computereinführung viele Partner im Bildungswesen beteiligt, außer der federführenden APW z.B. PH Güstrow, PH Dresden, und einige Testschulen, die bereits Erfahrungen mit Kleincomputern der KC-Serie hatten. Dadurch gelang es dem Bildungswesen, bereits per September 1989 umfangreiche Software auf Disketten mit Demo- und Lehrsoftware fertig zu stellen. (Natürlich wurde vieles davon letztlich durch die politische Wende nicht in geplanter Weise genutzt.)

Die letzte wesentliche Aktion aus Sicht der FuE-Arbeiten war die Übergabe der Vernetzungssoftware (BICLAN) im April 1990 an der Heinrich-Heine-Schule in Berlin an die APW. Danach erfolgte auch die letzte Zahlung für Entwicklungsleistungen vom Bildungswesen an den Herstellerbetrieb VEB Robotron Meßelektronik „Otto Schön“.

Mit dem Bildungscomputer BIC A5105 wurden bis Mitte 1990 einige Schulen, darunter fast alle Erweiterten Oberschulen der DDR mit Computerkabinetten ausgerüstet. Die Nutzung war, auch wegen der unterschiedlichen „politischen Beurteilung“ der Computer, dann bezüglich der inhaltlichen und zeitlichen Nutzung sehr unterschiedlich. Einige Schulen benutzten die BIC teilweise aber noch bis Ende der 90er Jahre.

Literaturverzeichnis

[G1] Gärtner, G.: Zusatzsoftware für Bildungscomputer, Mikroprozessortechnik 4/1990, S. 122

[K1] Keller, G.: Basic für Heimcomputer Z 9001, radio fernsehen elektronik 33 (184) H.9, S. 586-587, 593

[K2] Keller, G., Kleinmichel, G.: Bildungscomputer robotron A 5105; Neue Technik im Büro 33 (1989)2, S.62-64

[K3] Keller, G. Der neue Kleincomputer im Überblick, Mikroprozessortechnik, 1 (1987) H12, S. 22-25

[K4] Kerner, I.O.: Aspekte der Informatik-Ausbildung im Volksbildungswesen der DDR, Kleinstrechner-Tipps, H.8 1988, S 4-11

[K5] Keller, G., Kleinmichel, G.: Bildungscomputer robotron A 5105; Mikroprozessortechnik 2 (1988) 10, S.292-293; dto. in Neue Technik im Büro, 33 (1989) H.2, S. 62-64

[K6] Kerner, I.O.: Computer in der Ausbildung – ECCE 88, GI-Mitteilungen 4 (1989) H.1, S. 25-27

[L1] Löber, C.: BICLAN – Ein lokales Netz für Lehrkabinette, Mikroprozessortechnik, 4 (1990) H.8, S. 30

[NN1] Hard- u. Software für KC 87, Mikroprozessortechnik 3 (1989), S. 57

[NN2] Handreichungen zur Nutzung des Bildungscomputers A 5105, Ausgabe 1989, Hrsg.: Zentralinstitut für Berufsbildung der DDR

[R1] VEB Robotron Meßelektronik Dresden: Heimcomputer robotron Z9001, Prorammierhandbuch, 1985

[R2] VEB Robotron Meßelektronik Dresden: Bildungscomputer robotron A 5105, Programmierhandbuch, 1989

[R2] VEB Robotron Meßelektronik Dresden: Bildungscomputer robotron A 5105, BICLAN-Netz für Lehrkabinette, Bedienungsanleitung, 1990

[R3] VEB Robotron Meßelektronik Dresden: Bildungscomputer robotron A 5105, SCP-Handbuch Teil 1 und 2, 1989

[S1] Schulz, D.: Bildungscomputer robotron A 5105 - Grundlage der Informatikausbildung, Funkamateur 9/1989, S. 428

[SRZ1] 20 Jahre Schülerrechenzentrum Dresden, Chronik des Schülerrechenzentrums 1984-2004, Hrsg.: Rektor der TU Dresden, 2004

[W1] Weise, K-D.: Heimcomputer, Kleincomputer und Bildungscomputer des VEB Kombinat Robotron; Hrsg.: Arbeitsgruppe Rechentechnik bei den Technischen Sammlungen Dresden, 2005

Über ein „Prinzip der strikten simultanen Rekursion“ beim Entwurf von Rechnerarchitekturen und -arithmetiken

Hans-Jürgen Brosch

Lowetscher Str. 2
99089 Erfurt
juergen.brosch@online.de

Abstract: Über eine Mitte 1988 begonnene Studienarbeit zum Entwurf digitaler Verarbeitungsprinzipien der Numerik am Institut für Informatik und Rechentechnik der ADW, über ihre Orientierung und Ergebnisse, den thematischen Bruch durch die Wende und den Abbruch bei der Auflösung des Instituts wird unter heutiger - durch viele spätere Erkenntnisse bereicherter - Sicht berichtet. Erkenntnisse über Zusammenhänge der phänomenologischen und abstrakten Modellierung und über die algebraischen Grundlagen des effizienten maschinellen Rechnens bilden den zweiten Teil des Berichtes. Ihre Verwendung in Theorie und Lehre heute wäre ein kleines Stück dessen, was von der Informatik in der DDR bliebe.

1 Einleitung

Die Arbeitsaufgabe am Institut für Informatik und Rechentechnik hatte zwei Phasen, über drei Phasen in der Entwicklung des Prinzips der strikten simultanen Rekursion wird aber berichtet. Sie hatte anfänglich einen zwiespältigen Charakter, der einerseits auf die Herausarbeitung der inhaltlichen Orientierung der weiteren Architekturforschung am Institut und andererseits auf die Auslotung von Möglichkeiten eines von der Arithmetik ausgehenden Entwurfs von Architekturen orientiert war, nicht nur in der Nachfolge für die DDR Entwicklung des Matrixmoduls - sondern auch in Bezug auf den zukünftigen Entwurf von arithmetisch orientierten eingebetteten Systemen.
Die persönliche Ansicht über die Motivation dieser Forschungsarbeiten am Institut, die weitere Profilierung der Arbeit an Hand der ersten Ergebnisse, die Orientierung auf die strikte simultane Rekursion als ein Wirkprinzip beim Entwurf von arithmetischen Einheiten in Rechnern und die Schilderung der Arbeitsatmosphäre am Institut sind Gegenstand des zweiten Kapitels.
Bei allem Erkenntnisfortschritt blieben nach eineinhalb Jahren zur Zeit der Wende mehr Fragen offen als Antworten gefunden waren. Die Skizzierung der Umorientierung auf Entwurfsmethoden für komplexe Arithmetikeinheiten in der Folge des Wegfalls wesentlicher Motive mit der Wende, der situationsbedingt bescheidene Fortschritt der Arbeit an Entwurfmethoden in den eineinhalb folgenden Jahren und die erfolglosen Bemühungen um die Fortsetzung der Arbeiten im Rahmen von Förderprogrammen oder durch Überleitung zu Unternehmen sind Gegenstand des dritten Kapitels.

Das vierte Kapitel schildert die Ergebnisse des Nachdenkens über die wissenschaftlichen Hintergründe der noch nicht gelösten Probleme in der Arbeit am Institut, insbesondere über Methoden der algebraischen Modellierung in der Informatik, über die axiomatischen Grundlagen effizienten Rechnens und über die Rolle intuitionistischen Denkens in der Informatik während der vielen Jahre des Rentnerdaseins.

2 Arbeitsinhalt und Ergebnis 1988 bis 1991

2.1 Die Motive für die Thematik

Die ab Mitte 1988 bearbeitete Studie war Teil des langfristig unter Einbeziehung von Mitarbeitern aus der Industrie geplanten Aufbaus der Architekturforschung im Institut für Informatik und Rechentechnik. Dieser sollte die vorwiegend zweckgerichteten Architekturarbeiten am Institut ergänzen, meines Erachtens vor allem um

- die aus der wirtschaftlichen Situation und der nahezu totalen Vorbildorientierung in der Industrie resultierende weitgehende Einstellung der Forschungsarbeiten in der Industrie mindestens auf längere Sicht teilweise zu kompensieren,
- wenigstens auf Teilgebieten schöpferische Beiträge in die internationale Forschung einzubringen und dazu beizutragen, ein (weiteres) Absinken des wissenschaftlichen Niveaus zu verhindern
- Chancen für eine Weiterführung der Matrixmodul-Linien im Rahmen des ESER und für eine flexible Reaktion auf Anforderungen der Industrie - insbesondere für den Entwurf eingebettete Systeme - aufrechtzuerhalten
- wenigstens beschränkte Fähigkeiten für eigene Wege im Rahmen gemeinschaftlicher Arbeit des RGW bereitzuhalten. (Wie berechtigt dieses Anliegen war, zeigt heute die dramatisch gewordene Monopolisierung der informationstechnologischen Potenzen).

Damals war nicht abzusehen, dass die Spiele-PC-Ausstattung der Kinderzimmer eine solche gewaltige Auswirkung zeigt und viele Architekturforschungen wertlos machen würde - oder wir haben es nicht erkannt.

2.1 Die Arbeiten bis Mitte 1990

Die Tragfähigkeit eines selbst gewählten bottom-up-Ansatzes für Architekturkonzepte – also ein Rückschließen von der Arithmetik/Logik auf die Gestaltung oder die Ausgestaltung der Rechnerarchitektur war zu untersuchen. Die anfänglichen Arbeiten führten zu dem Schluss, dass

- eine theoretisch untermauerte Systematisierung der Computerarithmetik-Varianten zwingend ist, um eigene Vorschläge abzugrenzen und zu patentieren,
- die bottom-up-Analyse eine gute Methode bei der großen Breite des Gebietes – vom Superrechner über Spezialrechnerarchitekturen bis zu eingebetteten Systemen war und auch für die Synthese, den Entwurf, gute Grundlagen liefert,

- Techniken und Verfahren heranreifen (Computeralgebra, Entwurfs- und Layoutsysteme), die durchgehend berechenbare Entwürfe für dieses Gebiet ermöglichen werden,
- Einflüsse auf das ganze Architekturkonzept entstehen werden, wenn die Fähigkeiten von Zählern, zwischen abstrakten Ebenen unterschiedlicher (technischer) Arbeitsfrequenzen zu transformieren, genutzt werden um z. B. hierarchische Pipeline-Systeme zu konstruieren. Prozesse der Linearalgebra und der Signalverarbeitung bieten beste Voraussetzungen für die Nutzung solcher Effekte.
- große Vorteile aus der sich entwickelnden Technologie dreidimensionaler Schaltungstechnik im Festkörper für das ins Auge gefasste PSSR entstünden.

Als eigenes Ordnungsprinzip für die Vielfalt effizienter komplexer Arithmetiken hat sich bewährt, von der Art der Realisierung arithmetischer Funktionen mit elementaren Operationen auszugehen und diese nach den Graden ihrer Striktheit und Simultanität (also Vielfachheit, Tiefe und Kompliziertheit des Rückgriffs auf vorher Berechnetes) bei der Ausführung rekursiver bzw. iterativer Berechnungen zu unterscheiden. Das Ordnungsprinzip wurde daher als bottom-up-Konstruktionsmethode für effiziente Verarbeitungsverfahren (mit verschiedenen Realisierungsvarianten in Arithmetik und Architektur) gewählt: es sind möglichst strikte und möglichst simultane Rekursionen zu verwirklichen. Dieses Ordnungsprinzips wurde auch Basis der "Prinzip der strikten simultanen Rekursion" genannten Entwurfs-Methode für Arithmetiken zur Bewältigung linearer algebraischer Probleme. Die algebraischen Individuen einer vektoralgebraischen Darstellung werden im Abstieg eines rekursiv gedachten Entwurfsprozesses sukzessiv durch die Substitution paralleler und/oder linear-rekursiver Darstellungen bis zum binären Niveau herab dekomponiert. Es entsteht das abstrakte Modell einer rekursiven Folge mehrdimensional mehr oder weniger hoch strukturierter Elementaroperationen - eine Sequenz von „Strukturoperationen“ oder -verknüpfungen. Wurde dabei eine Hierarchie nichtlinearer Rekursionen (wie bei Zahlendarstellungen) dekomponiert, so bilden die dadurch entstandenen Abhängigkeiten in dieser Modellstruktur räumlich-zeitliche Nachbarschaften. Diese werden entsprechend den Fähigkeiten und Eignungen der elementaren arithmetischen Funktionen zu mehrstelligen atomaren Verknüpfungen zusammengefasst. Die resultierende Struktur der Knoten bestimmt im rekursiven Wiederaufstieg des Entwurfsverfahrens die topologische und algorithmische Struktur des transformierten Verarbeitungsmodells für den jeweiligen Anwendungsfall. Die Restriktionen durch Nichtlinearitäten der Definition von Zahlendarstellungen auf den niederen Ebenen werden dabei auf die höheren übertragen. In diesem Fall realisiert das PSSR eine direkte Abb. der Verknüpfungen einer Algebra hohen Niveaus auf das Niveau der binären Operationen.
Das PSSR ist auch ein Sammelbegriff für insbesondere in der Signalverarbeitung oft angewendete Methoden der Umwandlung einer Rekurrenz von Wertfolgerekursionen in zwei verschiedene oder in eine Abfolge von jeweils zwei unterschiedlichen Phasen. Die jeweils erste Phase ist eine der stellenweisen Akkumulation und die zweite eine Phase der Reproduktion einer Darstellung. Auch Überlegungen über die Einbindung schneller Arithmetiken in Spezialarchitekturen und über "im Fluge konfigurierbare" Arithmetiken, die eine ausgeglichen effiziente Verarbeitung paralleler und sequenzieller Verknüpfungsfolgen leisten könnten, führten zum PSSR. Die auf der Basis hoch paralleler Strukturen von Sequenzial- und/oder Parallelzählern skizzierten Arithmetiken und darauf fußenden Architekturen hatten nach dem damaligen Stand die Chance, in der Flexibilität

sowie in Spezialfällen auch in der Verarbeitungsgeschwindigkeit die aktuellen Pipelinelösungen zu übertreffen. Für universell und variabel auszulegende Arithmetiken wurden strikt simultan rekursiv eingesetzte Parallelzähler als günstig eingeschätzt, besonders bei „logic in memory"-ähnlichen Entwurfsskizzen. Für die Signalverarbeitung mit extrem hohem Datenfluss versprachen serialzählerbasierte Arithmetiken Vorteile, besonders wenn die Frequenzteiler- und/oder Seriell-Parallel-Wandler-Eigenschaften der Zähler genutzt werden.
Mehrere Varianten für Arithmetiken und Architektureinordnungen entstanden als Grobentwürfe, zwei damals zur Diskussion genutzte Skizzen seien als Beispiele angeführt.

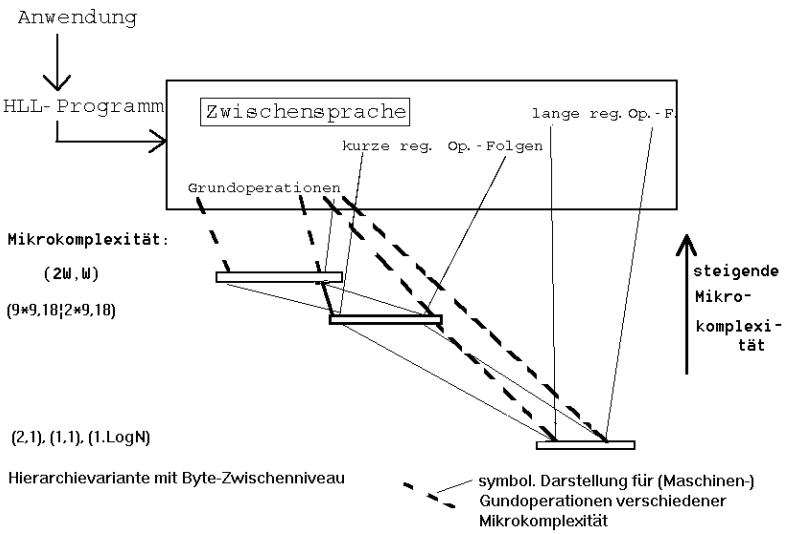


Abbildung 1: Architekturschema-Skizze

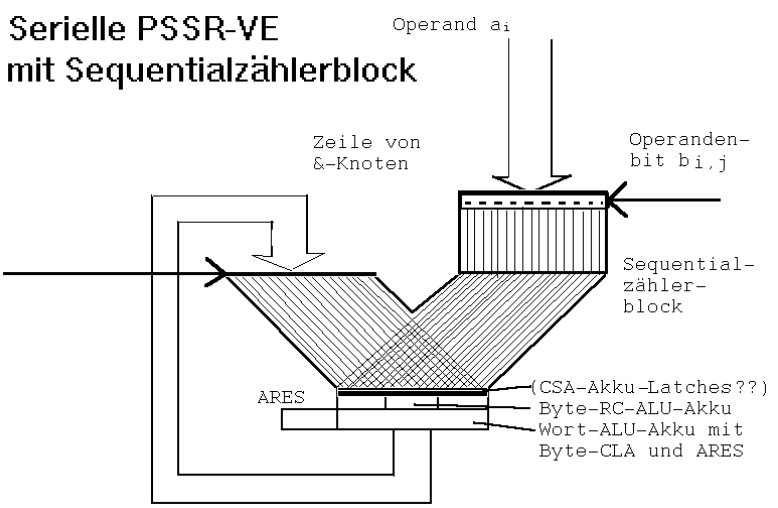


Abbildung 2: Blockschemata einer Serialzähler-Verarbeitungseinheit

Besonderes Augenmerk galt einer Arithmetik mit Kombination von Parallel- und Sequenzverarbeitung. Eine Variante wurde ausgearbeitet und teilweise mittels Assembler und Pascal bezüglich der „verlässlichen“ Funktionsweise simuliert. Die Prüfung dieser und einige anderer Lösungen bezüglich Patentfähigkeit war begonnen, ebenso die Systematisierung der Arithmetik-Prinzipien in einer mehrdimensionalen Tabelle. Mit der Ausarbeitung der Direktabbildungsmethode entwickelte sich die Vorstellung, ein Entwurfskonzept für numerische Komponenten zu entwickeln, das eine vektoralgebraische Beschreibung rechnend direkt auf das binäre Niveau abbildet und die Untersetzung damit also nicht über die „Implementierung in einer anderen Algebra“ erfolgt.

2.2 Die Arbeitsatmosphäre und -bedingungen am Institut

Als zuletzt aus einem Produktionsbetrieb des Kombinates Robotron Kommender fand ich am IIR für Studien-Arbeiten gute „gefühlte“ Verhältnisse vor: eine für unsere Bedingungen ordentlich ausgestattete Bibliothek; gute Unterstützung durch die Leitung; ein ausgezeichnetes junges Kollektiv, das auch einem Älteren aufgeschlossen zuhörte. Ich fand intelligente Streiter, hatte Kollegen, die man nicht nur um Hilfe bitten konnte, sondern die einem auch einen Rat oder eine Literaturstelle unangefordert brachten. In den Bereichen, mit denen ich Kontakt hatte, gab es für den Einzelnen wirklich auch Freiräume für schöpferische Arbeit.
In Manchem waren die materiellen Bedingungen bescheiden – wir hatten keine Wohnungen in Berlin, die Reichsbahn habe ich auch nicht mehr gerne gehabt, der Zugang zu Rechnern für Simulationen war fast unmöglich - und die Beschaffung einer Kopie verursachte einen unverantwortlichen Aufwand. In diesen letzten Jahren der DDR stieß man oft auf die inzwischen erstarrte Bürokratie. Aber die Aufgabe und die Atmosphäre machten manches wett. Der entscheidende Mangel war das völlige Fehlen internationaler Kontakte und die daraus erwachsene Sterilität – man konnte sich nicht mit den anderen auf diesem Gebiet Tätigen wirklich messen und von ihnen nicht im Gespräch die unterschwelligen Informationen bekommen, von denen die Wissenschaft in entscheidendem Maße lebt. Nicht nur die Devisen fehlten, das überzogene Sicherheits- und Schutzkonzept der DDR wurde hier schon oft als Ursache genannt. So war und blieb ein entscheidender Defekt die mangelnde Kenntnis über den wirklichen „Stand der Technik“. Zweifellos wirkte auch die Dominanz militärischer Interessen an Computerarithmetik für Signalverarbeitung dabei mit.

3 Die Arbeiten ab Mitte 1990 bis zur Institutsauflösung

In die in keiner Weise abgeschlossene Arbeitsphase fielen die Ereignisse der Wende und dann der Beschluss zur Auflösung der ADW. Das hat naturgemäß zu einer Desorientierung geführt. Die aus der Situation der DDR und des RGW resultierenden Motive für das Thema lösten sich auf. Die fruchtlosen Bemühungen um die Fortführung im Rahmen der Förderprogramme kosteten Zeit sowie Kraft und die Unsicherheit bremste zusätzlich den inhaltlichen Fortschritt der Arbeit in der Zeit bis zur Auflösung des Instituts.
Die Weiterentwicklung des PSSR in dieser Zeit bestand u. a. in der verfahrensmäßigen Einbeziehung des Zugriffs auf als Nachbarn in Shiftketten vorliegende aktuelle Bitlevel-

Operanden im Sinne eines Rückgriffs auf Vorgänger, womit man beispielsweise die Matrizenmultiplikation oder auch ein einzelnes Skalarprodukt nach den bekannten Skewing-Algorithmen der linearen Algebra berechnen kann (dekomponiert auf das Ziffern- oder Bitniveau). Das ist eine für die Bewältigung der hochfrequenten Datenzubringung ganz wichtige Erweiterung des Spektrums von PPSR-Architekturmerkmalen. Mit diesem Postulat werden solche Algorithmen und die systolischen Arrays Beispiele für die Gültigkeit des PSSR als relativ allgemeines Prinzip.
Da die Forschungsorgane der BRD kein Interesse für die architekturorientierten Arbeiten zeigten, wurde der Inhalt der Arbeit systematisch auf die Vertiefung der Grundlagen für eine Entwurfsmethode ausgerichtet. Fernziel war die Schaffung von Entwurfssystemen mit „Anschluss“ an die Computeralgebra einerseits und an Layoutsysteme andererseits. Ein Entwurfssystem erforderte eine größere lückenlose Einbeziehung der arithmetischen Möglichkeiten in das theoretische Konzept. Unzulänglichkeiten der damaligen Ausgestaltung der Transformation – sie erfasste nur die akkumulierenden Phasen im mathematischen Kalkül – waren unbefriedigend und wurden hinderlich. Noch nicht abgeschlossen war die Einordnung arithmetischer Konzepte wie Carry-Skip- und Carry-Select-Adder, Online-Arithmetik u.s.w. Besonders unbefriedigend war, dass die als besonders leistungsfähig bekannten Verfahren der schnellen Fouriertransformation, des Karatsuba-Baumes und andere nicht Bestandteil des PSSR-Konzeptes waren.
Zur Beantwortung dieser und anderer Fragen kam es nicht mehr in der Arbeit am IIR der ADW.
Ich persönlich hatte den Eindruck, dass die Evaluierungskommission, in der ja fast nur Konkurrenten um die knappen Fördermittel mitwirkten, von vornherein von der Abwicklung des Instituts überzeugt war. Die Berliner Industrie war an den gut ausgebildeten jungen Mitarbeitern interessiert. Zu den Gutachtern für Förderprogramme habe ich mein Vorhaben wohl nicht gut genug „herübergebracht“ - und ein Sozialfall war ich als möglicher Alterübergangsgeldbezieher auch nicht. Das WIP kam nicht in Frage, die Universitäten des Ostens entließen damals Mitarbeiter, die viel jünger waren. Für die um ihre Neuprofilierung und um das Überleben ringenden industriellen Einrichtungen war der Stand der Arbeiten nicht reif genug. Alle Versuche schlugen fehl.
Das Vorhaben „PSSR“ reihte sich ein in die Kette nicht vollendeter Forschungsvorhaben.

4 „Informatik nach der DDR“ als Hobby

4.1 Offen gebliebene Fragen

Irgendwann in den Jahren danach lebte das PSSR wieder auf – als Hobby. Es trieb die Neugier auf Antworten auf die offen gebliebenen theoretischen Kernfragen:

- worin bestehen bei kanonisch formulierten Modellen überhaupt die Möglichkeiten, effizienter zu rechnen als es der durch das termalgebraische Modell vorgegebene Algorithmus ermöglicht?
- lässt sich der intuitionistische Ansatz "Ausklammerung der Stellenwert-Struktur" zu einem fundierten, durchgängig berechenbaren Modell ausbauen?

- wo liegt die Ursache dafür, dass die Polynomdarstellung der Zahlen und die Exponentialform der komplexen Zahlen eine so fundamentale Rolle in effizienten Berechnungsverfahren spielen und welchen Zusammenhang gibt es zwischen diesen und der dem PSSR zu Grunde liegenden These vom strikt simultanen Rückgriff auf Vorberechnetes?
- liegt gar einerseits in der ideellen Verbindung der phänomenologischen Modelle statischer wie dynamischer Systeme mit ihren Relationen zwischen Eigenschaften und andererseits den axiomatischen Grundlagen unserer mentalen, formalen Modelle der Schlüssel für das Verständnis der Zusammenhänge zwischen den mathematischen Konzepten und den unterschiedlichen Methoden, effizientes Rechnen zu erreichen?

Für den Autor und das PSSR scheinen die Antworten gefunden zu sein. Sie werden - aus Platzgründen nur thesenhaft - in den Abschnitten 4.2, 4.3 und 4.4 wiedergegeben. Möglicherweise ist das den Theoretikern auch in diesen Zusammenhängen als Trivialität gegenwärtig. Es ist ja nur eine andere Sicht auf Bekanntes. In den durchgesehenen Lehrbüchern fand ich es so nicht. Das hier Darzustellende würde sicher auch Anderen den Zugang zu Zusammenhängen zwischen den mathematischen Konzepten in der Informatik erleichtern und vor allem für die Ausbildung nützlich sein.

4.2 Grundlagen des effektiven und effizienten Rechnens.

Die mit der Entwicklung von Arithmetiken und Algorithmen gestellte Frage ist die nach einer möglichst effizienten Reduktion des hoch redundanten Anteils eines als termalgebraische Relation formulierten Modells. Ein einfaches Beispiel ist die wichtigste komplexe Operation der linearen Algebra - die Auswertung eines Skalarprodukts über die Relation zwischen dem Zahlenwert des Ergebnisses einerseits und der als nicht reduzierbar angesehenen Summe von Produkten andererseits - für die ein effizienter Weg gefunden werden soll. Die Relation wird als erfüllt angesehen, während alle Variablen in der Darstellung zum Beispiel nicht belegt sind, was in diesem Falle die Nichtreduzierbarkeit ausmacht. Als zweites Beispiel sei das Produkt zweier Zahlen in polynomialer Darstellung angeführt, dessen Ergebnis ebenfalls bekannt sei. Die Polynomdarstellung dient dazu, mit Zahlen aus großen Zahlenbereichen überhaupt effektiv rechnen zu können. Sind die Darstellungen gewöhnliche Polynome, dann sorgt die Unbekannte x für die Nichtreduzierbarkeit der Terme, sind es Stellenwert-Darstellungen so ist es die nichtlineare Definition der Darstellung über der begrenzten Menge der Ziffern zur Basis B.
Bei der Auswertung eines solchen kanonischen Modells kann eine höhere Effizienz nur auf zweierlei Weise erreicht werden:

- durch Ausnutzung von Eigenschaften der ersten Verknüpfungen (z. B. Addition) einer Algebra, nachdem die höher priorisierten Operationen symbolisch „realisiert“ sind,
- durch Ausnutzung der Eigenschaften von Verknüpfungsebenen, die nach Substitution von Darstellungen für die Individuen der Algebra, also nach Transformation auf eine andere Algebra, entstehen.

Zu der ersten Kategorie gehören die DFT, die FFT und auch der Karatsuba-Algorithmus, zu der zweiten diejenigen arithmetischen Konzepte, die in der ursprünglichen Direktabbildung des PSSR als strikt simultane Rekursion auf elementarer Ebene behandelt wurden. Ein verallgemeinertes Modell muss beides erfassen. Bei einem PSSR-Modell wird vorausgesetzt, dass eine algebraische berechnende Transformation zwischen den algebraischen Ebenen möglich ist. Auch im Weiteren wird immer von der PSSR-Modellierung gesprochen, weil die Ausführungen keinen Anspruch darauf erheben, von größerer Allgemeingültigkeit zu sein.

4.3 Der top-down-Prozess der PSSR-Modellierung.

Die Mängel des älteren Modellierungskonzeptes veranlassten zu einer besser fundamentierten Analyse des top-down- und des bottom-up-Vorgehens bei der Modellierung. Das top-down ist als eine Verallgemeinerung und Algebraisierung des Gedankens der "Ausklammerung der Stellenwertstruktur" und das bottom-up als eine solche für die auf Bitniveau formulierte Strategie des Rückgriffs auf Vorberechnetes zu sehen. Man kann es auch als einen großen rekursiven Prozess betrachten, bei dem im rekursiven Abwärtsgehen die algebraischen Strukturen hohen Niveaus durch das Einsetzen ihrer Darstellungen in Elementen einer niederen Algebra schrittweise auf die Ebenen der Realisierung dekomponiert werden. Danach werden im weiteren Abwärtsschreiten für diese Ebenen die realisierenden Operationen auf die elementaren Grundlagen von Berechnungstheorien zurückgeführt. Im Wiederaufstieg sind die Bedingungen für den optimalen effizienten Einsatz der realisierenden Operationen zu ermitteln und effiziente Verfahren für die jeweiligen Ebenen zu suchen, wobei gleichzeitig die Restriktionen der niederen Ebenen, die aus beschränkten Mengen und nichtlinearen Definitionen entstehen, das auf höchste abstrakte Niveau des Modells abgebildet werden.
Aus dem phänomenologischen Modell der Anwendung wird auf derjenigen Ebene, auf der eine Transformation erfolgen soll, an das Konzept der Informatik "Ergebnis ist Operationen über Daten" anknüpfend ein formales Modell entwickelt, dessen Typ

*Ergebnisstruktur = Operatorenstruktur*Operandenstruktur*

sein soll. Dahinein fließt als allgemeine Erfahrung zur Modellierung, dass

- in phänomenologischen Modellen die Semantik der Eigenschaften den Charakter der Verknüpfungen bestimmt (Addition bei gleicher Semantik, Multiplikation bei verschiedener),
- die Abstraktion von Semantik „im Hintergrund wirkt“ und die Semantik bzw. der Kontext sich in Namen von Variablen und in der Definitionen zulässiger Verknüpfungen versteckt,
- die verschiedenen Typen von Vektoralgebren aus der semantischen Separation bzw. formalen Faktorisierung multiplikativer Verknüpfungen von nichtreduzierbaren Termen der Termalgebra entstehen und nur der abstrakte Vektorraum als solcher eine Abstraktion von jedem Bezug darstellt – es also für die Modellierung wichtige Relationen zwischen Term- und Vektoralgebren gibt,
- die semantikorientierte Separation als Verallgemeinerung der formalen Faktorisierung die geeignete Grundlage für das obige Operatoren-Operanden-Relationen-Modell darstellt. Auch Polynome werden als semantisch separierbar

angesehen und in dieser Weise in Strukturen von Koeffizienten und Potenzen dekomponiert,
- mit dem Ziel der Allanwendbarkeit statische wie dynamische Systeme mit dem obigen Operatorenmodell beschrieben werden können und müssen,
- in der Verknüpfung von Operatoren- und Operandenstruktur stets beide Strukturen simultan manipuliert werden müssen. Die Schritte der Manipulation dienen dem Ziel, die redundante, durch die Verknüpfung gekennzeichnete Seite der Relation der nicht redundanten Ergebnisstruktur der anderen Seite schrittweise anzugleichen.

Zu dem letztgenannten Vorgehen, das ein Schritt in Richtung der Algebraisierung der Ausklammerung von Stellenwertstrukturen darstellt, sei als Beispiel die Summation von Bitvektoren bei der Berechnung der Summe von Zahlen mit Hilfe von Tabellen erläutert. Die Operandenstruktur vor der Reduktion ist in der linken Tabelle gegeben. Auf der rechten Seite wird die Operandenstruktur nach der Reduktion gezeigt. Die Stellenwertstrukturen sind durch die Randzeilen und –spalten angedeutet. Dieses einfache Beispiel muss hier genügen. Später wird im Zusammenhang mit der Anwendung der Fouriertransformation beim Polynomprodukt ein noch instruktiveres Beispiel genannt.



4.4 Der bottom-up-Prozess der PSSR-Modellierung.

Dass termalgebraische Modellrelationen nur innerhalb von Verknüpfungen zwischen nichtreduzierbaren Termen reduziert werden können, wirft die Frage nach den Eigenschaften der Algebren auf, in denen diese Terme beschrieben sind. Die berechnungstheoretischen Grundlagen seien betrachtet.
Die Berechenbarkeitstheorie liefert mit den Konzepten der primitiven Rekursion und der μ-Rekursion die Basis. Aus ihr fließen Folgerungen über den Aufbau der algebraischen Zahlentypen und über die Hierarchie der Verknüpfungen (Addition, Multiplikation, Potenzierung). So entstehen durch die induktive Anwendung der einstelligen Operation der primitiven Rekursion jeweils über einer Konstanten die arithmetischen Grundfunktionen. Diese Grundfunktionen
- bilden die Voraussetzung für das Rechnen in großen Zahlenbereichen und gewährleisten mittels ihrer Rolle in Zahlendarstellungen das effektive Rechnen, weil sie unmittelbar zwischen zwei linear georderten Mengen abbilden, die in erster Ordnung rekursiv definiert sind.

- besitzen Konstanz-Eigenschaften (z. B. Krümmung), welche die Grundlage ihrer "arithmetischen Theoreme" und des effektiven Rechnens bilden.
- können als direkt in erster Ordnung rekursiv definierte Funktionen nur monotone diskrete Folgen von Funktionswerten generieren.
- Definieren keine Stellenwert- und Modul-Darstellungen von Zahlen, weil diese ihrer Generierung entsprechend zu den μ-rekursiv definierten berechenbaren Funktionen gehören.

Als auf die Berechenbarkeitstheorie aufbauende Grundlage für die effiziente Berechenbarkeit in Modellen wird die Möglichkeit angesehen,

1. das polynomiale Kodieren als eine auf der Erfahrung des modularen Zählens aufbauende Grundfunktionen zu akzeptieren,
2. nichtmonotone Grundfunktionen durch axiomatisch fundierte simultane Induktion zu generieren.

Zu 1. Auf der binären Ebene kommen die Zähler als realisierende elementare Funktionen zum Einsatz. Ihre Funktionalität ist die binäre Codierung, die auch rekursiv genutzt wird. Im rekursiven Einsatz effizient sind nur solche Anwendungen von Zählern, die eine vollständige Rekursivität gewährleisten, indem die Stelligkeit der Verknüpfung durch die Addition und die Rekursion systematisch voll genutzt wird. Dafür sind die 7-Bit- und die 3-Bit-Zähler allein oder als Kette günstig, weil sie redundanzfrei sowohl im asymmetrischen wie auch im symmetrischen Baum (letzteres als Kette zur Reduktion von vier Eingängen auf zwei Ausgänge) eingesetzt werden können. Sie sind im Sinne des PSSR effizient.
Zu 2. Für die effiziente formale Modellierung dynamischer Phänomene sind effizient berechenbare nichtmonotone Funktionen Voraussetzung. Wenn die monotonen Grundfunktionen mit ihren Konstanzeigenschaften das effektive Rechnen bewirken, weil sie direkt induktiv definiert sind, muss erwartet werden, dass für effizientes Rechnen mit dynamischen Modellen induktiv generierte nichtmonotone Grundfunktionen von Bedeutung sind. Sie müssen wie die monotonen Grundfunktionen Konstanzeigenschaften besitzen, welche zu arithmetischen Berechnungstheoremen führen.
Nichtmonotone Funktionen durch Induktion zu gewinnen erfordert die Akzeptanz der Erfahrung von Wirkung und Gegenwirkung in der Realität. Ein Generierungsaxiom („nur aus Axiomen und bereits Definierten kann Neues gewonnen werden“) erlaubt die Generierung in Abhängigkeit von allen Vorgängern. Damit die Induktion über den natürlichen Zahlen wegen einer wachsenden Zahl von Abhängigkeiten nicht „explodiert“, kann nur eine solche Abhängigkeit als Modell dienen, bei der diese Zahl klein ist und konstant bleibt. Es wird deshalb von vornherein ein strikt simultaner Ansatz für die Rekursion gemacht und ggf. dann deren Rückgriff auf alle Vorgänger bestimmt.
In der Algebra der Arithmetik kann aus Prioritätsgründen der „Zugriff“ auf Vorgänger nur über multiplikative und die Verknüpfung der Zugriffe nur durch additive Operationen erfolgen. Statt einer multiplikativen Iteration über einer Konstanten x wie bei der Generierung einer Folge von Potenzwerten ist unter Annahme der Gegenwirkung des zweiten Vorgängers eine Induktion über einer Summe mit einer Summe von Konstanten als Faktor auszuführen.

Gefordert ist demnach die Zuweisung $x_1 = x_1 \cdot y_1 - x_2 \cdot y_2$

.

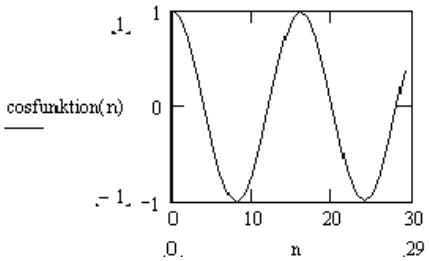
Um dieses durch arithmetische multiplikative Induktion zu bewerkstelligen, ist die Einführung der komplexen Zahl das Mittel der Mathematik. Die Iteration ist dann

$$x \leftarrow x \cdot y \quad mit \quad x = (x_1 + i \cdot x_2) \quad und \quad y = (y_1 + i \cdot y_2) .$$

Das geforderte Ergebnis ist der Realteil, der aus der Induktion entspringt. Das ist in einer Routine einfach zu simulieren. Mit der Iterationskonstanten y_1 wird auf den unmittelbaren Vorgänger als Verknüpfungspartner zugegriffen, mit der Konstanten y_2 auf den zweiten Vorgänger. Anfangswert für die Iteration über dem zu berechnenden Wert x_1 ist 1, für den „Vorgänger" x_1 ist der Anfangswert natürlich gleich 0.

Die Forderung nach Konstanzeigenschaften für nicht monotone Grundfunktionen führt auf das Modell harmonischer Funktionen als Projektion des Kreises und erlaubt es, über den Pythagoras die Generierungskonstanten so zu bestimmen, dass mit einer Routine mit dem nachfolgend angegebenen Schleifenkern die Kosinusfunktion generiert wird.

```
for k ∈ 0..(i − 1)
    xNeu1 ← x1·y1 − x2·y2
    x2 ← x2·y1 + x1·y2
    x1 ← xNeu1
```



Die imaginäre Zahl i kann als "Bypass-Quelle-" und „Bypass-Senke-Operator" oder als Merkmal einer nicht in das aktuelle Ergebnis eingerechneten Hilfsvariablen (Speichervariablen) interpretiert werden. Die multiplikative Verknüpfung zweier komplexer Zahlen symbolisiert also eine aktuelle Verknüpfung, die von der Vorgeschichte abhängig ist. Abstrahiert man von der Generierung und von der Rolle des negativ eingehenden Produkts als Rückgriff auf den zweiten Vorgänger, dann muss man diese Verknüpfung (von Realteilen) als kontextabhängige Verknüpfungen interpretieren. (Man denke an den Kontext im schwingenden System.)

Definiert man in gewohnter Weise die über der Menge der reellen Zahlen generierten komplexen Zahlen als Elemente der neuen Menge C, dann hat man einen direkten diskreten Weg zu dieser Menge. Implizit erzeugt die Routine als Imaginärteil die Sinusfunktion.

Damit ist obige Routine mit Konstanten, die sich aus der rekursiven Zweiteilung des Vollkreises ergeben, auch die Grundlage für die direkte induktive Generierung der diskreten komplexen Exponentialfunktion. Die Multiplikation ist die erste Verknüpfung in einer darauf basierenden Algebra. Daher kommt sicher ihre hohe Effizienz im Modell der "komplexen Wechselstromrechnung".

Als fortgesetzte Drehung eines gegebenen Zeigers ist die Konstanten-Iteration eigentlich überall bekannt und hat in der Moivreschen Formel Niederschlag gefunden. Diese Induktion als strikt simultan rekursive Generierung zu betrachten, hat aber doch grundlegende Bedeutung. Die Erweiterung durch Einbeziehung der dritten, dann der vierten Vorgänger mittels verknüpfter imaginärer Zahlen, die auch wieder imaginäre Zahlen erzeugen, entstehen die hyperkomplexen Zahlen. Der „Bypass über zwischenliegende

Vorgänger“ erklärt auch (durch die erforderlichen Relationen bei der Generierung neuer imaginärer Einheiten, z. B. für das Quaternion als i*j=k), warum es in der Reihenfolge (Komplexe Zahl, Quaternion, Oktion,...) jeweils (1, 3, 7,...) imaginäre Zahlen und eine reelle Zahl gibt. Es entsteht die Vermutung, dass die Effizienz des Einsatzes von Quaternionen z. B. in der Computergrafik auf diese Definition durch Generierung zurückzuführen ist. Interessant ist die Tatsache, dass die redundanzfreie Verkettung von Zahlen auch mit der Folge (1, 3, 7) zusammenhängt.
Die komplexe Exponentialfunktion als strikt simultane Grundfunktion besitzt zwei Konstanten, den arithmetischen Mittelwert und den quadratischen Mittelwert - jeweils über beliebig viele ganze Perioden. Das Multiplikationstheorem ermöglicht es, durch die Multiplikation zweier gegenläufiger Drehungen die Exponentialformdarstellung der Deltafunktion zu konstruieren, die sich als Mittel für eine Transformation aus der Abhängigkeit von allen Vorgängern zu der Abhängigkeit von allen Komponenten als direkte Vorgänger erweist.

Das wird in einem Beispiel für die PSSR-Modellierung bei der wichtigen Anwendung „Produkt zweier Polynome n-ten Grades“ benutzt. Die diskrete Deltafunktion dient als "Transformationsgenerator" des semantischen separierten Produkts. Das Ergebnispolynom in der Modellrelation ist in diesem einfachen Fall ein Polynom (2n-1)-ten Grades. Seine Koeffizienten sind die so genannten Faltungssummen. Für die Transformation von Operatoren- und Operandenstruktur werden die Deltafunktionen den Partialprodukten in den beiden Strukturen als Faktoren beigefügt.

$$\Delta(j,k,l) = \frac{1}{N}\cdot\left[\sum_{n}\left[a_j\cdot b_k\cdot e^{\frac{2\pi i(j+k-l)n}{N}}\right]\right]$$

Die PSSR-Modellierungsmethode führt mit dieser Hilfe direkt auf die diskrete Fouriertransformation. Das lässt den Schluss zu, dass die Fouriertransformation ebenfalls eine Übertragung der rekursiven Abhängigkeit von allen Vorgänger-Koeffizienten auf eine Abhängigkeit von allen transformierten Elementen in direkter, simultaner Vorgängerschaft erster Ordnung ist.
Schließlich ist die Transformation der DFT in die FFT als rekursive Faktorisierung der Transformationsmatrix eine Transformation auf eine simultane Rekursion erster Ordnung, die allerdings ein kompliziertes Verbindungsschema hat. Im Grunde genommen ist die FFT die Vernetzung von binären Bäumen und somit ein optimaler Rückgriff auf Vorberechnetes (und auf algebraisches Wissen).

Diese Ausführungen sollten andeuten, dass das Prinzip der strikten simultanen Rekursion viel weiter reicht als es der Autor ursprünglich angenommen hat. Es reicht bis in die axiomatischen Grundlagen hinein - und es lohnt sich, über die Zusammenhänge in der Modellierung diskreter Systeme von einem intuitionistischen Standpunkt aus nachzudenken. Das hilft dabei, diese wichtigen Zusammenhänge zu verstehen und sicher auch dabei, sie schöpferisch anzuwenden.

Sollte sich das Dargelegte in Diskussionen mit den Fachkollegen bewähren, so wäre es aus der egoistischen Sicht des Autors etwas Wichtiges, was aus einer kurzen Episode der Arbeit an der "Informatik in der DDR" bleiben würde.

Filmische Retrospektive - Rechentechnik der 1970-er Jahre

Klaus-Dieter Weise

Förderverein für die Technischen Sammlungen der Stadt Dresden e. V.
Junghansstraße 1-3
01277 Dresden
kw10@mail.inf.tu-dresden.de

Abstract: Bei der Einführung der Rechentechnik in der DDR wurden beginnend in den 1960-er Jahren und später verstärkt audiovisuelle Medien nicht nur in Rundfunk, Film und Fernsehen sondern auch zur Ausbildung und Qualifizierung von Fachpersonal eingesetzt. Die mit Unterstützung zahlreicher Autoren spezifischer Fachkompetenz produzierten audiovisuellen Medien sind spezielle Zeitdokumente, die über propagandistische Ansprüche hinaus zu sachlichen Inhalten, Methoden, Wertungen, Erfolgen und Grenzen in einer Periode der Einführung neuer Technik Auskunft geben. Zwei Film-Beispiele aus den 1970-er Jahren sollen dies demonstrieren.

Das Auffinden der heutigen Standorte und Verwertungsrechte dieser Art Zeitdokumente ist keine einfache Aufgabe und im Besonderen zum Thema Informatik/Rechentechnik bisher nicht bearbeitet. Basis weiterer Recherchen können jedoch Leistungen sein, die z. B. durch die DEFA-Stiftung, das Deutsche Institut für Animationsfilm in Dresden, den Filmverband Sachsen und das Institut für Bildung und Medien der Gesellschaft für Pädagogik geboten werden. Erste Ansätze mit Mediendokumentationen sind vorhanden.

Zugegeben, eine filmische Retrospektive ist ein etwas außergewöhnlicher Beitrag zum Informatiksymposium. Sie soll aber auf ein spezielles Gebiet bei der Vermittlung wissenschaftlicher bzw. populärwissenschaftlicher Kenntnisse im Zeitraum der Einführung innovativer Techniken bzw. Technologien in der DDR, im weitesten Sinne eben auch der Informatik, hinweisen. Wie bei den meisten Inhalten der verschiedenen Gattungen medialer Ausdrucksformen in Wort und Bild, standen dabei immer gesellschaftspolitische, im Einzelnen bildungspolitische und wirtschaftspolitische Zielstellungen mit zeitlich unterschiedlichen Schwerpunktthemen und Zielgruppen im Spannungsfeld der Propagierung einer materialistischen, marxistischen Weltsicht und künstlerischer Ansprüche sowie sachbezogener Szenarien und Gestaltungsmittel.

Über das weite Feld der Definition, Kategorisierung, Analyse oder historischen Bedeutung der entstandenen medialen Produkte als Zeitdokumente im Zusammenhang mit der Einführung der Informatik und elektronischen Datenverarbeitung in der DDR, gibt es nach meinen Recherchen noch keinen zusammenfassenden Überblick. Dieses Feld ist teilweise noch unaufbereitet und bleibt damit Arbeit für Historiker.

Trotzdem, fragt man nach einer Zeit der Wirren in den 1990-er Jahren heute nach dem Verbleib audiovisueller Medien, d. h. filmischer oder nicht-filmischer Medien, wie

- Diaserien, Folienreihen, Tonbildreihen, Tonträger und Begleitheften
- Lehr- und Unterrichtsfilmen,
- populärwissenschaftlichen Filmen
- Industrie-, Produkt-, Informations-, Image- und Werbefilmen,

so wird man bei mühsamer Kleinarbeit durchaus fündig, auch zu den Themen Informatik/Rechentechnik. In die Recherche sind dabei unbedingt die Gattung bzw. Sendeform der periodischen Magazinreihen oder Einzelbeiträge des DDR-Fernsehens und -Rundfunks einzubeziehen, die insbesondere durch das AHA-Magazin (1977-1990) des Bildungsfernsehens, die Fernseh- bzw. Neue Fernseh-URANIA (1956/74-1990) als Wissenschaftsmagazin, die Umschau - Aus Wissenschaft und Technik (1961-1991) [KLR98] oder die Sendungen des DDR-Rundfunks (ab 1987) im Rahmen des Schulfunks wie REM - Das Computermagazin oder DT 64 mit dem Computerclub und schließlich Tonträgerprodukte repräsentiert werden.

Ungezählt sind noch die o. g. audiovisuellen Medien und Titel. Mit den staatlichen Beschlüssen über die Entwicklung, Einführung und Durchsetzung der elektronischen Datenverarbeitung in der DDR (1964) begann auf diesem Gebiet eine forcierte Herstellung dieser Medien zum Zwecke der Vermittlung allgemeinen Wissens, zur Unterstützung der Ausbildung in den neuen Berufen, der allgemeinbildenden und universitären Ausbildung sowie der Weiterbildung. Der Hauptproduzent filmischer Medien o. g. Art, die DEFA, startete z. B. 1964 in ihrem Studio für populärwissenschaftliche Filme eine Produktionslinie Industriefilm, d. s. „ …Filme, die wissenschaftlich-technische Informationen und Dokumentationen und den hohen Leistungsstandard der Volkswirtschaft der DDR im In- und Ausland propagieren (…) und die außenpolitischen Aufgaben der Industriezweige unterstützen.“ [Kn96].

Erinnert sei z. B. an den vielleicht ersten Film zum Thema aus dem Jahre 1964, dem Dokumentarfilm „Moderne Hexenmeister“ vom Regisseur Heinz Müller des DEFA-Studios für Wochenschau und Dokumentarfilme, der über die Anfänge der Computertechnik und die Entwicklungen aus dem Unternehmen Carl Zeiss Jena und der Technischen Hochschule Dresden berichtet [FS06], des weiteren an eine Tonband- und DIA-Reihe zur EDV aus den Jahren 1967/68, hergestellt im Auftrage des damaligen Instituts für Datenverarbeitung in Dresden, die mehr als 700-mal Absatz fand. Oder an einen speziellen Begleitfilm für das Politbüro der SED zur Begründung der Bildung des Kombinates Robotron. Besondere Bedeutung hatte auch die Produktion des DEFA-Studios für populärwissenschaftliche Filme aus den Jahren 1968-1971 über die Elektronische Datenverarbeitung mit 65 Lektionen und Begleitbüchern mit einer Auflage von mehr als

365 000/Lektion, gezeigt im Rahmen der Fernsehakademie des DDR-Fernsehens. 135 000 Personen nahmen an einer schriftlichen Prüfung der Fernsehakademie teil.

Schließlich der erste Industrie- und Werbefilm über die in der DDR zur Legende gewordene EDVA R 300, produziert 1966 vom Regisseur Heinz Fischer im DEFA-Studio für Dokumentarfilme und im Auftrage der VVB Datenverarbeitungs- und Büromaschinen, gezeigt auf Messen und der Ausstellung Interorgtechnika in Moskau.

Zwei Film-Beiträge aus den 1970-er Jahren sollen zeigen, welche Inhalte mit welchen handwerklich-filmischen Mitteln transportiert wurden. Es handelt sich um die Industrie- und Werbefilme von Manfred Schreyer:

Elektronisches Datenverarbeitungssystem Robotron 21

hergestellt im DEFA-Studio für Trickfilme Dresden, Produktionsjahr 1971
Drehbuch und Regie: Manfred Schreyer
Manuskript: Egon Hofmann (Robotron), Manfred Schreyer
Fachliche Beratung: Jürgen Arnold, Frank Paulusch (beide Robotron)

und

Probleme-Programme-Lösungen Robotron 4000

hergestellt im DEFA-Studio für Trickfilme Dresden, Produktionsjahr 1979
Drehbuch und Regie: Manfred Schreyer
Manuskript: Egon Hofmann (Robotron)
Fachliche Beratung: Lothar Gräßler (Robotron)

Dank gilt dem Deutschen Institut für Animationsfilm e. V. in Dresden, das Bestände des ehemaligen DEFA-Trickfilmstudios bewahrt, welches auch von o. g. Filmen Kopien auf DVD herstellen ließ und die Aufführungsrechte vergab.

In nicht unerheblichem Umfange trugen zum Inhalt der produzierten Medien auch Fachspezialisten der jeweiligen Auftraggeber durch Beteiligung als Berater, Autoren, Coautoren oder Szenaristen bei, um vor allem Unterstützung bei der Herausarbeitung der sachlichen Zusammenhänge und Wirkungen zu geben.

Wie steht es heute mit dem Zugang zu den von verschiedenen Auftraggebern in Auftrag gegebenen und von ebenso verschiedenen Produzenten hergestellten audiovisuellen Medien zum Thema Informatik/Rechentechnik? Nach einigen Wirren der 1990-er Jahre, bei denen zahlreiche Bestände bei den Nutzern dieser Medien vernichtet wurden, ist diese Frage im Grundsatz heute entschieden und positiv zu beantworten.

Die Bestände, vor allem die filmischen Medien, sind durch entsprechende Übernahmen aus alten Archiven oder auch durch Zufallsfunde weitgehend gesichert. Über die wichtigsten Zugänge wird im Folgenden ein Überblick verschafft. Neben dem Auffinden der jeweiligen Standorte kommt dem Besitz der Verwertungs- bzw. Aufführungsrechte (bisher das Problem schlechthin) eine besondere Bedeutung zu. Ohne die Zustimmung des

aktuellen Rechteinhabers eines audiovisuellen Mediums aus Beständen ehemaliger DDR-Betriebe und Einrichtungen ist nach heute, wie auch damals geltendem, kompliziertem Urheberrecht und heutiger Rechtssprechung keine öffentliche Nutzung, also auch Aufführung (des Originals oder einer Kopie) möglich. Und meist auch nicht ohne entsprechende Löhnung. Es bleibt die mühsame Suche nach den Standorten, Rechteinhabern und Kopie- und Verleihfirmen. Hilfestellungen bei der Suche leisten dabei auch mehr oder weniger öffentlich zugängige Datenbanken. Sicher sind noch einige Bestände bei ehemaligen Nutzern, d. h. in Privathand, Bibliotheken, Mediatheken, Museen oder Betriebsarchiven auf dem ehemaligen DDR-Gebiet vorhanden, sie sind einem systematischen Zugang aber nicht unbedingt offen und daher Ansatzpunkt zur weiteren Fahndung nach diesen Beständen.

Über den Zugang und/oder die Rechteverwertung zu audiovisuellen Medien der in der ehemaligen DDR produzierten Titel zum Thema des Symposiums möchte ich den folgenden, jedoch keinesfalls vollständigen, Überblick und gleichzeitig Anregung geben, dieses Thema weiter zu vertiefen. Verwendet wurden hierfür teilweise Auszüge aus den jeweiligen Webseiten der Institutionen.

Bundesarchiv-Filmarchiv

Das Bundesarchiv-Filmarchiv verfügt über den vollständigsten Bestand an DEFA-Filmen. Soweit Benutzungsstücke vorliegen, steht das Material zur internen Nutzung für wissenschaftliche Zwecke und zur Ansicht zur Verfügung.

DEFA-Filme und filmbegleitende Materialien in Auswahl befinden sich außerdem im Bundesarchiv, Abt. Reich und DDR und Abt. Massenorganisationen der DDR. In diesen Abteilungen ist Schriftgut aus den DEFA-Studios und dem Ministerium für Kultur sowie Drehbücher zu Spielfilmen archiviert. Im Archiv der Parteien und Massenorganisationen der DDR (SAMPO) findet sich Schriftgut zum Filmwesen in der DDR im Aktenbestand des Zentralkomitees der SED.

Anträge auf Benutzung müssen schriftlich gestellt werden. Ein Verleih oder Kopieren audiovisuellen Materials ist nicht möglich. Gewerbliche und nichtgewerbliche Nutzungen sind nur mit Zustimmung des Rechteinhabers möglich. Es wird auf die Rechte verwertenden Organisationen/Unternehmen verwiesen.

Kontakt:
Bundesarchiv-Filmarchiv
Fehrbelliner Platz 3, 10707 Berlin
Telefon: 01888/777 0 - 0
www.bundesarchiv.de

DEFA-Stiftung

Die DEFA-Stiftung wurde 1998 errichtet. Als Stiftungsvermögen wurde ihr der DEFA-Filmstock übertragen. Der DEFA-Filmstock umfasst die gesamte Kinoproduktion der DDR-Filmstudios aus fast fünf Jahrzehnten.

Zum DEFA-Filmstock gehören rund 12 000 Filme:

950 Spielfilme und Kurzspielfilme,
820 Animationsfilme,
5.800 Dokumentarfilme und Wochenschauen,
4.000 deutschsprachige Synchronisationen ausländischer Filme,
nicht veröffentlichte und Restmaterialien aus der DEFA-Produktion,
Fotos, Plakate, Drehbücher und ihre literarischen Vorstufen, Werbematerialien,
Partituren und Zulassungsunterlagen.

Das Ausgangsmaterial zu DEFA-Filmen und eine Archivkopie befanden sich im Staatlichen Filmarchiv der DDR. Als einzige Institution des Filmwesens der DDR wurde das Archiv bei der Wiedervereinigung vom Bundesarchiv übernommen. Das physische Eigentum am DEFA-Filmstock liegt beim Bundesarchiv.

Alle zum Rechtebestand der DEFA-Stiftung gehörenden Filme sind mit den relevanten filmografischen Angaben in einer **Datenbank** (www.defa-datenbank.de) erfasst.

In den DEFA-Studios realisierte Auftragsproduktionen für Fernsehen und andere Auftraggeber sind nicht Bestandteil des DEFA-Filmstocks.

Auf die Stiftung sind alle Rechte, soweit sie im Produktions- und Distributionsprozess entstanden sind, übergegangen.

Die DEFA-Filme sind bei folgenden Institutionen zugänglich:

PROGRESS Film-Verleih GmbH - www.progress-film.de
Icestorm Entertainment GmbH - www.icestorm.de
Bundesarchiv-Filmarchiv - www.bundesarchiv.de

Kontakt:
DEFA - STIFTUNG
Chausseestraße 103, 10115 Berlin
Telefon: 030/2 46 56 21 01
www.defa-stiftung.de

Stiftung Deutsches Rundfunkarchiv (DRA), Standort Babelsberg

Das DRA stellt seine Bestände einer Nutzung durch Dritte zur Verfügung, indem es Nutzungsvorhaben in einem effektiven Prozess der Auftragsabwicklung begleitet und dabei seine beratende Funktion aktiv wahrnimmt.

Das DDR-Archiv des DRA besitzt die Filme und Rechte für alle vom DDR-Fernsehen hergestellten Beiträge oder Filmen, einschließlich der bei anderen Produzenten im Auftrage des Fernsehens hergestellten. Im Zusammenhang mit der Nutzung seiner Materialien durch Dritte ermittelt und übermittelt das DRA Informationen zu Urheber- und Leistungsschutzrechten. Kopien von Fernsehbeiträgen und Aufführungsrechte werden gegen Gebühr geleistet. Außerdem ist umfangreiches Schriftgut archiviert.

Zum Archivbestand Fernsehen zählen alle Eigen-, Co- und Auftragsproduktionen des Deutschen Fernsehfunks (DFF) bzw. des Fernsehens der DDR, an denen der DFF Ausstrahlungsrechte hatte im Zeitraum 1951-1991..

Der Archivbestand Hörfunk (Berliner Rundfunk / Deutschlandsender / Stimme der DDR / Deutschlandsender Kultur / Radio DDR / Radio Berlin International / Jugendradio DT 64 / Landessender) im Zeitraum 1945-1991 zählt u. a. 300.000 Musik- und 150.000 Worttonträger.

Die **Bestands- und Programmdatenbanken** sowie konventionelle Nachweissysteme sind interne Arbeitsmittel und können aus rechtlichen Gründen nicht öffentlich angeboten werden. Sie stehen z. T. jedoch vor Ort im Archiv für die persönliche Nutzung zur Verfügung.

Kontakt:
Stiftung Deutsches Rundfunkarchiv, Standort Babelsberg
Marlene-Dietrich-Allee 20, 14482 Potsdam-Babelsberg
Telefon: 0331/58 12 - 222
www.dra.de

PROGRESS Film-Verleih GmbH

Geschäftsbereiche:
Lizenzen und Vertrieb/ Internationaler Verleih
Filmlizenzhandel für die Bundesrepublik und weltweit
Filmverleih für die Bundesrepublik, Österreich und die Schweiz
Film-Ausschnittdienst national und international

Die PROGRESS Film-Verleih GmbH hat für 15 Jahre die uneingeschränkten Rechte zur weltweiten gewerblichen und nichtgewerblichen Verwertung des DEFA-Filmstocks. Progress vergibt Lizenzen für die Aufführung von DEFA-Filmen in Kino und Fernsehen, verkauft Ausschnitte aus DEFA-Produktionen und verleiht DEFA-Filme.

Im Bestand befinden sich die DEFA-Wochenschauen und DEFA-Dokumentarfilme sowie Auftragsfilme der Ministerien und ausgewählte Werbefilme. Recherchen in **unternehmensinternen Datenbeständen** zu speziellen Themen sowie Kopien und Aufführungsrechte werden angeboten und sind kostenpflichtig (Preisliste Dienstleistungen). In diesem Filmstock befinden sich wesentliche Teile auch zum Thema Informatik/Rechentechnik.

Kontakt:
PROGRESS Film-Verleih GmbH
Immanuelkirchstr. 14, 10405 Berlin
Telefon: 030/24003-304
www.progress-film.de

Deutsches Institut für Animationsfilm e. V. (DIAF)

Das Deutsche Institut für Animationsfilm e.V. (DIAF) wurde 1993 in Dresden gegründet. In Dresden produzierte von 1955 bis 1990 das DEFA-Studio für Trickfilme, das staatliche Studio der DDR für Animationsfilme. Neben einem Filmstock von rund 2.000 Filmkopien archiviert das DIAF zahlreiche Filmmaterialien und Dokumente insbesondere aus dem Bereich des ostdeutschen Animationsfilms. Am Standort der Geschäftsstelle befindet sich auch die Bibliothek und Videothek des Instituts. Sichtungen von 35mm- und 16mm-Kopien sind am Archivstandort möglich. Das DIAF besitzt die Rechte zur Verwertung der Filme des DEFA-Trickfilmstudios. Im DIAF sind einige wenige Filme zum Thema Informatik/Rechentechnik zu finden.

Kontakt:
DIAF Dresden e. V.
Königstraße 15, 0197 Dresden
Telefon: 0351/3119041, 0351/3158548
www.diaf.de

DDR-Bildungsmedien “DDR-MEDIA“; Datenbank auf CD-ROM

Herausgeber: Institut für Bildung und Medien IB&M der Gesellschaft für Pädagogik und Information GPI e.V., Bernd Mikuszeit, Ute Szudra, Berlin 2003
Verlag B+B Medien Düsseldorf, ISBN 3-9-928060-96-1

Die GPI hat den Bildungsmedienbestand und die dazugehörigen Materialien des ehemaligen Instituts für Unterrichtsmittel der Akademie der Pädagogischen Wissenschaften der DDR vor dem Verfall gesichert, erschlossen und in einer Datenbank dokumentiert. Das von der VolkswagenStiftung geförderte Projekt trägt den Titel: "Dokumentation von audiovisuellen DDR-Schulmedien - Erfassung und Erschließung des DDR-Schulmedienbestandes“.

Die Datenbank „DDR-AV-MEDIA“ enthält insgesamt 7.405 Dokumente und erfasst damit 7.405 audiovisuelle DDR-Bildungsmedien. Sie ist nach den in der DDR-Bildungsmedienausstattung enthaltenen audiovisuellen Medienarten gegliedert und enthält 2.795 Filmdokumente, 2.105 Lichtbildreihendokumente, 406 Schallplattendokumente, 1.270 Magnettonbanddokumente, 229 Tonbildreihendokumente und 600 Projektionsfoliendokumente.

Hervorzuheben ist: die Datenbank enthält umfangreiche Informationen bzw. Fachtexte zu Themen der DDR-Bildung und Bildungsmedien.

Eine Datenbankrecherche erbrachte zum Thema Informatik/Rechentechnik 9 Filme und 16 Dia-Reihen.

Der Gesamt-Bestand der DDR-Bildungsmedien, insgesamt mehr als 16500 Dokumente, befindet sich z. Z. im Bildungsmedienarchiv Berlin-Friedrichsfelde.

Kontakt:
IB&M
Alt-Friedrichsfelde 60, Haus 14, 10315 Berlin
Telefon: 0 30/90 21 20 18
www.gpi-online.de

Dresden - Eine Filmographie der Stadt; Datenbank auf CD-ROM

Herausgeber: Filmverband Sachsen e.V.
Projektentwicklung und Redaktion: Uwe Penkert
Projektleitung, Redaktion, Datenbankaufbau: Christel Schröder
Dresden 2006

Die Filmographie [FS06] umfasst 3700 Datensätze über Filme, deren Inhalte sich thematisch mit Dresden verbinden. Die höchst authentische Datenbank wurde auf Basis einer großen Anzahl von Quellen zusammengestellt, die bis in die Anfangszeit filmischer Tätigkeit zurück reichen. Allein schon das Quellenverzeichnis und die Menge der genannten Mitwirkenden bieten dem Nutzer ein weites Feld weiterer Recherchemöglichkeiten. Nicht immer ist es gelungen die aktuellen Rechteinhaber und Bestandsorte der Filme zu ermitteln.

Zum Thema Informatik/Rechentechnik kann man in der Datenbank 22 Filmbeiträge ausfindig machen, deren Auftraggeber das Kombinat Robotron oder Vorgängereinrichtungen waren, des weiteren 4 Filme über die TH bzw. TU Dresden.

Kontakt:
Filmverband Sachsen e. V.
Schandauer Str. 64, 01277 Dresden
Telefon: 0351/3360099
www.filmverband-sachsen.de

Fachhochschule für Technik und Wirtschaft Berlin

Ein Studiengang der Fachhochschule für Technik und Wirtschaft Berlin (FTHW Berlin) befasst sich u. a. im Rahmen der Restaurierung technischer Kulturgüter auch mit der professionellen Restaurierung von Foto, Film und Datenträgern. In einem vom ehemaligen DDR-Institut für Film, Bild und Ton (IFBT) übernommenen Archiv lagern ca. 4.000 Medien. Zu etwa gleichen Teilen sind dies Filme, Magnetbänder für Audio und Video, Diaserien und Overheadfolien, oft mit dem dazugehörigen Ausgangsmaterial und Begleitmaterialien. Der Bestand bildet die gesamte audiovisuelle *Hochschul*didaktik der DDR seit der Gründung des IFBT im Jahre 1964 ab. Ab 2003 hat eine professionelle Aufarbeitung und Betreuung des Archivs begonnen. Zurzeit werden die Voraussetzungen geschaffen, die Nutzung des Bestandes wieder zu ermöglichen. Eine Auswahl der Filme wurde bereits für das Online-Portal der IWF Medien gGmbH in Göttingen digitalisiert und wird so der Öffentlichkeit bald wieder zur Verfügung stehen. Zur weiteren Bearbeitung gehören zunächst die Entwicklung einer **Datenbank** und die Restaurierung von beschädigten Medien. Weiter gehören dazu Recherchen zur Geschichte der DDR, Aufnahme von Kontakten zu den Rechteinhabern und vieles mehr. Archivwürdige Exemplare werden dem Bundesfilmarchiv übergeben.

Aus dem bisher erschlossenen Archivbestand sind nach vorläufiger Zusammenstellung zum Thema Informatik/Rechentechnik ca. 9 Filmtitel enthalten, welche im Rahmen der Hochschulausbildung in der DDR genutzt wurden. Nichtfilmische Medien sind z. Z. noch unerschlossen. Bis zur Fertigstellung einer Datenbank kann über alte IFBT-Kataloge im Einzelfall Zugriff auf den Archivbestand erfolgen. Einige Medien oder Dubletten fanden Eingang in Datenbank und Archiv des Instituts für Bildung und Medien IB&M der Gesellschaft für Pädagogik und Information GPI e.V. (siehe dort).

Kontakt:
Fachhochschule für Technik und Wirtschaft Berlin
FB5 Gestaltung, Konservierung und Restaurierung audiovisuelles und fotographisches Kulturgut
Wilhelminenhofstraße 75A, 12459 Berlin
Telefon: 030/5019-2151
www.f5.fhtw-berlin.de/krg

Literaturverzeichnis

[FS06] Filmverband Sachsen e.V. (Hrsg.): Dresden - Eine Filmographie der Stadt, CD-ROM, Datenbank zu Filmen mit Bezug auf Dresden mit 3700 Datensätzen, Projektentwicklung und Redaktion: Uwe Penkert, Projektleitung, Redaktion, Datenbankaufbau: Christel Schröder, Dresden 2006, www.filmverband-sachsen.de

[KLR98] Kreutz, Anja, Löcher, Ute, Rosenstein, Doris: Von „AHA“ bis „Visite“ - Ein Lexikon der Magazinreihen im DDR-Fernsehen (1952-1990/91), Hrsg.: Deutsches Rundfunkarchiv/DFG-Sonderforschungsbereich 240 „Bildschirmmedien“ der Uni-GH Siegen (Veröffentlichungen des Deutschen Rundfunkarchivs, Bd. 13), Potsdam 1998, Verlag für Berlin-Brandenburg, 336 S., ISBN 3-930850-89-3

[Kn96] Knopfe, Gerhard.: Kalendarium einer deutschen Spezies - Der populärwissenschaftliche Film der DEFA, S. 295-342, In: Schwarzweiß und Farbe - DEFA Dokumentarfilme 1946-92, Redaktion Günter Jordan, Ralf Schenk, Hrsg.: Filmmuseum Potsdam, Berlin 1996, Jovis-Verlag, 463 S., ISBN 3-931321-51-7

Weiterführende Quellen sind außer in den o. g. Institutionen z. B. auch in der Hochschule für Film und Fernsehen Potsdam-Babelsberg (HFF) - www.hff-postdam.de oder im Verbundkatalog Film - www.kobv.de/vkfilm. zu finden.

GI-Edition Lecture Notes in Informatics – Proceedings

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Busi-ness 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometric and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT: Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

GI-Edition Lecture Notes in Informatics – Thematics

T-1 Naumann, Schade (Hrsg.): Informatik in der DDR – eine Bilanz

T-2 Werner Altmann, Albert Mas y Parareda (Hrsg.) IT-Aus- und Weiterbildung: Chance und Herausforderung für Wirtschaft und Hochschule

T-3 Birgit Demuth (Hrsg.) Informatik in der DDR - Grundlagen und Anwendungen

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de