

**Projekt Prozeßlenkung mit
DV-Anlagen**

**PDV-Mitteilungen
1/1971**

**Gesellschaft für
Kernforschung mbH, Karlsruhe**

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H. KARLSRUHE

Beigefügt übersenden wir Ihnen, wie bereits mündlich angekündigt, die PDV-Mitteilungen 1/1971. Das Heft enthält die Dokumentation des Informationsgespräches und der fünf Fachgespräche, die die Projektleitung zur Vorbereitung des Projektrahmenplanes veranstaltet hat.

Wir senden dieses Heft, da es eine wesentliche Grundlage für das Projekt PDV enthält, an alle Teilnehmer der Fachgespräche und an alle Personen und Stellen, die ihr Interesse am Projekt bekundet haben.

In Zukunft wollen wir unsere PDV-Mitteilungen jedoch nur noch an diejenigen verschicken, die uns darum gebeten haben. Sollten Sie die PDV-Mitteilungen weiterhin beziehen wollen, möchten wir Sie bitten, uns dies mitzuteilen - falls nicht schon geschehen.

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
Projekt Prozeßlenkung mit DV-Anlagen

PS. Sollte Ihre Anschrift einen Fehler enthalten, bitten wir um Berichtigung.

Die PDV-Mitteilungen sind ein nach Bedarf erscheinendes Informationsorgan allgemeiner Art für das Projekt Prozeßlenkung mit DV-Anlagen. Sie enthalten wichtige Hinweise, Protokolle, Fachvorträge, Zusammenfassungen von Projektberichten, Mitteilungen über Literatur, die Ankündigung neuer Projektvorhaben und Ausschreibungen usw.

Die PDV-Mitteilungen sollen insbesondere die Abwicklung des Projektes nach außen hin transparent machen und sind daher allen Interessenten zugänglich.

Das vorliegende erste Heft 1/1971 enthält die Dokumentation des Informationsgespräches und der fünf Fachgespräche, die die Projektleitung im Zuge der Erstellung eines Projektplans veranstaltet hat.

Vorangestellt ist ein Vortrag von Herrn Reg.Dir. Bertuleit, Bundesministerium für Bildung und Wissenschaft (BMBW), gehalten beim Informationsgespräch, über die Förderung der Erschließung neuartiger DV-Anwendungen im Rahmen des 2. Datenverarbeitungsprogramms der Bundesregierung, in das das Projekt Prozeßlenkung mit DV-Anlagen eingebettet ist.

Karlsruhe, 1. Dezember 1971

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
Projekt Prozeßlenkung mit DV-Anlagen

<u>Inhaltsverzeichnis:</u>	<u>Seite:</u>
Informationsgespräch am 30. 9. 1971	
Förderung der Erschließung neuartiger DV-Anwendungen im 2. DV-Programm der Bundesregierung, Vortrag v. Reg.Dir. Bertuleit	2
1. Fachgespräch: "Prozedurorientierte Prozeß- Programmiersprachen" am 30.9./1.10.1971	9
Teilnehmerliste	
Ergebnisprotokoll	
Referate	
2. Fachgespräch: "Betriebssysteme" am 4./5.10.1971	62
Teilnehmerliste	
Ergebnisprotokoll	
Referate	
3. Fachgespräch: "Rechnerangepaßte Prozeß- peripherie" am 8.10. 1971	90
Teilnehmerliste	
Ergebnisprotokoll	
4. Fachgespräch: "Kommunikationssysteme in Prozeßwarten" am 22. 10. 1971	97
Teilnehmerliste	
Ergebnisprotokoll	
5. Fachgespräch: "Systemanalyse, Systemsynthese, Systemsimulation" am 4./5.11.1971	103
Teilnehmerliste	
Ergebnisprotokoll	
Referate	

Förderung der Erschließung neuartiger DV-Anwendungen im 2. DV-Programm
der Bundesregierung
- Kurzfassung -

Reg.Dir. Dipl.-Ing. Bertuleit
Bundesministerium für Bildung und Wissenschaft

Gliederung

1. Ziele des 2. DV-Programms
 2. Übersicht über die vorgesehenen Maßnahmen
 3. Förderung der Erschließung neuartiger DV-Anwendungen durch
den Bundesminister für Bildung und Wissenschaft
 4. Schlußbemerkungen
-

Das 2. Datenverarbeitungsprogramm enthält die Vorstellungen der Bundesregierung für die Förderung der Datenverarbeitung in den Jahren 1971 bis 1975. Das Programm umfaßt die Maßnahmen der Bundesminister für Wirtschaft und Finanzen, für Arbeit und Sozialordnung, des Inneren sowie für Bildung und Wissenschaft und wurde in Zusammenarbeit mit dem Fachbeirat für Datenverarbeitung und weiteren Sachverständigen aus Wissenschaft, Wirtschaft, Politik und Verwaltung aufgestellt.

1. Ziele des 2. DV-Programms

Die Bundesregierung verfolgt mit dem 2. DV-Programm die nachstehenden allgemeinen Ziele:

- Schaffung der Voraussetzungen für eine stärkere und breitere Anwendung der DV in Wirtschaft und Wissenschaft als Instrument der Rationalisierung und Leistungssteigerung zur Erhaltung und Förderung der Wettbewerbsfähigkeit unserer Volkswirtschaft
- Rationalisierung und Leistungssteigerung der von der öffentlichen Hand zu erbringenden Dienstleistungen und in der Hoheitsverwaltung mit Hilfe von DV-Anlagen
- Beherrschung der DV als eine der bedeutsamsten Schlüsseltechnologien

- Schaffung ausgewogener Wettbewerbsverhältnisse auf dem stark expandierenden DV-Markt.

2. Übersicht über die vorgesehenen Förderungsmaßnahmen

Besondere Bedeutung wird im Rahmen des 2. DV-Programms der Ausbildung beigemessen. In der beruflichen Bildung und an den Hochschulen müssen personelle Voraussetzungen dafür geschaffen werden, daß dem rasch steigenden Bedarf an Datenverarbeitungspersonal aller Qualifikationsebenen entsprochen, berufsbezogene Kenntnisse vermittelt und angemessene Fortbildung gewährleistet werden können.

Eine Integration der Datenverarbeitung in Forschung und Lehre aller Bereiche der Hochschule soll durch Bereitstellung einer ausreichenden und leicht zugreifbaren Rechenkapazität ermöglicht werden. Das Rechenanlagenbeschaffungsprogramm der Deutschen Forschungsgemeinschaft, die Rechnerbeschaffung im Rahmen der Bundesförderung des Ausbaus von Hochschulen und das Programm für regionale Großrechenzentren soll von der Deutschen Forschungsgemeinschaft fachlich koordiniert werden.

Die Forschung auf dem Gebiet der Datenverarbeitung an den Hochschulen soll durch ein überregionales Forschungsprogramm Informatik gefördert werden, das von Bund und Ländern gemäß Art. 91 b Grundgesetz gemeinsam finanziert wird. Es wird sich nicht nur auf die allgemeinen Grundlagen der Informatik, sondern auch auf die fachbezogene Anwendung der Datenverarbeitung erstrecken.

Die Bundesregierung wird sich in der Bildungsplanungskommission dafür einsetzen, daß die Datenverarbeitung über die eigentlichen DV-Berufe hinaus auch in den Ausbildungs- und Prüfungsplänen anderer Fachrichtungen angemessen berücksichtigt wird, damit die Möglichkeiten der Datenverarbeitung für alle Disziplinen und Berufszweige ausgeschöpft werden können.

Um den Bedarf an Hochschullehrern zu decken, sollen u.a. Fachkräfte aus Industrie und Ausland für eine Lehrtätigkeit gewonnen werden. Die Ausbildung von DV-Fachkräften auf der Fachschul- und Erstausbildungsebene soll an regionalen Schulen erfolgen. Neben dem Ausbau derartiger Schulen soll durch ein

Stipendium- und Kursusprogramm die Gewinnung einer ausreichenden Zahl von geeigneten Lehrkräften für diese Schulen gefördert werden.

Zur Erschließung neuartiger Aufwendungen der Datenverarbeitung sollen Forschungs- und Entwicklungsarbeiten gefördert werden, um das System- und Anwendungswissen verfügbar zu haben, mit dem im öffentlichen und nichtöffentlichen Bereich Datenverarbeitungsanlagen als Instrument der Planung, Entscheidungshilfe und Rationalisierung nutzbar gemacht werden können. Die Förderungsmaßnahmen werden sich auf folgende Bereiche erstrecken:

- Rechnerunterstützte Informations-, Dispositions- und Entscheidungssysteme
- Datenverarbeitung im Bildungswesen
- Datenverarbeitung in der Medizin
- Prozeßlenkung mit DV-Anlagen
- Rechnerunterstütztes Entwickeln und Konstruieren.

In Fortführung und Ergänzung der vorgenannten Förderungsmaßnahmen sollen für DV-Anwendungen im öffentlichen Bereich (z.B. Medizin und Bildungswesen) und bei solchen DV-Anwendungen, an denen ein besonderes öffentliches Interesse besteht, ausgewählte Demonstrations-DV-Projekte realisiert werden.

Die Entwicklung standardisierter Software-Pakete soll helfen, kurzfristig eine wesentliche Rationalisierung mittels Datenverarbeitungsanlagen bei einer größeren Zahl von Anwendern in der gewerblichen Wirtschaft zu erreichen (Maßnahme des Bundesministeriums für Wirtschaft und Finanzen).

Richtlinien für die Beschaffung von Datenverarbeitungsanlagen mit Bundesmitteln sollen sicherstellen, daß die technischen Möglichkeiten für eine Standardisierung von Anlagen und Programmen ausgeschöpft werden und bei den Beschaffungen Hersteller mit europäischer Basis mindestens einen ihrer Leistungsfähigkeit entsprechenden Anteil erhalten.

In Fortführung der bisherigen Maßnahmen soll auch künftig die industrielle Forschung auf Gebieten, deren Beherrschung für eine künftige Konkurrenzfähigkeit der Datenverarbeitungs-Unternehmen von entscheidender Bedeutung ist, durch Zuschüsse gefördert werden.

Wegen ihrer großen Bedeutung soll die Entwicklung von Großrechner-Teilnehmersystemen weiterhin unterstützt werden. Einzelne laufende zukunftsorientierte

Projekte zur Entwicklung von mittleren Universalrechnern und von Prozeßrechnern sollen zu Ende geführt werden. Darüber hinaus soll die Förderung ausgewählter Projekte zur Entwicklung von Datenverarbeitungsanlagen und Programmen im marktnahen Bereich bis zur Herstellung ausgewogener Wettbewerbsverhältnisse fortgeführt werden.

3. Förderung der Erschließung neuartiger DV-Anwendungen durch den BMW

DV-Anlagen können in nahezu allen Aufgabengebieten eingesetzt werden. Förderungsmaßnahmen sind jedoch nicht für alle denkbaren Anwendungen möglich, sie müssen sich vielmehr gezielt auf ausgewählte Bereiche erstrecken, um effektiv zu sein.

Ausgehend von dieser Feststellung ist die Bundesregierung der Auffassung, daß insbesondere Förderungsmaßnahmen mit Multiplikatoreffekt erforderlich sind. Damit soll initiativ die Entwicklung und Anwendung integrierter DV-Systeme eingeleitet und bis zu einem solchen Stand gebracht werden, von dem aus die Pflege, Weiterentwicklung und Verbreitung der Systeme von Anwendern, ihren Verbänden oder Interessengemeinschaften, von DV-Dienstleistungsunternehmen oder der DV-Industrie aus eigener Kraft weitergeführt werden kann. Die Bundesregierung geht bei der Auswahl der Förderungsmaßnahmen von folgenden Kriterien aus:

- Durch die Förderungsmaßnahme ist ein wesentlicher Nutzen für die Gesellschaft zu erwarten.
- Die Ergebnisse der Maßnahmen führen zu einer Steigerung oder Verbesserung der von der öffentlichen Hand zu erbringenden Dienstleistungen.
- An den Ergebnissen besteht Bedarf bei einer Vielzahl von potentiellen Anwendern, die nicht in der Lage sind, entsprechende Entwicklungen aus eigener Kraft durchzuführen.
- Mit der Förderungsmaßnahme wird ein Standardisierungseffekt erreicht.
- Es besteht begründete Aussicht, daß die mit öffentlichen Mitteln entwickelten Programme später ohne Förderung weiterentwickelt werden.
- Die Förderungsergebnisse sind allgemein zugänglich (bei Eigenbeteiligung des Zuwendungsempfängers an den Entwicklungskosten zu angemessenen Gebühren).

- Die Höhe der notwendigen Förderungsmittel steht in einem angemessenen Verhältnis zu dem voraussichtlichen Nutzen der späteren DV-Anwendung.

Für zukunftsorientierte, breite Anwendungen aus dem öffentlichen und nicht-öffentlichen Bereich soll das notwendige System- und Anwendungswissen erarbeitet werden, um DV-Anlagen als Instrument der Planung, Entscheidungshilfe und Führung anwenden und die technischen Möglichkeiten der DV-Anlagen voll nutzen zu können.

Es sollen Forschungs- und Entwicklungsarbeiten gefördert werden, um

- ausgewählte Bereiche auf die Möglichkeit der Anwendung von Datenverarbeitungsanlagen zu untersuchen
- den Wissensstoff dieser Anwendungsgebiete bis zur Programmierung aufzubereiten
- die Programmierung bis zur Anwendungsreife durchzuführen
- die Programmsysteme praktisch zu erproben und umfassend zu dokumentieren.

Schwerpunkt der Arbeiten bei der Erstellung der Programme (problemorientierte Software-Pakete) wird es sein, Programme zu gewinnen, die modular aufgebaut sind, damit sie vom jeweiligen Anwender leicht seinen speziellen Anforderungen angepaßt werden können, und die gleichzeitig weitgehend unabhängig von bestimmten Anlagen sind, damit ihre breite Verwendbarkeit erreicht werden kann. Von gleichrangiger Bedeutung sind problemorientierte Programmiersprachen, die einen hohen Anwendungskomfort bieten und den besonderen Belangen einzelner Anwendungsgebiete (z.B. Auskunftssysteme, Prozeßregelung) Rechnung tragen. In diesem Zusammenhang müssen z.B. auch die Probleme der Datensicherung sowohl gegen technische Fehler als auch gegen Mißbrauch behandelt werden. Ein weiterer Schwerpunkt ist die Entwicklung von speziellen Geräten zur Datenerfassung, Dateneingabe und Datenausgabe, die für bestimmte Anwendungen erforderlich sind.

Der wesentliche Teil der Arbeiten wird im Rahmen von ausgewählten, langfristigen Projekten ausgeführt werden. Die Bearbeitung dieser Projekte soll aus Gründen der Effizienz in der Regel jeweils einem Projekträger übertragen werden, der

- die Detailkoordination des Projektes übernimmt,
- zu einem gewissen Teil Forschungs- und Entwicklungsarbeiten auf dem jeweiligen Gebiet durchführt, um über den speziellen Sachverstand unmittelbar zu verfügen,
- einen wesentlichen Teil der Arbeiten als spezifizierte Unteraufträge vergibt (z.B. an Hochschulinstitute, Software-Unternehmen, DV-Hersteller),
- gegebenenfalls die Erprobung und Einführung von Pilotsystemen übernimmt.

Dem Projektträger ist auch die Verantwortung zur Prüfung der Voraussetzungen zu übertragen, die gewährleisten, daß zu fördernde Entwicklungen den Bedürfnissen der Praxis entsprechen und eine breite Nutzung der erarbeiteten Lösungen erzielt werden kann.

In Fortführung und Ergänzung der Maßnahmen zur Förderung von Forschung und Entwicklung sollen für DV-Anwendungen im öffentlichen Bereich (z.B. Medizin und Bildungswesen) und bei solchen DV-Anwendungen, an denen ein besonderes öffentliches Interesse besteht, ausgewählte Demonstrations-DV-Projekte realisiert werden. Hierbei sollen die Anwendungsmöglichkeiten der erarbeiteten und sonst verfügbaren Programmsysteme in praktischen Routinebetrieb demonstriert werden. Damit soll erreicht werden, daß die Innovationsphase möglichst schnell durchlaufen wird und künftige DV-Märkte erschlossen werden.

Ein ständiger Kontakt und eine enge Zusammenarbeit zwischen den Forschungs- und Entwicklungsstellen und den künftigen Anwendergruppen ist sicherzustellen, damit einerseits die Anforderungen der Benutzer an die zu entwickelnden Systeme ständig bekannt sind und berücksichtigt werden können und damit andererseits die Benutzer frühzeitig über die für sie entwickelten Systeme informiert sind.-Bei der Vorbereitung und Durchführung der Förderungsmaßnahmen - einschließlich der Projektkontrolle - wird sich der BMBW durch Sachverständigenausschüsse gutachterlich beraten lassen. Die Projekte sollen in eine Definitionsphase und mehrere Phasen der Verwirklichung gegliedert werden.

4. Schlußbemerkungen

Bei dem bisherigen Verfahren der Förderung von DV-Anwendungen wurden Mittel bereitgestellt fast ausschließlich aufgrund von einzelnen, mehr oder minder

zufällig eingereichten Anträgen. Für die ersten Jahre der Förderung mag dies ein vertretbares Verfahren gewesen sein, um DV-Aktivitäten, die in der BRD vorlagen, anzuregen und zu verstärken. Jetzt jedoch soll Grundlage für Förderungsmaßnahmen für die einzelnen Anwendungsbereiche je ein konkreter Projektplan sein. In einem solchen Projektplan müssen insbesondere angegeben sein die angestrebten Ziele, die notwendig auszuführenden, auf die Belange des jeweiligen Anwendungsbereichs abgestellten, nach Prioritäten geordneten und aufeinander abgestimmten Forschungs- und Entwicklungsarbeiten sowie die jährlich benötigten Mittel. Ein solcher Projektplan soll etwa jährlich fortgeschrieben werden. Ferner soll sichergestellt werden, daß das Verfahren und der Inhalt der Förderungsmaßnahmen transparent gemacht wird und bei der Durchführung der Maßnahmen, z.B. bei der Vergabe von Unteraufträgen im Rahmen eines Projekts, auch eine gesunde Konkurrenz zwischen den an einer Mitwirkung interessierten Institutionen stattfindet.

Der soeben genannte Projektplan soll von jeweiligen Projektträgern erarbeitet und fortgeschrieben werden. Für das Projekt "Prozeßlenkung mit DV-Anlagen" hat die Gesellschaft für Kernforschung die Trägerschaft übernommen, mit dem Aufbau eines entsprechenden Projektstabes ist begonnen worden.

Die VDI/VDE-Fachgruppe Regelungstechnik hat auf Anregung des BMW eine Studie "Prozeßregelung mit Datenverarbeitungsanlagen" erarbeitet und Anfang dieses Jahres vorgelegt. Das in der Studie vorgeschlagene Konzept - Herr Dr. Syrbe wird es noch erläutern - ist in einem Sachverständigenausschuß des BMW beraten worden, der Ausschuß hat empfohlen, auf der Grundlage der Studie eine detaillierte Projektplanung vorzunehmen.

Wie Sie der Einladung zu dem heutigen Informationsgespräch und zu den anschließenden Fachgesprächen entnommen haben, soll die Ermittlung der Aufgabenschwerpunkte und die Erarbeitung von Vorschlägen für konkrete Förderungsmaßnahmen auf der Grundlage einer breiten, kritischen Sachdiskussion erfolgen. Ich bitte Sie hierbei um Ihre Unterstützung und aktive Mitarbeit.

30. September 1971

1. Fachgespräch am 30.9. und 1.10.1971
über "Prozedurorientierte Prozeßprogrammiersprachen".

Teilnehmerliste

für das Informationsgespräch am 30. 9. 1971 und das Fachgespräch
über "Prozedurorientierte Prozeßprogrammiersprachen" am 1. 10.1971
im Kernforschungszentrum Karlsruhe

<u>Name:</u>	<u>Institution:</u>
Mußtopf	SCS, Hamburg
Hotes	SCS, Hamburg
Staschke	AEG-TELEFUNKEN, Konstanz
Schwerdtner	IBM, Sindelfingen
Werum	Entwicklungsbüro Werum, Erbstorf
Müller	TU Braunschweig
Born	Betriebsforschungsinstitut VDEh
Quack	Universität Stuttgart
Eichner	Universität Stuttgart
Pfeifer	TH Aachen
Schreitmüller	TH Aachen
Reh	ESG, München
Eichenauer	ESG, München
Elzer	Universität Erlangen
Holleczeck	Universität Erlangen
Hofmann	Farbwerke Hoechst AG
Syrbe	Fraunhofer-Gesellschaft
Radius	AEG-TELEFUNKEN, Frankfurt
Willibald	AEG-TELEFUNKEN, Frankfurt
Ehling	AEG-TELEFUNKEN, Berlin
Ent	Daimler Benz AG
Kießling	Daimler Benz AG
Nolte	Gutehoffnungshütte
Endres	IBM, Böblingen
Tarara	Bundesanstalt für Flugsicherung, Frankfurt
Ende	Messerschmidt-Bölkow-Blohm GmbH, München
Frech	Unicomp und DFE, Blankenloch
Greifeld	GfK/GF
Krüger	GfK/IDT
Stams	GfK/PDV
Eckert	GfK/PDV

Heinrich	Mathematischer Beratungsdienst
Klessmann	Hahn-Meitner-Institut, Berlin
Wölcken	Strahlencentrum, Gießen
Melcher	Robert Bosch GmbH, Sindelfingen
Bühler	Robert Bosch GmbH, Sindelfingen
Rohde	AEG-TELEFUNKEN, Konstanz
Didic	Unicomp und DFE, Blankenloch
Haase	Unicomp und DFE, Blankenloch
Lauber	Universität Stuttgart
Bertuleit	BMBW
Bollmann	Siemens AG, Erlangen
Sinzinger	Siemens AG, Erlangen
Kaltenecker	Siemens AG, Karlsruhe
Lemcke	Gemini Computer Systems, Offenbach
Kaufmann	KFA, Jülich
Baumann	DEMAG, Duisburg
Schäfer	DEMAG, Duisburg
Widdel	ADV/ORGA Unternehmensberatung, Wilhelmshafen
Müller	ADV/ORGA Unternehmensberatung, Universität Karlsruhe
Goos	GfK
Tradowsky	GfK
Ottes	GfK
Stittgen	GfK
Niedermeyer	GfK/ZAED
Baumann	TU München
Heller	BASF, Ludwigshafen
Henn	TU München
Müller	BBC, Mannheim
Brandes	Universität Karlsruhe
Kreuter	Siemens, Karlsruhe
Timmesfeld	Entwicklungsbüro Werum, Erbstorf
Aumann	I.R.A., Friedrichshafen
Fröhlich	Unicomp und DFE, Blankenloch
Pfeiffer	KFA, Jülich
Stephan	Nixdorf
Mittendorf	Siemens AG, Karlsruhe

Hofmann
Überschär
Kleinlogel
Frevert
Wedel
Martin
Wettstein

Siemens AG, Karlsruhe
Siemens AG, Erlangen
Interdigit
Hahn-Meitner-Institut, Berlin
Mathematischer Beratungsdienst
BBC, Mannheim
Universität Karlsruhe

E r g e b n i s - P r o t o k o l l

zum Fachgespräch "Prozedurorientierte Prozeß-Programmiersprachen "
am 30.9./ 1.10. 1971 im Kernforschungszentrum Karlsruhe.

1. Prozeßrechnersprachen

Das Fachgespräch konzentrierte sich auf das generelle Problem, Prozeß-rechnersprachen als Arbeitshilfsmittel zu benutzen. Es bestand volle Einigkeit darüber, daß die gesamte Programmierarbeit unabhängig von speziellen Maschinen betrieben werden sollte, damit die erarbeiteten Programme langfristig, auch bei Wechsel von Maschinen, benutzt werden können.

Aus den verschiedenen Aufgabenstellungen und Funktionen von Prozeß-rechnersprachen ergaben sich 4 verschiedene, nach Möglichkeit maschinen-unabhängige Sprachtypen:

- Zwischensprachen (1), die dazu dienen, Compiler und Betriebs-systeme zu programmieren.
- Zwischensprachen (2), in die ein Compiler das Quellprogramm übersetzt, damit die Compiler selbst weitgehend maschinen-unabhängig sein können.
- Allgemeine prozedurorientierte Sprachen (3) für Echtzeit-Anwendungen wie etwa PEARL.
- Einfachere prozedurorientierte Sprachen (4) auf einem niedrige-ren Niveau, die aber noch geeignet sind, Benutzerprogrammpakete zu erstellen.

Es bestand Unklarheit darüber, ob es möglich ist, die Sprachen (1), (2) und evtl. (4) als eine "Low Level Language" zusammenzufassen.

In der Diskussion zeigte sich, daß von verschiedenen Gruppen zwei ver-schiedene Tendenzen vertreten wurden:

- Die 1. Tendenz geht in Richtung zur generellen Programmierung von Anwendungsprogrammsystemen in einer maschinenunabhängigen

"Low Level Language". Diese Tendenz wird vor allen Dingen von den Vertretern der Software-Häuser verfolgt.

- Die 2. Tendenz zielt darauf hin, eine leistungsfähige, allgemeine prozedurorientierte Sprache gemäß (3) zu entwickeln und die Aufgabe der Programmierung dem Benutzer der Rechenanlage zu überlassen. Diese Vorstellung wird von den Vertretern der Rechenmaschinenhersteller und der Anwendergruppen verfolgt.

In den Rahmen der Diskussion über Prozeßrechnersprachen fielen noch folgende besondere Bemerkungen:

- Anwenderprogramme müssen gut testbar sein. Hierzu wäre als Testebene die Ebene einer Zwischensprache eventuell brauchbar.
- Die Informationsübergabe von der höheren Programmiersprache gemäß (3) oder auch (4) auf eine "Low Level Language" gemäß (1) oder (2) beim Compilieren sollte untersucht und bei der Sprachdefinition beachtet werden.
- Als ein besonderes Mittel zur Definition geeigneter Zwischensprachen wäre die Erstellung des Modells eines virtuellen Prozeßrechners zu empfehlen.
- Die Relationen zwischen Rechner-Hardware-Strukturen, Betriebssystem-Strukturen, Zwischensprachen und prozedurorientierten Programmiersprachen sollten grundsätzlich erforscht werden.
- Für Planung, Entwicklung und Test ganzer Systeme wurde die Entwicklung von Emulationsverfahren vorgeschlagen.
- Bei der Sprachentwicklung müssen auch Rechnergröße und Mehrrechner-systeme berücksichtigt werden. Subsets von höheren Programmiersprachen sollen möglichst frühzeitig entwickelt werden. Möglichkeiten der Compilation auf derselben Maschine (selfsupporting) oder auf einer anderen Maschine (non-selfsupporting) sollten besonders studiert werden.

2. Standards

Über die Frage der Standardisierung wurde ausführlich diskutiert. Folgende Gesichtspunkte wurden als wichtig erkannt:

- Standardisierung ist die Grundlage für die Zusammenarbeit verschiedener Gruppen und damit für einen raschen Fortschritt.
- Standardisierungen sollten nicht zu voreilig eingeführt werden, da sie sonst entwicklungshemmend wirken könnten.
- Wir sollten uns an den internationalen Standardisierungsbestrebungen möglichst intensiv beteiligen, damit eventuelle für uns schädliche technische und finanzielle Auswirkungen vermieden werden und damit wir unsere Entwicklungen dem internationalen Standard frühzeitig genug anpassen können.

3. Pilotimplementierungen und praktische Erprobung

Die Durchführung von Pilotimplementierungen von PEARL und angepaßten Betriebssystemen sollte mit Vorrang betrieben werden aus folgenden Gründen:

- Sammeln von Erfahrung über die Brauchbarkeit der PEARL-Definitionen
- Ermittlung zweckmäßiger Eigenschaften von Betriebssystemen
- Erproben von Methoden zum Compiler-Bau
 - a) die "Bootstrapping-Methode"
 - b) die Erzeugung über "Low Level Languages (1)"
- Untersuchung geeigneter Zwischensprachen (2), in die übersetzt wird
- Erprobung dieser Programmieretechniken im Hinblick auf ihre Auswirkung auf die Erschließung der Datenverarbeitung im Bereich der Prozeßlenkung

- Sammeln von Erfahrung und dadurch bessere Möglichkeiten der Einflußnahme auf die internationale Standardisierung

4. Vorschläge von Themen für den Projektplan

- Definition von PEARL abschließen
Priorität: sehr schnell
- Codierung vorgegebener Anwendungsprobleme in PEARL und anderen Sprachen zu Sprachvergleichszwecken und zur Erprobung von PEARL
Priorität: schnell
- Pilotimplementierung von PEARL und ähnlichen Sprachen
Priorität: schnell
- Untersuchungen über Subsetting, speziell als Anwendung für kleineren Maschinen, Untersuchung über selfsupporting und non-selfsupporting Systeme für mittlere und kleine Maschinen
Priorität: schnell
- Untersuchungen über Kopplung getrennt compilierter Programme
Priorität: mittelfristig
- Untersuchung des Problems der Zwischensprachen und des virtuellen Rechners
 - + Zwischensprachen zum Herstellen von Compilern und Betriebssystemen
 - + Zwischensprache, in die der Compiler übersetzt
 - + Zwischensprache, geeignet zur Erstellung von Anwendungsprogramm-paketenPriorität: mittelfristig
- Studien über stufenweisen Aufbau von Sprachen von der niederen Sprache bis zur Benutzersprache
Priorität: mittelfristig
- Untersuchungen über Testsysteme
Priorität: mittelfristig

- Programmbibliothek von problembezogenen Modulen

Priorität: langfristig

- Erforschung von für Programmiersprachen besonders geeigneten Hardware- und Betriebssystemstrukturen

Priorität: langfristig

5. Allgemeine Empfehlungen des Plenums an die Projektleitung

Folgende Empfehlung wurde von den Teilnehmern des Fachgesprächs (einstimmig) gegenüber der Projektleitung ausgesprochen:

Die Teilnehmer des Fachgesprächs empfehlen die Förderungsmaßnahmen auf dem Gebiet der prozedurorientierten Programmiersprachen und ihrer Übersetzer zum frühestmöglichen Zeitpunkt aufzunehmen. Dabei scheint es notwendig, den besonders im Ausland forcierten Bemühungen, internationale Standards für Prozeßsprachen auf unzureichenden technischen Grundlagen kurzfristig durchzusetzen, geeignete - durch Prototypentwicklungen und Erfahrungen gestützte - eigene Konzepte entgegenzustellen. Die Teilnehmer des Fachgesprächs unterstreichen die Bedeutung einer ausreichenden Repräsentanz deutscher Fachleute in den mit der Standardisierung befaßten internationalen Gremien und empfehlen der Projektleitung, durch geeignete Maßnahmen diese Vertretung zu unterstützen.

gez. Stams

gez. Elzer



BETRIFFT:

Fachgespräch über die Förderung
der Forschung und Entwicklung auf dem Gebiet der
prozedurorientierten Prozeßprogrammiersprachen
- Diskussionsbeitrag Dr. Ehling -

Die Erfahrung hat gezeigt, daß Betriebssysteme aus mehreren Schichten bestehen können. Jede Schicht definiert für die darüber liegende eine neue, eine virtuelle Maschine. Programme einer Schicht werden von ihrer virtuellen Maschine aufgerufen und unterbrochen, oder sie geben die Kontrolle an diese virtuelle Maschine zurück. Diese kann sich wie ein Mehrprozessorsystem verhalten. Dies gilt für die Schichten untereinander wie auch für das gesamte Betriebssystem bezüglich der Anwender-Programme.

In Anbetracht dieses Sachverhaltes bedarf es großer Sorgfalt bei der Festlegung von Schnittstellen zwischen einer Prozedursprache und Betriebssystemen. Weiterhin besteht die Möglichkeit von Differenzen zwischen den seitens der Sprache einerseits und vom Betriebssystem andererseits vorgesehenen Mittel für die Steuerung quasiparalleler Abläufe.

Dies ist besonders zu berücksichtigen beim Betrieb mit Hintergrundspeichern, wo Transfer-Zeiten länger als Laufzeiten der Programme sein können.

Bei größeren Programmsystemen ist es meistens zwingend notwendig, Teilsysteme getrennt zu kompilieren und das System folglich im Vollzug eines Ladevorgangs zusammenzustellen. Dies rechtfertigt sich auch aus der Natur der Sache. Prozeßrechnersysteme lassen sich häufig - ähnlich wie Betriebssysteme - als Gruppierungen relativ unabhängiger Einzelfunktionen um einen gemeinsamen, permanent im System vorhandenen Datenbestand auffassen. Dieser Datenbestand ist ein Modell für den Zustand des Prozesses, er enthält komprimiert die zeitlichen Abläufe, soweit diese für weitere Entscheidungen wesentlich sind. In dieser Sicht ist ein Programmsystem nicht mehr ein monolithischer "Ablauf", der in einem Zuge entworfen und implementiert wird. Das System wird dann vielmehr zu einem mehr oder weniger lose gekoppelten Netzwerk einzelner Funktionen. Arbeitsteilige Behandlung wird damit möglich und ist bei größeren Systemen auch unterläßlich.

Es scheint uns nur eine Frage der Zeit, daß diesem Sachverhalt auch von höheren Programmiersprachen Rechnung getragen werden muß.

Manuskript-Niederschrift

des am 1. 10. 71 vorgetragenen Kurzreferates

von Gerd Müller, BEC Mannheim

Prozeßorientierte Programmiersprachen

Eines der wesentlichsten Probleme beim Einsatz von Rechnern für die Prozeßlenkung ist die Anfertigung der Rechnerprogramme, der Test derselben und die abschließende Dokumentation.

Es ist heute unbestritten, daß dies nicht mehr mit einer Assembler-Sprache bewältigt werden kann, sondern daß hierzu eine Programmier-Sprache erheblich höheren Niveaus nötig ist, eine prozeßorientierte Programmier-Sprache. Dies wird ersichtlich aus den ganz enormen Bemühungen in Europa und in den USA, eine solche Sprache zu entwickeln, bzw. einzusetzen.

Bei der Diskussion der Frage, auf welchem Sprachen-Niveau sich eine solche prozeßorientierte Programmiersprache befinden soll, wird leider auch heute noch das Argument vorgebracht, der zu erlaubende Programmier-Komfort müsse dort seine Grenze haben, wo das erzeugte Objektprogramm nicht mehr Speicherplatz benötige, wie ein gleiches Programm, das in der Assembler-Sprache abgefaßt wurde.

Diese Auffassung von dem Begriff "Effektivität" zeugt von mangelndem Verständnis für den Gesichtspunkt der Wirtschaftlichkeit beim Einsatz von Prozeßrechnern.

"Effektivität" bedeutet im Bereich der industriellen Datenverarbeitung:

- zur Lösung einer gestellten Aufgabe die Anzahl der einzusetzenden Arbeitskräfte einerseits und den Umfang der benötigten Hardware andererseits kostenoptimal zu kombinieren.

D. h. zwischen den beiden Extrema:

sehr billige Hardware → hoher Einsatz von Arbeitskräften
und

sehr, komfortable und teure Hardware → wesentlich geringerer
Einsatz von Arbeitskräften,

muß ein wirtschaftlicher Mittelweg gefunden werden.

Zur Beschreibung dieses Mittelweges, über den man sich bei den entsprechenden Entwicklungsabteilungen in den USA und in Europa weitgehend einig ist, zitiert man am besten die Eigenschaften, die eine prozeßorientierte Programmiersprache mindestens besitzen sollte. Auch sollte man sich dabei vor Augen halten, daß sich der Kreis der künftigen Anwender in der Mehrzahl der Fälle aus Ingenieuren und Naturwissenschaftlern zusammensetzt, bei denen die Kenntnisse über den einzusetzenden Prozeßrechner nicht an erster Stelle ihres beruflichen Wissens stehen.

Eine solche Sprache muß es ermöglichen:

- bei Anweisungen für den Austausch von Daten mit dem Prozeß die Prozeßendstellen mit einem vom Programmierer zu wählenden Namen ansprechen zu können, ohne jedesmal die betroffenen peripheren Geräte bezeichnen zu müssen; dies gilt sinngemäß auch für die Interruptleitungen;
- den gesamten Hardware-Plan einfach in ein Programm umzusetzen;
- mit absoluten und relativen Zeiten zu operieren;
- mit arithmetischen Daten, sowie mit Bit- und Zeichenketten zu arbeiten;
- das Taskmanagement eines Programms einfach zu handhaben;

- für die Programmierung der Ausgabeprotokolle entsprechende Anweisungen solchen Niveaus zu verwenden, wie es z. B. von PL/1 her bekannt ist;
- Optimierungs-Anweisungen zur Steuerung des Compilations-Ablaufs einzusetzen.

Die Lösung, die BBC Mannheim bisher erarbeitet hat, (in Zusammenarbeit mit Fa. Werun, Lüneburg), benutzt die, in Zukunft wohl am häufigsten verwendete Programmiersprache PL/1 als Basis. Aus dem umfangreichen PL/1 wurde ein geeignetes Subset ausgewählt. Als prozeßspezifischer Zusatz wurde eine weitere Sprache, PAS1, entwickelt. Diese ist nicht in das syntaktische Korsett von PL/1 eingezwängt worden; sie ist jedoch "PL/1-like".

Dieses Software-Paket, das den oben geforderten Eigenschaften sämtlich gerecht wird, ist in einer ersten Ausbaustufe fertig. Für Vorprojekte wurden umfangreiche Prozeßlenkungs-Programme mit sehr guten Ergebnissen angefertigt.

Der rein syntaxorientiert arbeitende Compiler, (syntaktische Änderungen der Sprache werden somit leicht bewältigt!), ist so konzipiert, daß er weitgehend maschinenunabhängig arbeitet und erst ganz zum Schluß anhand einer (austauschbaren) Befehlsbibliothek auf die Zielmaschine Bezug nimmt. Der Compiler ist überdies selbst in PL/1 geschrieben und ermöglicht somit die Kompilation des Quellenprogramms sowohl

- auf dem Prozeßrechner selbst, als auch
- in einem Rechenzentrum.

Für die Förderung nach dem 2. DV-Programm sollte man nach Meinung unseres Hauses von jener Entwicklungs-Gruppe ausgehen, in der jetzt schon die namhaftesten Anwender von PDV'n mit Erfolg zusammenarbeiten, nämlich dem PEARL-Arbeitskreis. Der dort in Bälde fertiggestellte Sprachvorschlag müßte an mehreren Stellen mit einem festen zeitlichen Limit implementiert werden. Daran anschließen sollte sich ein freimütiger Austausch von know-how, eine Weitergabe-sämtlicher Arbeitsunterlagen usw.

Es ist sehr wahrscheinlich, daß auf diesem Weg Ergebnisse erzielt werden, die im Vergleich zu entsprechenden Entwicklungen in den USA zu einem deutschen Anwendungs-Vorsprung führen können.

- 23 -

Kurzreferat
anlässlich des Fachgesprächs "Allgemeine Prozedursprachen"
vom 1. 10. 1971
im Rahmen des 2. DV Förderungsprogrammes,
Projekt "Prozeßlenkung mit DV-Anlagen"

RATIONALISIERUNG DER PRODUKTION VON PROGRAMMSYSTEMEN

Günter Mußtopf

SCIENTIFIC CONTROL SYSTEMS
LTD & CO GMBH

2000 Hamburg 39
Überseering 8

1. Einleitung

Die im Hause SCS durchgeführten Projekte auf dem Sektor Systemsoftware und Realzeit-Anwendungen führten zu der Feststellung, daß auch heute noch eine große Anzahl von Aufgaben existieren, die eine maximale Effizienz der Maschinenprogramme fordern. Üblicherweise werden in diesem Bereich maschinenorientierte Programmiersprachen eingesetzt. Das natürliche Streben eines von Hardware-Herstellern unabhängigen Software-Hauses ist, im Interesse einer größeren Flexibilität und rationelleren Produktion derartige Programmsysteme mit Hilfe maschinenunabhängiger Programmiersprachen herzustellen. Der Realzeit-Anwender besitzt ähnliche Wünsche, wenn auch aus anderen Motiven.

Das Thema dieses Kurzreferates wird auf das oben beschriebene Teilgebiet der Software-Produktion eingeschränkt. Alle folgenden Aussagen und Argumente sind stets unter dieser Voraussetzung zu betrachten.

Untersucht wurde im Hause SCS die Produktion mittlerer bis umfangreicher Programmsysteme. Dabei wurde gefordert, daß die von einem Compiler erzeugbare Qualität näherungsweise gleich der von einem Assembler erzeugbaren Qualität sein muß. Die skizzierte Problematik tritt beispielsweise bei der Entwicklung von

- o Betriebssystemen,
- o Compilern,
- o Basiselementen für modulare Anwendungssysteme und
- o speziellen Anwendungen

auf. Die Arbeiten führten zur Entwicklung einer prozedurorientierten, mittleren Programmiersprache, deren Grundideen im folgenden geschildert werden.

Bemerkenswert erscheint noch, daß aus der heutigen Sicht betrachtet an anderen Stellen Programmiersprachen mit ähnlichen Zielsetzungen konzipiert wurden. Dies sind beispielsweise PL 360, PS 440, ASTRE, JOVIAL und CORAL 66.

2. Ziele

Während der in unserem Hause durchgeführten Untersuchungen wurde neben den Entwicklungskosten und laufenden Kosten

- o der Effizienz des erzeugbaren Produktes,
- o der Methodenorientierung der Sprache und
- o der Sicherheit

besondere Beachtung geschenkt. Wesentlichen Einfluß auf das Ergebnis der Analyse hatten nicht zuletzt die Randbedingungen, daß

- o Sprache und Pilotcompiler in naher Zukunft (Mitte bis Ende 1972) zur Verfügung stehen soll,
- o die Sprache ohne Änderung des Compilers in Richtung Anwendungsorientierung erweiterbar ist und
- o das Testen von Programmen für Kleinrechner auf Großanlagen unterstützt werden soll.

Unter der Effizienz des Produktes wird hier die erzeugbare Qualität des Maschinenprogrammes in bezug auf Laufzeit und/oder Speicherbedarf verstanden. Dabei muß jeweils vom Programmierer bestimmt werden können, ob Geschwindigkeit oder Speicherbedarf in dem betreffenden Programmteil primär ist.

An dieser Stelle muß darauf aufmerksam gemacht werden, daß die geforderte Qualität des Produktes nicht allein durch den Einsatz oder die Neuentwicklung von Compiler-Optimierungsverfahren erreicht werden kann. Die Übergabe von Parametern bei einem Prozeduraufruf und die Verarbeitung von Parametern innerhalb eines Prozedurrumpfes stellt ein solches Beispiel dar. In Sprachen wie ALGOL 60 und PL/I tritt ein zusätzlicher Organisationsaufwand gegenüber der maschinenorientierten Programmierung zwangsweise auf, dem im maschinenorientierten Programm "die Verantwortung des Programmierers" entspricht.

Ein weiteres Beispiel ist die Verarbeitung von Daten aus Feldern oder Tabellen innerhalb von Schleifen. Der Einsatz von Indexregistern ist durch Optimierungen nur innerhalb einer Schleife möglich, deren Laufvariable sich aus der Sicht des Compilers linear verhält. Eine weitergehende Verbesserung der Qualität läßt sich nur durch Einführung von Adreßvariablen einschließlich einer Adreßarithmetik erreichen.

Der Begriff Problemorientierung, bezogen auf eine Programmiersprache, sollte in die Begriffe

- o Methodenorientierung und
- o Anwendungsorientierung

aufgespalten werden. Methodenorientierte Sprachen eignen sich besonders gut zur allgemeinen Beschreibung von Algorithmen. Beispiele hierfür sind FORTRAN, ALGOL 60, ALGOL 68, PL/I, CORAL und JOVIAL.

Als wesentliche Eigenschaften höherer Programmiersprachen werden angesehen:

- o Lesbarkeit,
- o Dokumentationswert,
- o Möglichkeit der Wartung und
- o Weitergabe von Erfahrungen.

Die letztgenannte Eigenschaft ersetzt die Forderung nach Austauschbarkeit von Programmen. Diese Abwandlung ist in dem oben definierten Aufgabenbereich sinnvoll, da beim Übergang von einem Rechner zu einem anderen im allgemeinen zwangsweise die Effizienz des erzeugbaren Maschinenprogrammes verschlechtert wird. Der Austausch von Programmen oder Programm-Moduln ist also nur in Verbindung mit Modifikationen des Quellenprogrammes sinnvoll.

Der Begriff Sicherheit besitzt im Bereich von Realzeit-Anwendungen besonders große Bedeutung. Als Komponenten werden betrachtet:

- o Funktion,
- o Wartung und
- o Termin.

Die Forderung nach funktioneller Sicherheit ist elementar. Unter Sicherheit der Wartung wird hier die Geschwindigkeit bzw. Reaktionszeit verstanden, in der ein aufgetretener Fehler analysiert und beseitigt werden kann. Als Termsicherheit wird hier die Einhaltung von Terminzusagen für die Fertigstellung von Programmpaketen und Programm-Modifikationen bezeichnet.

3. Konsequenzen

Für die Beurteilung eines Werkzeuges sind nicht nur seine Vorteile und Einsatzmöglichkeiten zu betrachten. Ebenso wichtig ist das Aufzeigen der Konsequenzen beim Einsatz des Instruments.

Bereits im Abschnitt 2 wurde indirekt festgestellt, daß die Qualität des erzeugbaren Maschinenprogramms nicht allein vom Compiler, sondern ebenso vom Programmierer, d.h. von der

Qualität des Quellenprogrammes

beeinflußt wird. Mit anderen Worten: aus einem schlechten Quellenprogramm entsteht ein schlechtes Maschinenprogramm. Dies soll selbstverständlich nicht bedeuten, daß von einem Compiler keinerlei Optimierungen gefordert werden.

Die oben aufgestellte Forderung bewirkt eine entsprechende nach der

Qualifikation des Programmierers.

Der Programmierer muß die Algorithmen, die er jeweils programmiert, weitgehend verstehen. Auch die Einordnung und die Bedeutung zum gesamten Produkt müssen ihm klar sein. Nicht zuletzt muß er die Struktur des jeweiligen Rechners kennen. Detailkenntnisse wie beispielsweise die Beherrschung von Programmiertricks auf der Maschinenebene sind jedoch nicht erforderlich.

Die bereits früher geforderte Einführung von Adreßvariablen und Adreßarithmetik zieht zwangsweise ein größeres Maß an

Verantwortung des Programmierers

nach sich. Dies bedeutet, daß eine Reihe von Prüfungen nicht mehr vom Compiler durchgeführt werden können. Wesentlich ist jedoch, daß der Programmierer sein Maß an Verantwortung den jeweiligen Forderungen des Problems individuell anpassen kann.

Mit anderen Worten: Der Programmierer kann selbst entscheiden, ob er beispielsweise Adreßvariable und Adreßarithmetik einsetzt oder die Prozedurparameterlisten selbst verwaltet.

Für die Entwicklung von umfangreichen Programmsystemen ist es empfehlenswert, zwischen einem

- o Software-Produktionssystem und einem
- o Laufzeitsystem

zu unterscheiden. Das Software-Produktions-System bietet alle Vorteile eines für kommerzielle Anwendungen entwickelten Betriebssystemes einschließlich der zugehörigen Rechnerperipherie. Dieses System dient der Entwicklung und dem Einzeltest der Programm-Moduln. Das Laufzeitsystem enthält im allgemeinen neben dem speziellen Realzeit-Betriebssystem einen wesentlich geringeren Komfort für Programmentwicklung und -test. Für beide Systeme können auch verschiedene Rechner eingesetzt werden. In diesem Fall muß die Programmiersprache zusätzliche Hilfsmittel bieten. Weiterhin müssen für die Simulation der Umgebung entsprechende Testsimulatoren hergestellt werden.

4. Problemsprache - Steuersprache

Wie bereits eingangs erwähnt, führten die geschilderten Überlegungen zur Entwicklung einer prozedurorientierten mittleren Programmiersprache. Diese erhielt den Namen POLYP (Problem Oriented Language for system software Programming). Die Namensgebung zeigt, daß zunächst als Anwendungsgebiet die Systemsoftware vorgesehen wurde.

Ausgehend vom Niveau höherer Programmiersprachen wurde nach Erweiterungen in Richtung maschinenorientierter Sprachen gesucht, durch die die Effizienz des erzeugbaren Maschinenprogramms verbessert werden kann. Als besonders wichtig wurden dabei

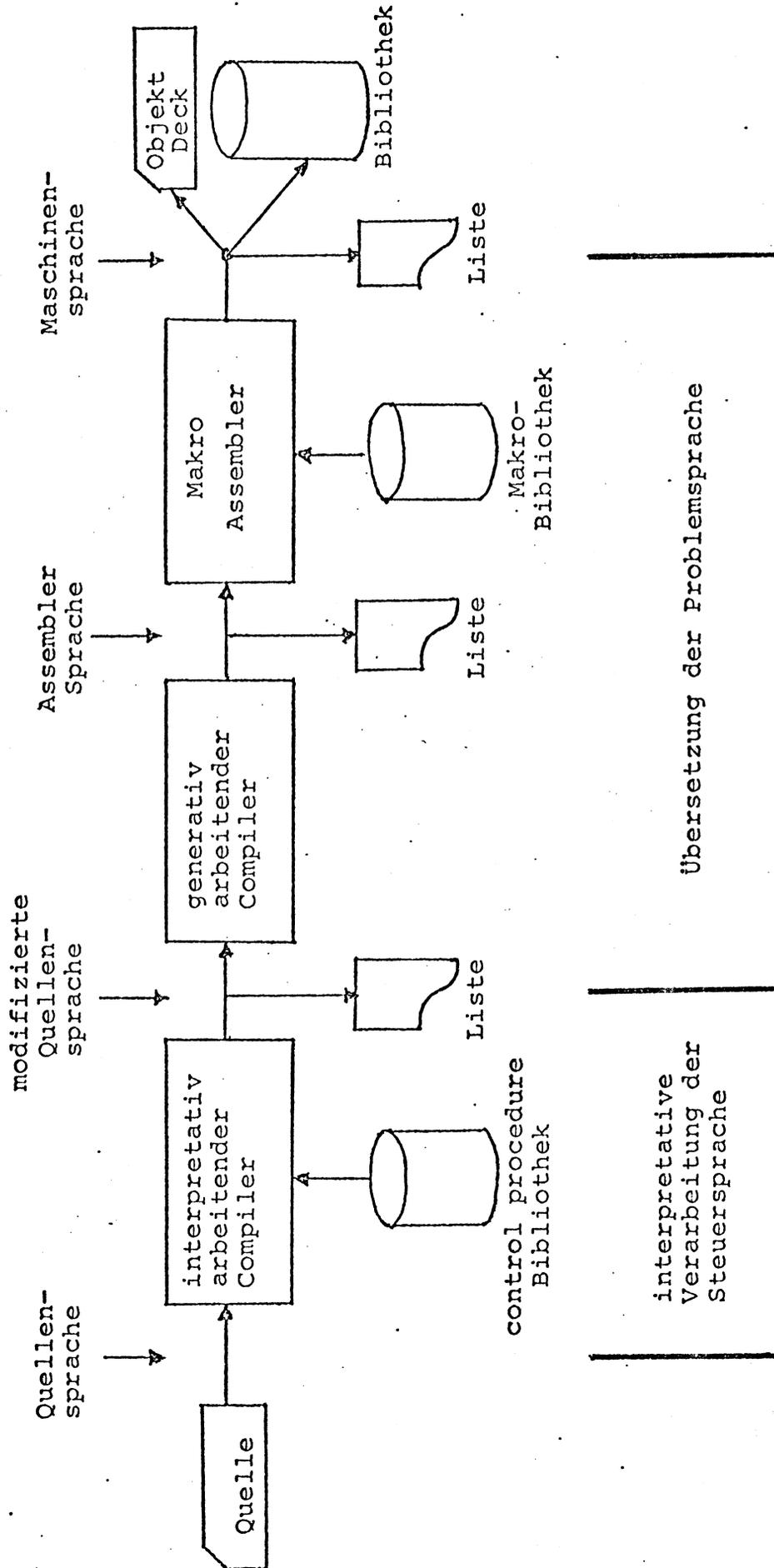
- o Adreßvariable einschließlich Adreßarithmetik
- o freie Definition von Informationslängen und die
- o Möglichkeit der relativen Positionierung von Informationen im Speicher

erkannt. Ergebnis war das Grobkonzept einer Sprache. Besonders schreibaufwendig waren dabei die Deklarationen.

Anschließend wurden die Hilfsmittel für maschinenorientierte Programmierung wie Makrogeneratoren und conditional assembly analysiert. Die Verallgemeinerung dieser Verfahren führte zu der Definition einer Steuersprache. Wird die oben erwähnte Sprache als Problemsprache bezeichnet, so wurde die Steuersprache so gewählt, daß diese einen subset der Problemsprache darstellt.

Die Semantik wurde ebenfalls sinngemäß übertragen. Die Definition einer control procedure entspricht der Definition eines Makros. Der Aufruf einer control procedure entspricht einem Makro-Aufruf. Die Übertragung von beispielsweise control variable, control assignment, control goto und control conditional goto ist elementar, falls man die entsprechenden Verfahren aus conditional assembly als Vorbild nimmt.

Daraus ergibt sich, daß der Compiler aus einem interpretativen Teil, der die Steuersprache verarbeitet, und einem generativen Teil besteht.

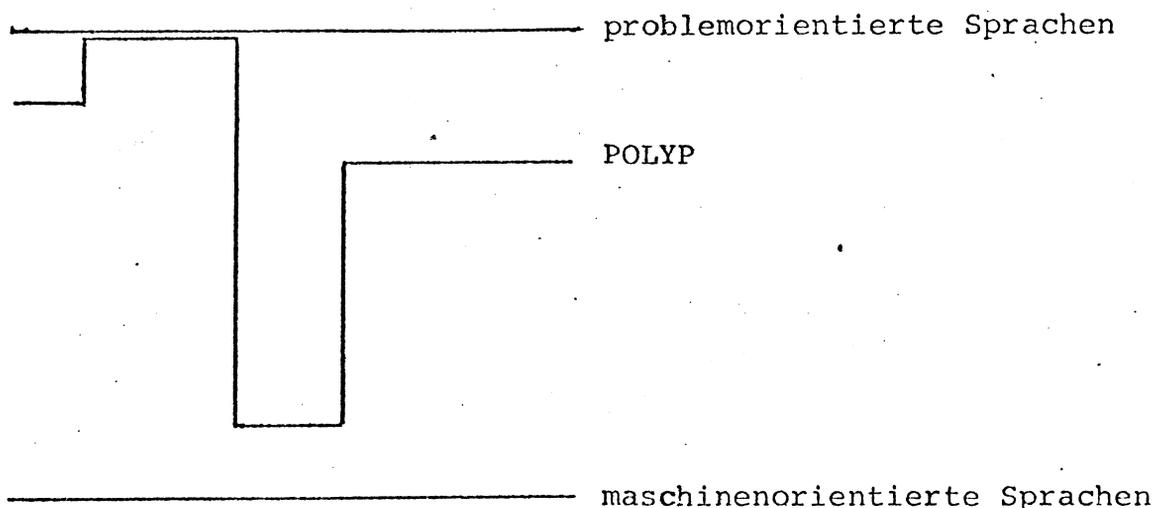


AUFBAU DES VOLLSTÄNDIGEN POLYP-COMPILERS

Die Steuersprache dient also der Modifikation der Quellsprache.

Durch dieses Vorgehen ist eine Modifikation der Sprache in Richtung der Anwendungsorientierung und abkürzender Schreibweisen ohne Eingriff in den Compiler möglich. Durch entsprechende Struktur des Prozeduraufbaues und Unterstützung des Compilers ist es durch ein spezielles Leseprogramm möglich, die Vorbesetzung der identifizier-Liste zu ändern oder zu ergänzen. Dies entspricht der Erweiterung der Sprache durch spezielle Standardfunktionen, Ein/Ausgabefunktionen und Betriebssystemaufrufen. Wesentlich hierbei ist, daß diese Erweiterungen wiederum ohne Eingriff in den Compiler möglich sind.

Zum Schluß soll noch kurz das Niveau von POLYP betrachtet werden. Auf Grund der bisherigen Ausführungen ist ohne weiteres klar, daß die einzelnen Sprachelemente ein sehr unterschiedliches Niveau besitzen. Während beispielsweise Prozeduren und vektorielle Zuweisungen sich auf der Ebene höherer Programmiersprachen befinden, ist die Adreßarithmetik fast auf dem Niveau maschinenorientierter Sprachen. Dieser Sachverhalt kann durch eine Zeichnung verdeutlicht werden:



Sprachen wie POLYP werden mit dem Namen GECOL (General Computer Oriented Language) belegt⁽¹⁾. Auf Grund dieser Dinge zählt POLYP zu den prozedurorientierten Sprachen und sollte nicht mit den Zwischensprachen verwechselt werden, die als wünschenswerte Schnittstelle zwischen maschinenunabhängigem und maschinenabhängigem Teil des generativen Compilers angestrebt werden.

(1) Siehe auch: SCS Schriftenreihe Heft 2,
Realtime Programmiersprachen

Bernd Eichenauer

Elektronik - System - GmbH

Entwicklungskonzept für ein Programmsystem
zur Erstellung und Erzeugung von
System - Software für die Prozeß - Automation

Inhalt

1. Überblick	- 2 -
2. System - Software für die Prozeß - Automation	- 3 -
3. Konzept eines erzeugenden Programmsystems	- 11 -
4. Referenzen	- 17 -

Kurzreferat, vorgetragen anlässlich der Fachgespräche über die
Förderung der Forschung und Entwicklung von prozedurorientierten
Prozeß - Programmiersprachen am 1. 10. 71 bei der GFK, Karlsruhe

1. Überblick.

Für die kostenwirksame Lösung von Prozeß - Automationsaufgaben werden umfangreiche Programmsysteme benötigt, die aus Zeit- und Kostengründen nicht für jeden zum Einsatz kommenden Rechnertyp eigens erstellt werden können. Um trotzdem die zur effektiven Lösung von Prozeß - Automationsvorhaben erforderliche System - Software für die jeweils zum Einsatz kommenden Rechnertypen bereitstellen zu können, bedarf es einer Implementierungsmethode, die es gestattet,

- o die Systemprogramme möglichst weitgehend in rechnerunabhängiger Form zu erstellen und
- o mit möglichst geringem Anpassungsaufwand auf den verschiedenen Rechenanlagen lauffähig zu machen.

In dem folgenden Referat wird zunächst die zur kostenwirksamen Lösung von Prozeß - Automationsvorhaben erforderliche System - Software umrissen und dann ein Programmsystem skizziert, mit dem sich die oben genannten Forderungen erfüllen lassen.

2. System - Software für die Prozeß - Automation.

Die zur Lösung einer Prozeß - Automationsaufgabe erforderliche System - Software soll im folgenden anhand des in Abb. 1 dargestellten vereinfachten Ablaufs eines Automationsvorhabens erläutert werden. Im Hinblick auf das Ziel dieses Abschnitts braucht dabei auf die Rückkopplung zwischen den einzelnen Lösungsschritten nicht eingegangen werden.

Die Abwicklung eines Automationsvorhabens lässt sich grob gesehen in die folgenden 5 Teilschritte gliedern:

- o Systemanalyse
- o Systemdefinition
- o Erstellen des Automationsprogramms
- o Erstellen, Aufstellen und Integrieren der Automationsmittel
- o Einfahren des Prozesses.

Bei der Analyse einer Prozeß - Automationsaufgabe muß zuerst die Prozeßanalyse durchgeführt werden. Dabei sind unter anderem folgende Fragenkomplexe zu beantworten:

- o Grobstruktur des Prozesses:
 - . Wie untergliedert sich der Prozeß in Teilprozesse bzw. wie kann der Prozeß in Teilprozesse untergliedert werden?
- o Ablaufanalyse und Prioritätsstruktur:
 - . Unter welchen Bedingungen werden die Teilprozesse gestartet?
 - . Welche Reaktionszeiten sind nach Eintritt der Startbedingungen zulässig?
 - . Wie laufen die Teilprozesse in der Zeit ab?
 - . Wie bedingen sich die Teilprozesse gegenseitig?
 - . In welcher Reihenfolge müssen die Teilprozesse ablaufen?

o Charakteristische Prozeßgrößen:

- . Über welche charakteristischen Prozeßgrößen kann man hinreichend genauen Aufschluß über den Prozeßzustand gewinnen (Definition von Meßgrößen, Signalen, u.s.w.)?
- . Über welche charakteristischen Prozeßgrößen kann man den Prozeßablauf beeinflussen (Definition von Stellgrößen, Stimulifunktionen, u.s.w.)?
- . Wie hängen die charakteristischen Prozeßgrößen voneinander ab?
- . Wie genau müssen die charakteristischen Prozeßgrößen gemessen bzw. eingestellt werden und in welchem Wertebereich variieren sie?

o Sicherheit und Wiederanlauf:

- . Darf der Prozeßablauf unterbrochen werden und wenn ja, wie groß muß die MTBF sein?
- . Welche Meßgrößen bzw. Stellwerte müssen redundant erfassbar bzw. einstellbar sein?
- . Welche Wiederanlaufspunkte sind zulässig?

o Eingriff und Kommunikation:

- . Welche Eingriffsmöglichkeiten in den Prozeßablauf sind für das Bedienungs- und Wartungspersonal vorzusehen?
- . Welche Informationen über das Prozeßgeschehen müssen zur Überwachung und zur Dokumentation des Prozeßablaufs bereitgestellt werden?

Nach und auch teilweise schon parallel zu der Prozeßanalyse kann die Festlegung der Automationsverfahren erfolgen. Dabei sind unter anderem Methoden und Modellvorstellungen zu erarbeiten, die das Wechselspiel zwischen den Prozeß - Ein/Ausgaben regeln (Steuerungs- und Regelungsverfahren). Nach Festlegung der

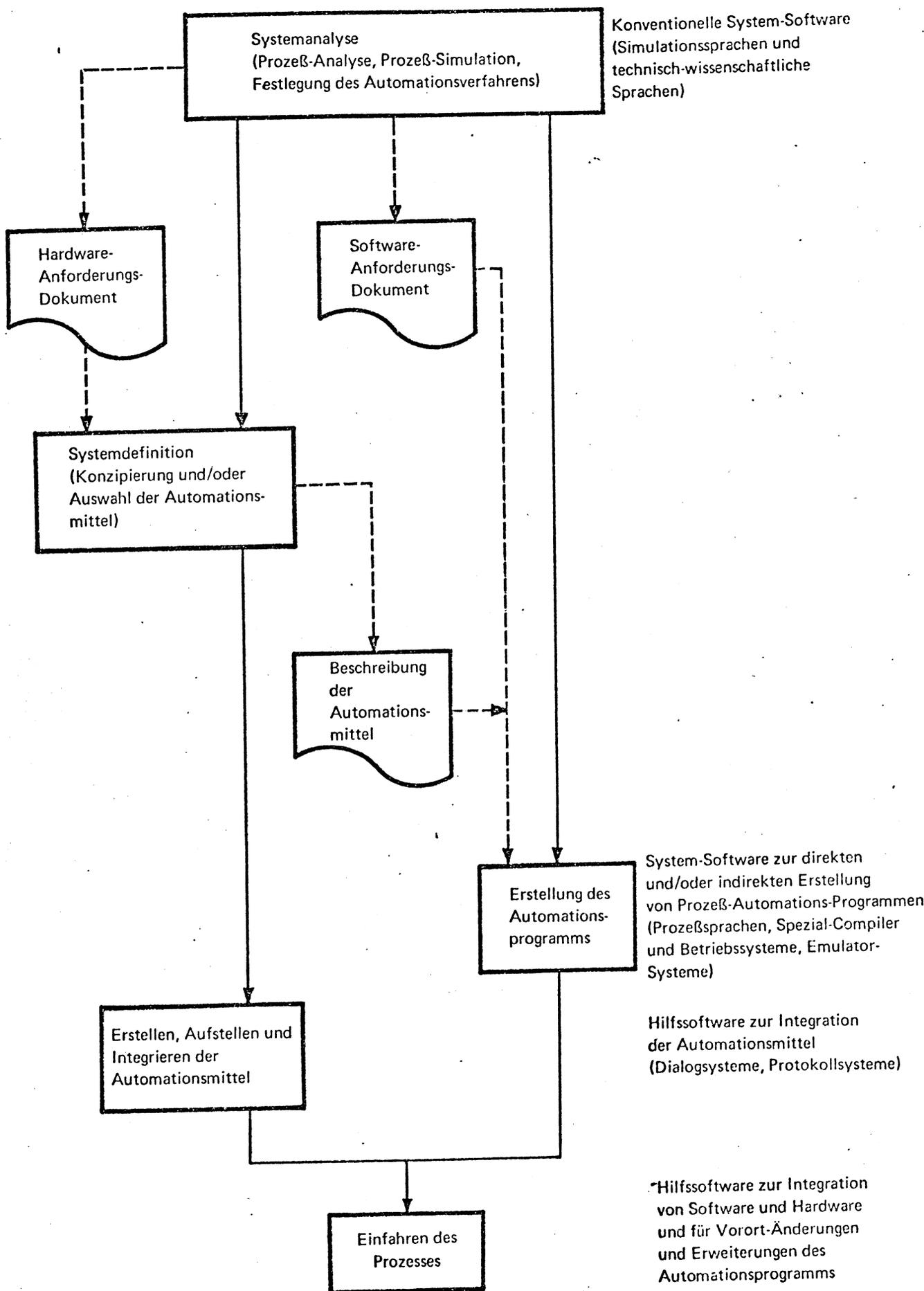


Abb. 1 Vereinfachte Darstellung des Ablaufs eines Automations-Vorhabens

Automationsverfahren für die Teilprozesse muß entschieden werden, welche Verfahrensteile durch Hardware - Komponenten realisiert werden müssen und welche Teile der Automationsverfahren im Rahmen des Automationsprogramms verwirklicht werden können (z. B. Regelkreise oder DDC).

Bei der Festlegung des Automationsverfahrens für größere Prozesse werden Simulationsprogramme in Simulations Sprachen (z. B. SL-1, CSSP) und/oder technisch-wissenschaftlichen Programmiersprachen erstellt, mit denen insbesondere die Modellvorstellungen über den Zusammenhang zwischen den charakteristischen Prozeßgrößen überprüft werden. Teile dieser Simulationsprogramme sind bei dem indirekten Test der Automationsprogramme in Form von Prozeß - Simulationsroutinen erneut einsetzbar.

Die Ergebnisse der Systemanalyse münden normalerweise in zwei Dokumenten, in denen die Anforderungen an die Automationsmittel und an das Automationsprogramm spezifiziert sind. In Einzelfällen, z. B. bei der Definition von Prüfvorschriften für Elektronik - bausteine, die mit schon realisierten rechnergesteuerten Prüfanlagen getestet werden sollen, können beide Dokumente durch ein Testprogramm ersetzt werden, das in einer geeigneten höheren Programmiersprache (z. B. ATLAS) erstellt wird.

Aufgrund der Angaben in dem Hardware - Anforderungs - Dokument kann die Konzipierung und/oder Auswahl der Automationsmittel (Prozeßrechner, Standard- und Prozeß - Peripherie, Prozeß - Endstellen, Bedienfelder, u. s. w.) erfolgen. Parallel dazu kann schon mit der Erstellung des Automationsprogramms begonnen werden. Dabei werden in zunehmendem Maße prozeßorientierte Programmiersprachen eingesetzt (z. B. INDAC8 [1]).

Eine wesentliche Vereinfachung der Prozeß - Programmierung darf man nach Implementierung von PEARL [2] erwarten.

Bekanntlich besteht ein PEARL - Programm aus einem Systemteil und einem Problemteil. Der Problemteil eines PEARL - Programms kann, da er weitgehend unabhängig von der Konfiguration und den Eigenschaften der Automationsmittel ist, im wesentlichen aufgrund der Angaben in dem Software - Anforderungs - Dokument zeitlich parallel zur Systemdefinition erstellt werden. Nach Konzipierung und/oder Auswahl der Automationsmittel wird das Automationsprogramm durch den Systemteil vervollständigt.

Außer dieser für den Ablauf eines Automationsvorhabens vorteilhaften Programmaufteilung werden in PEARL auch die sprachlichen Hilfsmittel bereitgestellt, mit denen sich die Ergebnisse der Systemanalyse und der Systemdefinition problemangemessen darstellen lassen. Beispielsweise

- o lassen sich die Automationsprogramme für die Teilprozesse als Tasks und Subtasks definieren.
- o Kurze Reaktionszeiten lassen sich durch Anhängen von Anweisungsfolgen an das Interruptsystem erreichen (On Condition).
- o Der Start von Automationsprogrammen kann über programmierbare Interrupts (Signal) und durch Aktivieren von Tasks veranlasst werden.
- o Die zeitliche und logische Synchronisation des Programm- und Prozeßgeschehens wird durch SUSPEND - Anweisungen und durch Semaphore - Operationen ermöglicht.
- o Die Reihenfolge, in der die Automationsprogramme für die Teilprozesse ablaufen müssen, kann durch Prioritätszuweisungen an Tasks definiert werden.
- o Für die Formulierung von Automationsalgorithmen (Eich- bzw.

Kalibrierungsfunktionen, Kontrollalgorithmen, Regelungsalgorithmen, u. s. w.) sind die aus technisch - wissenschaftlichen Programm-sprachen bekannten Sprachelemente einsetzbar.

- o Die zulässigen Wiederanlaufspunkte lassen sich in einer Steuer- oder Führungstask festlegen. u.s.w.

Um die Übersetzung und den Test der Automationsprogramme parallel zum Erstellen, Aufstellen und Integrieren der Automationsmittel und damit das Einfahren des Prozesses mit weitgehend ausgetesteten Programmen durchführen zu können, werden neuerdings (insbesondere in den USA) häufig Simulatoren oder Emulatoren eingesetzt. (z. B. [3 - 5]) Mit den bisher realisierten einfachen Emulator - Systemen kann ein in Assembler vorliegendes Programm auf einem Großrechner entweder direkt oder durch interpretatives Ausführen des Assemblers für den Zielrechner in dessen Code übersetzt werden. In Assembler oder Objektform vorliegende Automationsprogramme lassen sich bei Simulation der Prozeßreaktionen im gedehnten Zeitmaßstab interpretativ ausführen.

Zur Integration der Automationsmittel werden Dialog - und Protokollsysteme benötigt, mit denen das Integrationsteam die Funktionsfähigkeit und das Zusammenspiel der Automationsmittel testen kann. Beispielsweise müssen sich über ein Bediengerät Stellwerte und Kanäle zur Ausgabe von Stellwerten spezifizieren lassen. Nach Definition zugeordneter Eingangskanäle kann die Reaktion auf die Ausgaben gemessen und protokolliert werden. Durch Vergleich der protokollierten Meßwerte mit den physikalischen Prozeßwerten können z. B. Eichparameter ermittelt werden. Für Integrationsarbeiten haben sich dialogfähige Versionen der FORTRAN - Erweiterung BASIC bewährt.

Nach Integration der Automationsmittel und Erstellen des Automationsprogramms kann mit dem Einfahren des Prozesses begonnen werden. Dabei werden in der Regel die kritischen Teile der Automationsmittel und des Automationsprogramms zunächst anhand von Modellprozessen im Labor erprobt. Für diese Arbeiten werden außer den schon früher erwähnten Dialog- und Protokollsystemen für die Hardware - Integration auch Dialogsysteme benötigt, mit denen die Ausführung des Automationsprogramms und die Protokollierung der Programmgrößen und des Programmablaufs gesteuert werden kann. Erst nach erfolgreichem Einfahren der Laborprozesse kann mit dem Einfahren des realen Prozesses begonnen werden.

Zur kostenwirksamen Lösung von Prozeß - Automationsaufgaben sind nach dem Vorangegangenen im wesentlichen folgende Software - Komponenten wünschenswert:

1. Programmiersprachen

- o Simulationssprachen und technisch - wissenschaftliche Sprachen
- o Eine allgemeine Prozeßsprache zur Formulierung beliebiger Prozeß - Automationsprogramme (PEARL)
- o Spezialsprachen für häufig zu bearbeitende Teilgebiete der Prozeß - Automation (z. B. ATLAS für Check - out - Aufgaben, BASIC für Integrationsarbeiten)

2. Dialogfähige Übersetzungsprogramme für die zum Einsatz kommenden Sprachen und Rechenanlagen

3. Betriebssysteme

- o Invariante Grundsysteme
- o Sprachorientierte Systeme
- o Bedienungssysteme

- o Protokollsysteme

4. Emulatorsysteme

- o für den indirekten Test von Automationsprogrammen im gedehnten (bitgenau) und im physikalischen Zeitmaßstab (funktionell)
- o bei Simulation bzw. Teilsimulation des Prozeßgeschehens und der Automationsmittel

5. Dienstprogramme z. B. zur Erweiterung bzw. Reduzierung des Gesamtprogramms um in sich abgeschlossene Teilprogramme ohne Störung des Ablaufs der übrigen Teilprogramme (Laborautomation)

3. Konzept eines erzeugenden Programmsystems.

Die in Abschnitt 2 skizzierte System - Software zur kostenwirksamen Lösung von Prozeß - Automationsaufgaben ist in dem dargestellten Umfang bisher noch für keine Rechenanlage verfügbar gemacht worden. Dafür gibt es im wesentlichen zwei Gründe:

- o Die Spezifikationen für diese Software sind bis heute nicht vollständig erarbeitet worden.
- o Bei Einsatz der z. Z. gängigen Implementierungsmethoden können Systemprogramme nur speziell für jeweils ein Rechnermodell bzw. eine Rechnerfamilie erstellt werden. Das durch die hohen Entwicklungskosten hervorgerufene Risiko ist - insbesondere bei kleinen und mittelgroßen Anlagen - sehr groß.

Um beiden Schwierigkeiten zu begegnen, bedarf es eines Implementierungskonzepts, bei dessen Anwendung

- o einerseits eine schnelle Erstellung und Modifikation von System - Programmen möglich wird,
- o andererseits der wesentliche Teil der System - Software nur einmal erstellt zu werden braucht und dann mit möglichst geringem Anpassungsaufwand auf unterschiedlich strukturierte Rechenanlagen überführt werden kann.

Um dieses Ziel zu erreichen, kann man von folgenden System - Komponenten ausgehen:

1. einer niederen System - Programmiersprache (im folgenden kurz mit SYSLAN bezeichnet), mit der die
 - o Off - line - Software (z. B. Compiler - Generator, Emulator - Generator, Betriebssystem - Composer, u. s. w.) und alle nicht laufzeitintensiven Systemdienste (z. B. Bedienungssystem, Teile der Taskorganisation, u. s. w.) in rechner-

unabhängiger Form und

o alle laufzeitintensiven Systemdienste (z. B. Ablauforganisation, Interruptprogramm, u. s. w.) in einer auf den jeweiligen Zielrechner zugeschnittenen Weise auf formalem Niveau erstellt werden können.

2. einem in SYSLAN geschriebenen Metacompiler, der sich mit möglichst geringem Aufwand an den Code unterschiedlicher Rechnermodelle anpassen lässt.
3. einem Lotsen - Compiler, mit dem SYSLAN - Programme in den Code einer größeren Rechenanlage übersetzt werden können. Der Lotse für SYSLAN kann in einer höheren Programmiersprache geschrieben werden und braucht keinen effektiven Code zu erzeugen. Er wird bei Generierung des Ursystems ein einziges Mal benötigt.

Die Erstellung des Urcompilers für SYSLAN geht in zwei Schritten vor sich:

1. Übersetzen des SYSLAN - Metacompilerprogramms mit dem Lotsen in einen Zwischencompiler. Wegen der oben erwähnten Eigenschaft des Lotsen handelt es sich bei dem Zwischencompiler um ein langsam laufendes Programm. Da in dem Metacompilerprogramm die erforderlichen Optimierungsverfahren realisiert sind, erzeugt der Zwischencompiler effektiven Code.
2. Übersetzen des Metacompilerprogramms mit dem Zwischencompiler in den angestrebten SYSLAN - Compiler.

Durch Boot - strapping, d. h. Übersetzen aller in SYSLAN erstellten System - Programme mit dem an verschiedene Rechenanlagen angepassten Urcompiler kann man die System - Software für diese Rechenanlagen erzeugen. u. s. w.

Um auch die höheren Programmiersprachen für die Prozeß - Automation mit möglichst geringem Aufwand implementieren zu können, ist es zweckmäßig, bei der Konzipierung des Metacompilers auf Sprachenunabhängigkeit der Listenstrukturen sowie der Analyseprogramme zu achten und den Metacompiler durch einige Hilfsprogramme zu einem Compiler - Generator zu erweitern. Die Methoden dafür wurden in den letzten Jahren entwickelt und erprobt (siehe z. B. [6 - 9]).

In Abb. 2 ist die Struktur eines Metacompilers dargestellt, der gegenüber der Literatur um die zur Übersetzung von prozeß - orientierten Programmiersprachen erforderlichen Komponenten erweitert wurde. Aus Zeitgründen kann hier nur auf die wesentlichste Struktureigenschaft des Metacompilers eingegangen werden.

Wie aus der Abbildung ersichtlich ist, werden alle Quellenprogramme zunächst in eine compilerinterne Zwischensprache übersetzt.

Dadurch wird erreicht, daß

- o die Implementierung einer Programmiersprache bis auf Teile des sprachorientierten Betriebssystems nur ein einziges Mal auszuführen ist und daß
- o der codeerzeugende Compilerteil für jede Rechenanlage nur einmal erstellt werden muß.

Die Zwischensprache ist der Assembler oder Code eines virtuellen Rechenautomaten, dessen Struktur so gewählt werden muß, daß er

- o einerseits von der Quellsprache her mit vertretbarem Aufwand erreicht werden kann, daß
- o andererseits aber auch effektiven Code erzeugende Zwischensprachen - Übersetzer (Codeerzeugung) für heutige und in Zukunft zu erwartende Digitalrechner mit möglichst geringem Aufwand erstellt werden können.

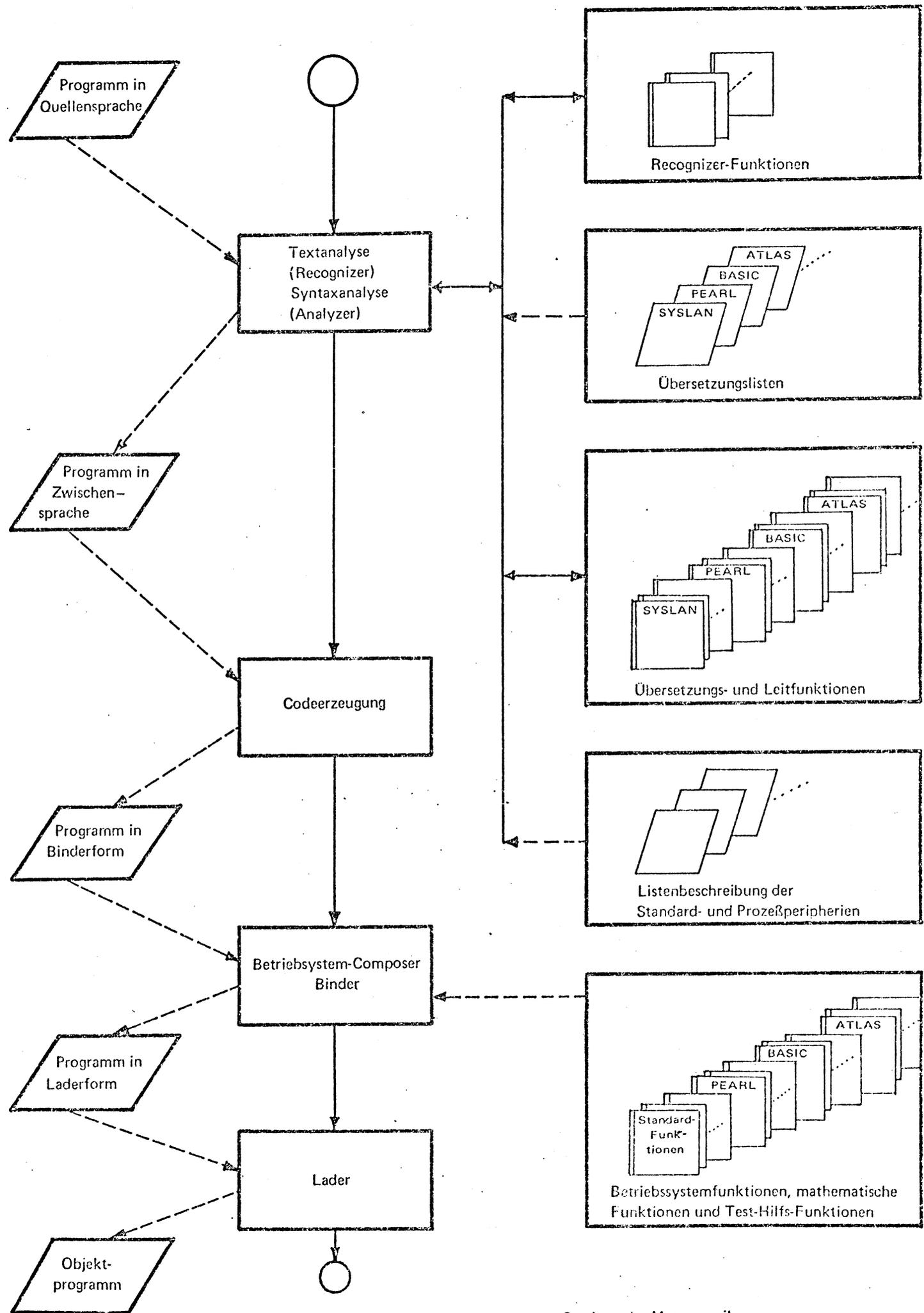


Abb. Struktur des Metacompilers

In den bisher erstellten compilererzeugenden Programmsystemen für technisch - wissenschaftliche und kommerzielle Programm-sprachen wird als Zwischensprache meist der Assembler oder Code eines virtuellen Kellerautomaten (siehe z. B. [10 - 12]) verwendet. Um die oben genannten Anforderungen an die Zwischen-sprache auch bei compilererzeugenden Programmsystemen für prozeßorientierte Programmiersprachen gut erfüllen zu können, sind die bisher konzipierten Kellerautomaten in ihrer Struktur und in ihrem Befehlsvorrat zu modifizieren bzw. zu erweitern. Beispielsweise sollten die Grundfunktionen der Taskorganisation in der Struktur realisiert und über Befehle direkt ansprechbar sein.

In Abb. 2 wurde von den heute verfügbaren Zwischensprachen ausgegangen. Da in diesem Fall gelegentlich Nachoptimierungen laufzeitintensiver Betriebssystemteile erforderlich sein dürften, wurde die Ankopplung der zur Ausführung des jeweiligen Automationsprogramms erforderlichen Betriebssystemfunktionen erst nach der Codeerzeugung vorgesehen.

Eine weitere wichtige System - Komponente, auf die noch kurz eingegangen werden soll, ist das früher erwähnte Emulator - System. Auch hier kann man durch eine zweckmäßige System - Strukturierung erreichen, daß große Teile des Systems unabhängig von der Struktur der zu emulierenden Rechenanlagen und unabhängig von der Art der zu simulierenden Prozesse erstellt werden können (Metaemulator). Durch Hilfsprogramme z. B. zur Verwaltung und zum Ankoppeln der Rechnerbeschreibungen (Befehls - Interpretations - Routinen, Registerstrukturen) sowie von Prozeß - und Peripherie - Simulations - Routinen kann ein Emulator - Generator aufgebaut werden. In Abb. 3 ist der Aufbau eines Metaemulators für die bitgenaue Emulation vereinfacht dargestellt.

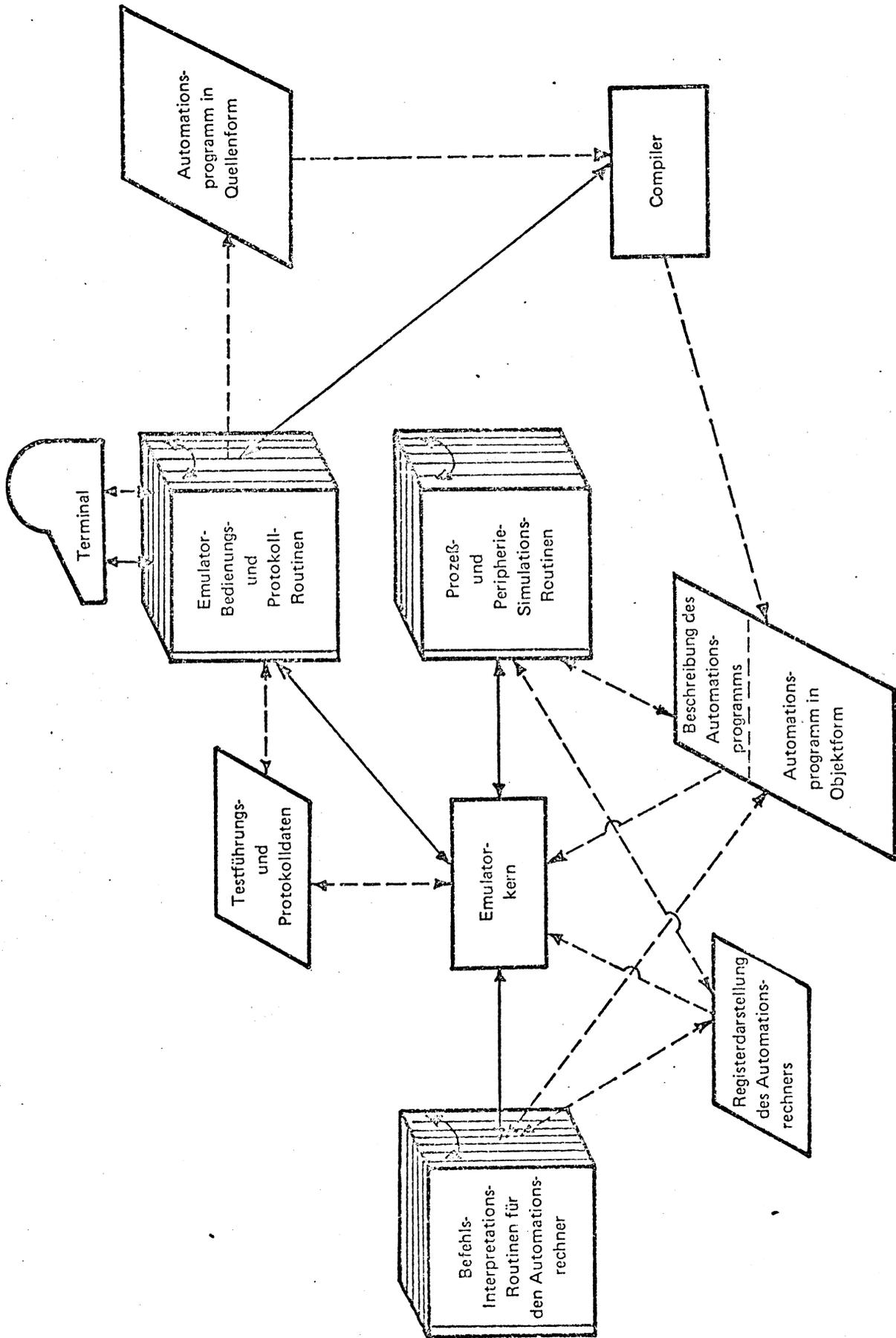


Abb. 3 Struktur des Metaemulators für Einprozessoranlagen

Referenzen

- [1] INDAC8, Digital Equipment Corporation
- [2] J. Brandes et al: PEARL, The Concept of a Process -
and Experiment - oriented Programming Language,
Elektron. Datenverarb. 12, H. 10 (1970), 429 - 442
- [3] The ALPHA MIMIC Simulator - Loader - Preparer, Control -
Data - Corporation
- [4] The HCM - 230 Simulation System, Hughes Aircraft Corporation
- [5] The FOCUS Simulator, Singer - General Precision Inc.,
Kearfott Division
- [6] J. Feldman und W. D. Gries: Translator Writing Systems,
Comm. ACM, 11, No. 2 (1968), 77 - 113
- [7] P. Gilbert und W. D. McLellan: Compiler Generation using
formal specifications of procedure - oriented and machine
languages, Spring Joint Comp. Conf. 1967, 447 - 455
- [8] W. M. McKeemann et al: The XPL compiler generator system,
Fall Joint Comp. Conf. 1968, 617 - 635
- [9] A. Kley: Ein compilererzeugendes Programmsystem,
Wiss. Ber. AEG - Telefunken 43, (1970), 1
- [10] B. Randell und L. S. Russel: Algol 60 Implementation,
Academic Press, London (1964)
- [11] E. A. Hauck und B. A. Dent: Borrough's B6500/B7500 stack
mechanism, Spring Joint Comp. Conf. 1968, AFIPS Conf.
Proc. 32, 245 - 251
- [12] J. Swoboda, Sprachorientierte Rechner - Probleme der
Adressierung, Elektron. Rechenanlagen 12, H. 1 (1970),
26 - 35

Johannes Reh,
Elektronik-System-GmbH

Grundsatzarbeiten der ESG zur Prozeßlenkung

mit DV - Anlagen

und

Vorschläge förderungswürdiger Fragestellungen.

Kurzreferat am 1.10.1971 anlässlich der
Diskussions- und Anhörungsveranstaltungen
zum 2.DV-Förderungsprogramm bei der
Gesellschaft für Kernforschung mbH,
Karlsruhe

Inhaltsübersicht

1. Grundsatzarbeiten der ESG zur Prozeß-
lenkung mit DV-Anlagen
 - 1.1 Aufgaben-Dualismus in den Fachabteilungen
zwischen Projekt- und Grundsatz-Arbeiten
 - 1.2 Randbedingungen und Optimierungskriterien
 - 1.3 Vorstellungen der ESG über eine Zentralstelle
für Prozeßrechentechnik; Einsatzgebiete
2. Förderungswürdige Fragestellungen;
Themenvorschläge der ESG
 - 2.1 Offene Fragen; Themenkreise
 - 2.2 Modell für eine Zentralstelle für Prozeß-
rechentechnik; ERZ-Modell
 - 2.3 Entwicklungskonzept der ESG für ein Programm-
system zur Erstellung und Generierung von
Systemsoftware für Prozeßrechner

1. Grundsatzarbeiten der ESG zur Prozeßlenkung mit DV-Anlagen

1.1 Aufgaben-Dualismus in den Fachabteilungen zwischen Projekt- und Grundsatzarbeiten

In einer Mehrprojekt-Firma stellt sich den Fachabteilungen täglich neu die Frage eines sinnvollen Kompromisses zwischen projekt-bezogener und projekt-unabhängiger Facharbeit.

Über die Gesamtheit der anstehenden und angestrebten Projekte hinweg ist eine Optimierung zu betreiben mit dem Ziel, einen möglichst hohen Anteil an Aktivitäten so zu gestalten, daß er für alle Projekte gemeinsam eingesetzt werden kann (- unter Beibehaltung eines möglichst geringen Anteils an projekt-spezifischen Aktivitäten).

Es ist allgemein bekannt, daß der angestrebte Kompromiß nur zu erreichen ist durch

- Standardisierung der Methoden und Verfahren
- Verlagerung von Aktivitäten aus riskanten und teuren Bereichen
- Bereitstellung universell verfügbarer Anlagen und Einrichtungen.

1.2 Randbedingungen und Optimierungskriterien

Selbstverständlich kann sich die angestrebte Optimierung nur im Rahmen einschränkender Randbedingungen bewegen:

1) Jedes der anstehenden Projekte ist innerhalb der generellen Randbedingungen

- Kostenwirksamkeit
- Einhaltung von Termin- und Kosten-Limits
- Minimalisierung des Risikos

abzuwickeln.

2) Jedes der anstehenden Projekte ist so anwendungsorientiert und benutzer-freundlich wie nur möglich voranzutreiben.

3) Zu den generellen Randbedingungen, unter welchen über alle Projekte hinweg zu optimieren ist, kommen für die Fachabteilung Informationsverarbeitung noch die folgenden Zusatzbedingungen:

- Vielfalt der Programmiersprachen:

Zur Formulierung der Prozeßautomations-Aufgaben werden in zunehmendem Maße höhere Programmiersprachen verwendet. Diese Sprachen entstehen zu meist innerhalb gewisser Anwendergruppen und bilden, jede für sich, ein für die jeweiligen Anwender optimal benutzer-freundliches Hilfsmittel. Da die Anwender auf dessen Einsatz bestehen, ist in der nächsten Zeit mit einer Vielfalt von Programmiersprachen zu rechnen; ungeachtet der starken Tendenzen zur Entwicklung einer einzigen, generell geeigneten Sprache für alle Arten der Prozeßautomation, wie man sie eigentlich als Standard für alle Projekte haben möchte.

- Vielfalt der Rechnerstrukturen

Auch hier ist in absehbarer Zeit keine Standardstruktur der zum Einsatz kommenden Prozeßrechner zu erwarten. Selbstverständlich gibt es Stan-

dardisierungsbestrebungen auch hier, sei es in Einzelfragen wie Wortlänge, Grund-Befehlsrepertoire oder Kompatibilitäten zu Rechnern anderer Größenordnung. Tatsächlich gilt jedoch für heute und wohl auch noch für die nächsten Jahre, daß sogar in ein und demselben Elektroniksystem Prozeßrechner unterschiedlicher Struktur gemeinsam auftreten.

- Mangelnde oder gänzlich fehlende Verfügbarkeit und Funktionsfähigkeit der System-Hardware;

Dies ist ebenfalls eine typische, auch in Zukunft vorliegende Situation bezüglich der zu programmierenden Prozeßrechner und der zu integrierenden Systemkomponenten. Mangel an Verfügbarkeit äußert sich zum Beispiel primär im Fehlen ausreichender Bedienbarkeit und entsprechender Systemsoftware, während Funktionsunfähigkeit zum Beispiel durch grundsätzliches Nichtfunktionieren eines Sensors außerhalb seiner natürlichen Umgebung bedingt sein kann.

1.3

Vorstellungen der ESG über eine Zentralstelle für Prozeßrechentechnik; Einsatzgebiete

In der ESG werden zur Zeit Vorstellungen ausgearbeitet, wie innerhalb dieser doch erheblich einschränkenden Randbedingungen eine Optimierung im angestrebten Sinn durchgeführt werden kann. Diese Vorstellungen über den zukünftigen Weg kreisen um den Aufbau einer Zentralstelle für Prozeßrechentechnik im Sinne einer universellen Einrichtung zur Steuerung der in den einzelnen Phasen eines oder mehrerer Projekte anfallenden Aktivitäten.

Es liegt bei den geschilderten Randbedingungen auf der Hand, maximale Unabhängigkeit von echten Systemkomponenten im Verein mit maximaler Beherrschbarkeit der Systemfunktionen durch Simulation anzustreben.

Dementsprechend denkt die ESG an den Aufbau eines Rechenzentrums zur System-Entwicklung und -Integration, auch "Emulationsrechenzentrum" genannt, welches einerseits dem Benutzer die volle Bedienungseffektivität eines konventionellen Rechenzentrums verfügbar machen soll, andererseits jedoch aufgrund seiner speziellen Labororientierung den beliebigen Übergang von voll simulierten zu voll realisierten Systemfunktionen und Systemkomponenten ermöglichen soll.

Dem Projektablauf entsprechend ergeben sich für eine Zentralstelle für Prozeßrechentechnik die folgenden Einsatzgebiete:

1. Indirekte Erstellung der Prozeßrechner-Software nach der "Emulationsmethode"
 - 1.1 Bitgenaue Emulation der Prozeßrechner
 - 1.2 Funktionelle Simulation
2. Vorintegration der Software durch Systemabbildung
 - 2.1 In pseudo-Realtime
 - 2.2 In Realtime, mit Anschluß kritischer Geräte und "man-in-loop"
3. Rechner-gestützte/geführte Systemintegration und Systemprüfung
 - 3.1 Überwachung und Bewertung der Systemfunktionen
 - 3.2 Ersetzung nicht verfügbarer Systemkomponenten
4. Rechner-gestützte/geführte Inbetriebnahme, "Einfahren des Prozesses"
 - 4.1 Planung und Vor-Simulation der Anfahr-Strategie
 - 4.2 Auswertung von Echtdateien.
5. Rechner-gestützte/geführte Ausbildung des in den Prozeß einbezogenen Menschen - Trainer Simulator
 - 5.1 Normalverhalten
 - 5.2 Kritische Zonen.

Beim Aufbau einer derartigen Stelle ist eine Modularität "quer über alle Einsatzgebiete" in dem Sinne anzustreben, daß beim Übergang von einem Einsatzgebiet zu einem anderen möglichst viel von den schon erstellten Komponenten in Hardware und Software übernehmbar ist.

Selbstverständlich sind damit nur die Prozeßsystem-orientierten Einsatzgebiete umrissen. Bei entsprechender Betriebsorganisation spricht nichts dagegen, auch konventionelle Aktivitäten wie etwa Batch oder interaktives Timesharing über die Zentralstelle abzuwickeln.

2. Förderungswürdige Fragestellungen;
Themenvorschläge der ESG

2.1 Offene Fragen: Themenkreise

Bei den Konfigurationsüberlegungen für eine Zentralstelle mit den beschriebenen Einsatzgebieten ergibt sich eine Fülle von Problemstellungen und Themenkreisen, die nach Meinung der ESG durchaus eine Diskussion unter dem Aspekt der Förderungswürdigkeit im Rahmen des 2. DV-Programms wert sind.

Offene Fragen bestehen u.a. bezüglich

- Einsatzphilosophie, Leistungsanforderungen und Konfiguration der Zentralstelle
- Konventionelle Komponenten in Hardware und Software
- Interface zu den Laborsystemen der zu bedienenden Projekte
- Spezielle Systemsoftware für die Zentralstelle

Die angeschnittenen Fragen korrespondieren eng mit allen Themenkreisen des Vorhabens "Prozeßlenkung mit DV-Anlagen". Schließlich ist ja die Konzipierung und Realisierung eines Emulationsrechenzentrums selbst ein Projekt, in welchem Prozeßlenkung mit DV-Anlagen betrieben wird.

2.2

Modell für eine Zentralstelle für Prozeßrechenstechnik

Die ESG stellt daher die Frage zur Diskussion, inwieweit es als praktikabel empfunden wird, einen Teil der hier auszuwählenden förderungswürdigen Vorhaben anhand der Realisierung eines Modells einer entsprechenden Zentralstelle zu bearbeiten ("ERZ-Modell", d.h. Modell eines "Emulationsrechenzentrums").

Dies böte den nicht zu unterschätzenden Vorteil eines sach- und anwendungsgerechten Verfahrens-Rahmens, in welchem eine Vielzahl weiterer Einzelvorhaben abgewickelt werden könnte.

Zugleich böte sich hier die Chance eines Demonstrativ-Vorhabens zur Einführung einer zukunfts- und richtungweisenden Methodik, die unseres Wissens in der beabsichtigten vollen Breite noch nirgends realisiert worden ist.

Wir sehen uns in dieser Meinung nicht zuletzt durch unsere weltweiten Recherchen auf dem speziellen Prozeßrechengebiet der AVIONIK bestätigt: Überall, insbesondere auch bei den entsprechenden Firmen und Forschungsstellen in den USA, haben sich ähnliche Vorstellungen über Notwendigkeit und Einsatzgebiete einer derartigen Zentralstelle herausgebildet; überall ist man gerade dabei, bisher dezentral abgewickelte Einzelaktivitäten zusammenzufassen und den auch von der ESG angestrebten integrierten Betrieb über ein zentrales Rechenzentrum zur System-Entwicklung und -Integration zu verwirklichen.

Nicht zuletzt haben die Diskussionen in diesem Kreise hier ergeben, daß ähnliche Vorstellungen auch bei

anderen Stellen in der Luft liegen und zum Teil sogar bis auf die Terminologie ganz ähnlich gesehen werden.

Ein mögliches Vorgehen in dieser Richtung könnte wie folgt aussehen:

- 1) Auswahl eines "repräsentativen" Prozesses;
Demonstrationswirkung, d.h. mit entsprechendem Innovations- und Multiplikatoreffekt
- 2) Erarbeitung der funktionellen Anforderungen für diesen Prozeß; abwickelbar als Modellverlauf für eine Systemanalyse.
- 3) Spezifikation der Prozeßkomponenten in Hardware und Software
- 4) Bereitstellung der entsprechenden Standard- und Automationsmittel - ERZ - Rechner, Prozeßperipherie und Prozeßendstellen
- 5) Indirekte Erstellung und Vorintegration der Prozeßsoftware auf dem Emulationsrechenzentrum
- 6) Stufenweiser Aufbau des Prozesses im Labor;
Adaptierung an das Emulationsrechenzentrum
- 7) Integration, Test und Einfahren des Prozesses.

2.3

Entwicklungskonzept der ESG für ein Programmsystem
zur Erstellung und Generierung von Systemsoftware
für Prozeßrechner

In der ESG haben sich konkrete Vorstellungen samt ersten Lösungsideen sowohl bezüglich eines derartigen Modells als auch bezüglich der schon erwähnten offenen Fragen herausgebildet. Wir sind gerne bereit, diese uns beschäftigenden Probleme in a l l e n Anhörungsveranstaltungen zu diskutieren.

Im Rahmen der Sprach- und Betriebssystem-Veranstaltungen würden wir Sie gerne bekannt machen mit dem Entwicklungskonzept der ESG für ein Programmsystem zur Erstellung und Erzeugung von System - Software für die Prozeß - Automation.

Auf die Einzelheiten dieses Konzepts wird Herr Eichenauer in einem gesonderten Referat zu sprechen kommen. Lassen Sie mich zum Abschluß noch auf die Vorteile hinweisen, die sich u.a. nach Realisierung des Entwicklungskonzepts einstellen werden:

- 1) Die Umstellung der gesamten Systemsoftware für Prozeßrechner auf andere Rechenanlagen wird mit relativ geringem Aufwand möglich sein.
- 2) Nach einmaliger Implementierung auf einem Rechner werden zukünftige Spezialsprachen auf allen Rechner-typen verfügbar sein, an deren Code der Compiler-generator angepaßt wird.

2. Fachgespräch am 4./5.10.1971
über "Betriebssysteme".

Teilnehmerliste

für das Fachgespräch "Betriebssysteme" am 4./5. 10. 1971
im Kernforschungszentrum Karlsruhe

<u>Name:</u>	<u>Institution:</u>
Feig	GID, Aschaffenburg
Werum	Entwicklungsbüro Werum, Erbstorf
Dehn	Nixdorf Computer AG, Berlin
Kamp	TU Berlin
Sinzinger	Siemens AG, Erlangen
Rüb	TU München
Würz	Hahn-Meitner-Institut, Berlin
Überschaer	Siemens AG, Berlin
Tarara	Bundesanstalt für Flugsicherung, Frankfurt
Rost	AEG-TELEFUNKEN, Berlin
Martin	BBC, Mannheim
Koch	BBC, Mannheim
Reh	ESG, München
Eichenauer	ESG, München
Lauber	Universität Stuttgart
Sonderman	VW AG, Wolfsburg
Jung	I.R.A., Frankfurt
Eichner	Universität Stuttgart
Hanbaba	SCS, Hamburg
Morlock	Interdigit
Wettstein	Universität Karlsruhe
Hofmann	Siemens AG, Erlangen
Pfeiffer	KFA, Jülich
Abend	Hahn-Meitner-Institut, Berlin
Schüring	TH Aachen
Schmidt	Mathematischer Beratungsdienst, Dortmund
Elzer	Universität Erlangen
Rongen	KFA, Jülich
Krüger	GfK/IDT
Stams	GfK/PDV
Eckert	GfK/PDV

Schwerdtner

Schlurick

Hotes

Bracht

Haase

Mailänder

Brandes

Goos

IBM, Sindelfingen

Universität Erlangen

SCS, Hamburg

I.R.A., Frankfurt

Unicomp und DFE, Blankenloch

PSI, Berlin

Universität Karlsruhe

Universität Karlsruhe

E r g e b n i s - P r o t o k o l l

zum Fachgespräch über "Betriebssysteme" am 4./5.10.1971
im Kernforschungszentrum Karlsruhe, Schule für Kerntechnik

1. Ablauf

Das Fachgespräch wurde in folgenden Abschnitten durchgeführt:

- Vorlage des Themenkatalogs durch Herrn Professor Krüger,
- Vorstellung der Gesprächsteilnehmer mit Angabe des Tätigkeitsbereichs und der speziellen Vorstellungen über das Thema Betriebssysteme,
- Kurze Referate aus dem Bereich Betriebssysteme,
- Brain storming,
- Anfertigung einer Stoffsammlung,
- Einordnung der Stoffsammlung in den von Professor Krüger vorgelegten Themenkatalog, Erweiterung dieses Katalogs,
- Vorschläge über notwendige Aktivitäten.

2. Der Themenkatalog

1. Anforderungen an Rechnerkonfiguration und Betriebsorganisation in der Prozeßrechnertechnik
Methodik und Technik der Anwendungsanalyse,
Zusammenstellung und Systeme von Anwendungsprofilen,
2. Komponenten und Struktur von Betriebssystemen
Planung und Steuerung von Aufträgen und Abläufen
Verwaltung von Informationsbeständen in realen und virtuellen Speichern
Probleme des Datenaustausches mit Endstellen

Maßnahmen zur Erhöhung der Systemsicherheit.

Manuelle Eingriffsmöglichkeiten in die interne Struktur und interne Abläufe

Aufbau von Betriebssystemen aus Komponenten

Klassen von Betriebssystemen

Mehrprozessor- und Mehrrechner-Systeme

3. Herstellung von Betriebssystemen

Programmiersprachen für das Schreiben und Beschreiben von Betriebssystemen
Entwurfssystematik, Betriebssysteme - Generatoren.

4. Prüfung, Wartung, Betriebssicherheit, Fortentwicklung

5. Terminologie zur Beschreibung von Betriebssystemen.

3. Vorschläge für notwendige Aktivitäten:

1. Anwendungsanalyse

Anwendungsgebiete und Anzahl der dort eingesetzten Prozeßrechner und speziell die verwendeten und von der Aufgabe her wünschenswerten Betriebssysteme sollen festgestellt werden. Die Ergebnisse sollen systematisch nach Anwendungsklassen und Anwendungsprofilen geordnet werden.

2. Vergleichende Studie über z.Zt. existierende Betriebssysteme und Methoden ihrer Herstellung

Die Funktionen und Eigenschaften sollen festgestellt und verglichen werden, insbesondere sollen die Methoden der Betriebssystem-Herstellung untersucht und beschrieben werden.

3. Terminologie

Eine Beschreibung und Darstellung der Betriebssysteme ist nur dann möglich, wenn hierfür geeignete Terminologien vorhanden sind. Deshalb sollte in Zusammenhang mit den Arbeiten 1 und 2 eine Studie über Begriffsabgrenzung und Begriffsinhalt der verschiedenen Termini angefertigt werden. Diese

Weiterentwicklung der Möglichkeit des gegenseitigen Verstehens muß natürlich angeschlossen werden an die bestehenden Festlegungen, Normungen und Sprachgebräuche. Bei dem Fachgespräch fiel auf, daß verschiedene Gruppen ganz unterschiedliche Sprachelemente in unterschiedlicher Weise benutzen.

4. Anforderung von prozedurorientierten Sprachen an die Funktionen der Betriebssysteme

Im Rahmen der Entwicklung von prozedurorientierten Sprachen werden in absehbarer Zeit neue Definitionen von Sprachen und ihren Eigenschaften verlangt werden. Für die Planung der weiteren Entwicklung von Betriebssystemen ist es wichtig zu wissen, welche Funktionen die Betriebssysteme haben sollen, damit prozedurorientierte Sprachen ohne allzu große Schwierigkeiten implementiert werden können. Um zu einer brauchbaren Konzeption und Methodik zu gelangen, sollte der Versuch gemacht werden, insbesondere eine Definition der Schnittstelle zwischen PEARL bzw. PEARL-Subsets und Betriebssystemen zu geben.

5. Pilot-Implementierung von Betriebssystemen zusammen mit der geplanten Pilot-Implementierung von PEARL.

Ein wichtiges Ergebnis des Fachgespräches über prozedurorientierte Prozeßprogrammiersprachen am 30.9./1.10. 1971 bestand darin, daß mit höchster Priorität eine Pilot-Implementierung von PEARL bzw. geeigneten PEARL-Subsets durchgeführt werden sollen. Dies ist nur dann möglich, wenn die bestehenden Betriebssysteme in geeigneter Form ergänzt, erweitert oder abgeändert werden. Es sollte versucht werden, diese Erweiterung der bestehenden Betriebssysteme so durchzuführen, daß sie auch auf anderen Anlagen wieder verwendbar sind. Andererseits sollte das Streben nach Generalisierung und Transferierbarkeit nicht so weit getrieben werden, daß sich die Pilot-Implementierung von PEARL zu lange hinausschiebt. Zeit und Aufwand für Speziallösungen müssen mit Zeit und Aufwand für transferierbare Lösungen verglichen werden, damit der zweckmäßigste Weg gefunden wird.

6. Weiterentwicklung von Betriebssystemen.

Eine gezielte, den Anforderungen entsprechende Weiterentwicklung der Betriebssysteme ist eine weitere sehr wichtige Aufgaben. Bei der Durchführung sollten insbesondere die im folgenden aufgeführten Gesichtspunkte berücksichtigt werden:

6.1. Wechselwirkung Betriebssystem - Hardware

Hierbei sind zwei Gesichtspunkte wichtig:

- Welche Hardware-Strukturen sind möglich und wünschenswert um Betriebssystemfunktionen in günstiger Weise zu realisieren, z.B. fest verdrahtete Mikroprogramme usw.
- Für langfristige Konzepte der Entwicklung von Betriebssystemen muß unbedingt der langfristige Trend der Entwicklung der Rechnerstrukturen beachtet werden. Möglicherweise sollen hierüber besondere Untersuchungen angestellt werden.

6.2. Sicherheitsanforderungen sollen in viel stärkerem Maße als bisher beachtet werden. Der Betrieb muß bei Störungen in möglicherweise veränderter und eingeschränkter Form aufrechterhalten werden. Hierfür muß das Betriebssystem geeignete Eigenschaften besitzen. Das Betriebssystem muß auch die Möglichkeit zur manuellen Fehlerdiagnose enthalten.

6.3. Es sollen neue Strategien und Verfahren bereitgestellt werden für die Unterstützung der Anwendungsprogrammierung hinsichtlich z.B. Programm-Tests oder Effizienz in Laufzeit und in Betriebsmittelbenutzung.

6.4. Adaptives Verhalten von Betriebssystemen entsprechend der Inanspruchnahme der Hardware/Software-Konfiguration sollte untersucht und entworfen werden. Hierbei müssen insbesondere Entscheidungskenngrößen für Strategien festgelegt werden.

6.5. Betriebssysteme für Mehrrechnerkonfigurationen werden in Zukunft von größerer Bedeutung sein und sollten daher rechtzeitig konzipiert werden.

6.6. Die Austauschbarkeit von Betriebssystemen von Maschine zu Maschine ist ein weiterer wichtiger Gesichtspunkt.

6.7. Modularer Aufbau und "Zwiebelstruktur" von Betriebssystemen und sonstigen Systemkomponenten sind für die Entwicklung von Betriebssystemen von Wichtigkeit.

7. Herstellung von Betriebssystemen

Die Methoden zur Herstellung von Betriebssystemen sollen weiter entwickelt werden. Insbesondere soll geprüft werden, ob es möglich ist, höhere, maschinenunabhängige Programmiersprachen zu entwickeln, mit deren Hilfe Betriebssysteme programmiert werden können.

7.1. Es wurde vorgeschlagen, daß Anfang 1972 eine Arbeitstagung durchgeführt wird, bei der die Möglichkeiten für solche höhere Programmiersprachen erörtert werden. Die Gesellschaft für Kernforschung wird die Planung der Organisation dieser Arbeitstagung übernehmen.

7.2. Bei der Herstellung von Betriebssystemen sollte von der Möglichkeit der Simulation und Emulation Gebrauch gemacht werden.

8. Bewertungsmaßstab, Leistungsvergleich:

Zur Beurteilung und zum Vergleich von Betriebssystemen sollen Bewertungsmaßstäbe erarbeitet werden. Ein darauf aufbauender Leistungsvergleich könnte als Grundlage für die Weiterentwicklung von Betriebssystemen wichtige Hinweise geben.

9. Allgemeine Simulationsmethoden

Es wurde mehrfach darauf hingewiesen, daß zur Herstellung ganzer Prozeßrechnersysteme der Einsatz der Simulation sehr nützliche Dienste leisten könnte. Es wurde vorgeschlagen, hier eine besondere Technik bereitzustellen.

10. Rechnerverbund

Die vorliegenden Erfahrungen sollen erfaßt und ausgewertet werden und sollten Hinweise auf die zur Zeit bestehenden Möglichkeiten und Schwierigkeiten für eine Weiterentwicklung zeigen.

gez. Stams

gez. Krüger

K u r z r e f e r a t

anlässlich des Fachgesprächs "Betriebssysteme für Prozeß-
rechenanlagen" am 5. Oktober 1971 im Rahmen des 2. DV-
Förderungsprogramms, Projekt "Prozeßlenkung mit DV-Anlagen"

Anforderungen an Betriebssysteme bei Einsatz von
mehreren Rechnern zur Produktionssteuerung

Dr.-Ing. Peter Hanbaba

SCIENTIFIC CONTROL SYSTEMS

LTD & CO GMBH

Geschäftsstelle Essen

43 Essen 1

Limbecker Platz 1

Anforderungen an Betriebssysteme bei Einsatz von mehreren Rechnern zur Produktionssteuerung

SCS realisiert zur Zeit ein Mehrrechner-Projekt zur Fertigungssteuerung in einem Grobblechwalzwerk. Dazu wird ein hierarchisch gegliedertes Rechnersystem eingesetzt. Die planerischen Aufgaben werden von einem Großrechner wahrgenommen; die Aufgaben der Betriebsdatenvorgabe und -erfassung sowie Prozeßsteuerung übernimmt das Nachrichtensystem. Hier soll über Hard- und Software des Nachrichtensystems berichtet werden.

1. Hardware

Abbildung 1 gibt einen schematischen Überblick über die Hardware-Konfiguration.

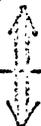
An ein Duplex-Steuerrechnersystem sind Terminals angeschlossen. Die Terminals sind so ausgelegt, daß bei Ausfall die zur Aufrechterhaltung einer Notorganisation erforderlichen Geräte auf das Nachbarterminal umgeschaltet werden können. Bei Ausfall eines Steuerrechners wird auf den anderen umgeschaltet, der dann zu den gleichen Daten Zugriff hat. In unserem Anwendungsfall ist diese Umschaltung nicht stoßfrei, sondern die Ausführung einiger Funktionen wird wiederholt. Die Datenspeicherung ist voll redundant (Platte-Platte oder Kernspeicher-Platte oder Kernspeicher-Kernspeicher).

Für dieses Terminal-Konzept sprechen folgende Gründe:

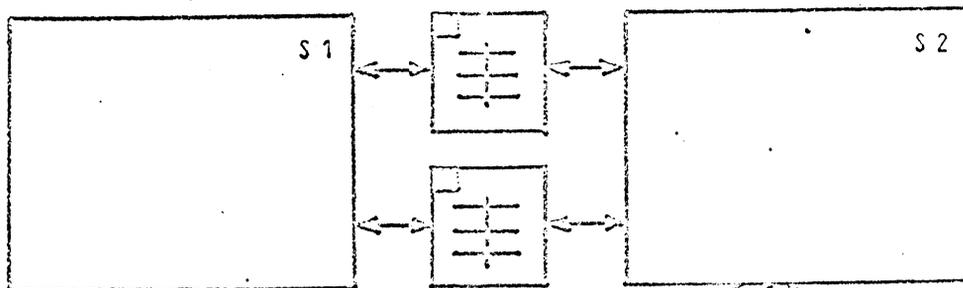
- . Dezentrale Aufstellung der Terminals
- . Lösung rechenintensiver Aufgaben
- . Lösung unabhängiger Teilaufgaben großen Datenumfangs
- . Durchführung E/A-intensiven Dialogverkehrs
- . Steigerung der Betriebssicherheit bei vertretbarem Aufwand
- . Ermöglichung einer stufenweisen Automatisierung
- . Schaffung größtmöglicher Flexibilität

Planung

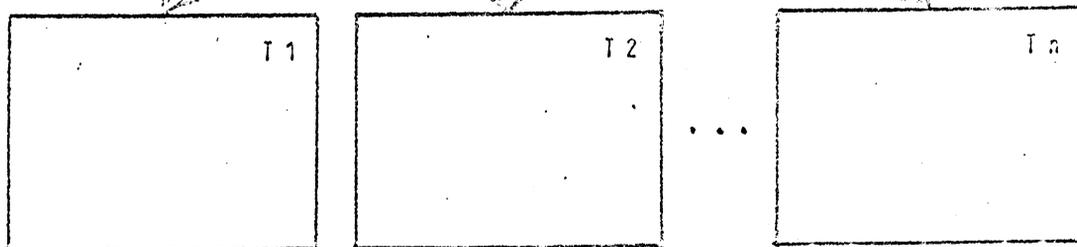
Planungsrechner



Steuerrechner

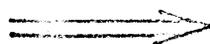


Terminals



Betrieb

Mensch Prozess



Fertigungsstraße

Abb. 1: Hardware

2. Software

In Abbildung 2 sind die wesentlichen Aufgaben der Steuerrechner und der Terminals zusammengestellt.

Die Aufgaben der Steuerrechner sind:

- . Datenverwaltung
- . Sollwertweitergabe
- . Istwerterfassung
- . Soll-Ist-Vergleich
- . Reaktion auf Produktionsstörungen

Die Aufgaben der Terminals lassen sich klassifizieren nach:

- . E/A-intensiver Dialogverkehr
- . Rechenintensive Auswertungen
- . Behandlung von Teilkomplexen mit großem Datenbestand

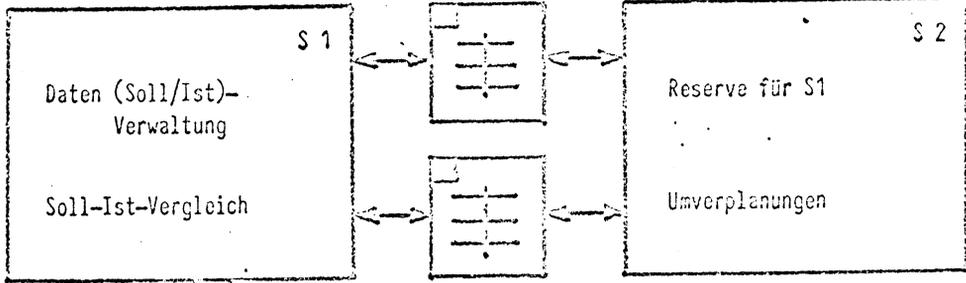
Planung

Planungsrechner

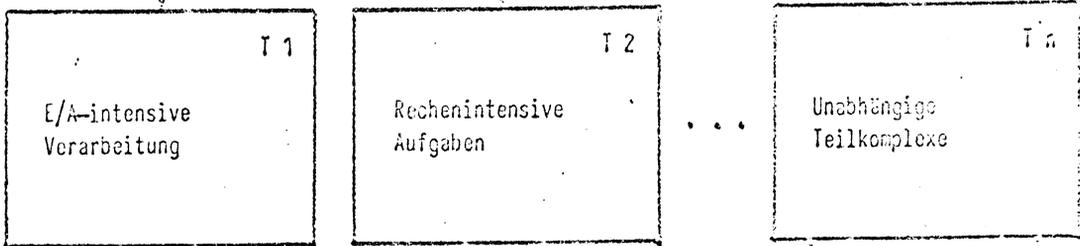


Bestelldaten
Fertigungsdaten

Steuerrechner



Terminals



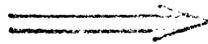
Betrieb

Mensch

Prozeß

Mensch

Prozeß



Fertigungsstraße

Abb. 2: Software

3. Betriebssysteme

Grundsätzlich vorhanden ist ein Standard-Betriebssystem (SBS) mit folgenden Funktionen:

Aufgabenverwaltung: Beauftragen
Fortsetzen
Warten
Beenden

Interruptverwaltung: Aktivieren
Deaktivieren

Zeitverwaltung: Anschalten
Abschalten

Geräteverwaltung: Beauftragen
Belegen
Freigeben

Gerätetreiber: Ein-/Ausgabe
Datenprüfung
Datenkonvertierung

Bedienung
Systemtest

Hier soll nur auf die zusätzlich benötigten Funktionen eingegangen werden, die in der Abbildung 3 aufgeführt sind.

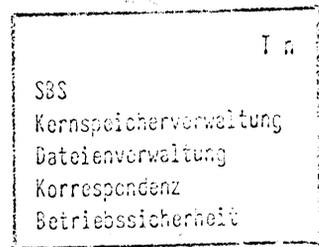
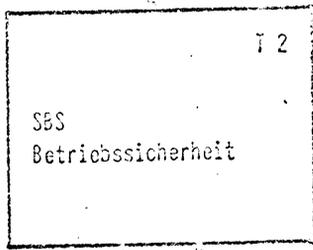
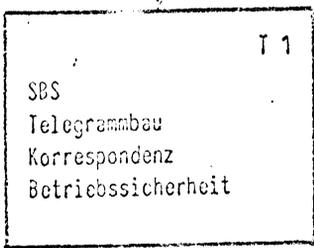
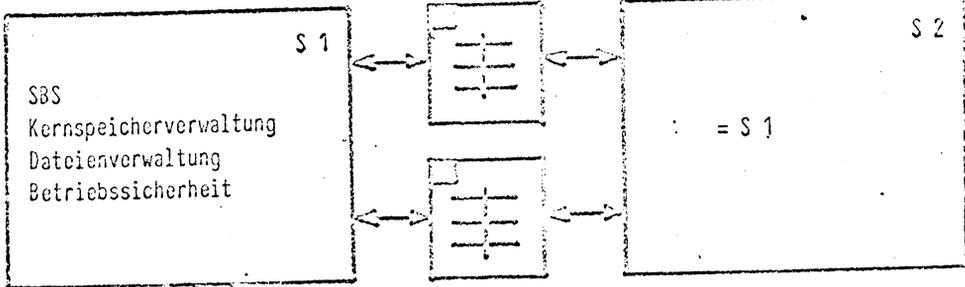
Planungsrechner

Planung

Steuerrechner

Terminals

Betrieb



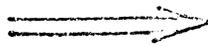
Mensch

Prozeß

Mensch

Prozeß

S B S = Standard-Betriebssystem



Fertigungsstraße

Abb. 3: Betriebssystem

Zusätzliche Funktionen bei den Steuerrechnern:

1) Kernspeicherverwaltung

- a) für Programme: Je nach Anwendungsfall ist die Anwendung unterschiedlicher Verfahren (Koordination der Programmlaufzeiten, dynamische Zuteilung, virtuelle Speicher) notwendig.

Frage:

Unter welchen Voraussetzungen ist die Anwendung der einzelnen Methoden zweckmäßig?

- b) für Daten: Das Betriebssystem sollte
- . das zeitweise Zwischenspeichern von Daten
 - . und das Laden von Dateien übernehmen.

2) Dateienverwaltung

Da unterschiedliche Dateitypen behandelt werden müssen, benötigt der Anwender ein Spektrum von einfachen Bausteinen.

Standard-Funktionen: Adresse suchen
Schreiben - mit direkter
Lesen) Adressierung
Belegen
Freigeben

Spezielle Funktionen: Eröffnen
Löschen
Schließen
Schützen
Koordinieren (bei Zugriff von mehreren Rechnern)

Frage:

Auf welche Typen und Funktionen kann man sich beschränken?

3) Betriebssicherheit

Eine wesentliche Rolle bei Mehrrechnersystemen spielt die Betriebssicherheit. Aufgabe des Betriebssystems ist es,

- . die gegen Ausfall gesicherte Datenspeicherung
- . die Überwachung des Datenverkehrs
- . die Überwachung der Zentraleinheiten
- . und die automatische Umschaltung

zu übernehmen.

Frage:

Welche Methoden zur Fehlererkennung sind anzuwenden?
Wie soll darauf reagiert werden?

Zusätzliche Funktionen bei den Terminals:

Die Auswahl der benötigten Bausteine wird durch die zu lösenden Aufgaben bestimmt.

1) Korrespondenz Mensch - Terminal - Steuerrechner

In den Terminals mit E/A-intensiven Aufgaben sind folgende Abläufe typisch:

- . Dialog Mensch - Terminal über Betriebstastaturen und Anzeigegeräte
- . Anforderungstelegramm zum Steuerrechner
- . Antworttelegramm vom Steuerrechner
- . Ausgabe zum Bediener

Für den gesamten Bereich der Produktionssteuerung scheinen diese Abläufe standardisierbar zu sein. Daher wäre es wünschenswert, wenn das Betriebssystem Funktionen für

- . die Korrespondenz Mensch-Maschine und
- . den Telegrammbau

beinhalten würde.

Frage:

Gilt dieser Wunsch auch für andere Anwendungsfälle?

2) Betriebssicherheit

Zur Erhöhung der Betriebssicherheit sind

- . der Telegrammverkehr zum Steuerrechner
 - . und die Funktionstüchtigkeit der Peripheriegeräte
- zu überwachen.

4. Folgerungen

Ein Mehrrechner-Betriebssystem müßte demnach das folgende Spektrum an Funktionen umfassen:

		<u>S1/2</u>	<u>T1</u>	<u>T2</u>	<u>Tn</u>
SBS		x	x	x	x
Kernspeicherverwaltung:	Koordination	x			x
	dyn. Zuteilung				
	virtueller Speicher				
	Zwischenspeicherung				
	von Daten	x			
Dateienverwaltung:	Lesen	x			x
	Schreiben	x			x
	Eröffnen				
	Löschen				
	Schützen	x			
	Koordinieren	x			
	Adresse suchen	x			x
	Belegen	x			
	Freigeben	x			
Ereignisverwaltung:	Anbinden				
	Entbinden				
	Auslösen				
Datenübertragung:	Telegrammverkehr	x	x	x	x
	Telegrammbau		x		
Korrespondenz Mensch-Maschine:			x		
Betriebssicherheit:	Datenspeicherung	x			
	Fehlererkennung	x	x		
	Überwachung Datenverkehr	x	x	x	x
	Automat. Umschaltung	x			

SBS = Standard-Betriebssystem

Bei der Auswahl für den konkreten Anwendungsfall spielen folgende Einflußgrößen eine Rolle:

- . die in der Zeiteinheit zu bearbeitende Datenmenge
- . der gesamte Datenbestand
- . das geforderte zeitliche Auflösungsvermögen
- . die Forderungen an die Betriebssicherheit und
- . die Art und Intensität des Verkehrs Mensch-Rechner

Die hier gemachten Aussagen basieren auf einem speziellen Anwendungsfall.

Die Konfiguration des Nachrichtensystems ist jedoch auch für andere Arten der Fertigungssteuerung und weitere Anwendungsfälle von Interesse. Daher sollte nach unserer Meinung untersucht werden, wie ein allgemeines Realzeit-Betriebssystem für ein solches Mehrrechnersystem aussehen müßte.

Bernd Eichenauer

Elektronik - System -GmbH

Einige Anmerkungen über die Beziehungen zwischen
Prozeßsprache und Betriebssystem

Kurzreferat, vorgetragen anlässlich der Fachgespräche über die
Förderung der Forschung und Entwicklung von Betriebssystemen
für Prozeßrechner am 4. 10. 71 bei der GFK, Karlsruhe

Die Bereitstellung von Sprachelementen

- o zur Beschreibung der Hardware - Konfiguration,
- o für den Verkehr zwischen Rechenanlagen und Prozeßendstellen und
- o zur Synchronisation des Programm- und Prozeßgeschehens

in prozeßorientierten Programmiersprachen (insbesondere in PEARL) unterlag in letzter Zeit starker Kritik. Dabei sind im wesentlichen folgende Einwände vorgebracht worden:

1. Sprache und Betriebssystem haben nichts miteinander zu tun.
Die Sprache soll nur die Ankopplung an das jeweils vorhandene Betriebssystem gestatten.
2. Man kann heute nicht wissen, welche Betriebssysteme in einigen Jahren benötigt werden. Es könnte sich zeigen, daß der jetzt konzipierte Sprachvorrat nicht ausreichend oder zumindest nicht adäquat ist.
3. Bei Einsatz prozeßorientierter Programmsprachen (insbesondere von PEARL) sind nur Teile der bisher erstellten Betriebssysteme für Prozeßrechner weiterverwendbar.
Da bei der Implementierung von prozeßorientierten Programmiersprachen jeweils das umfangreiche Betriebssystem vollständig erstellt werden muß, was aus Effektivitätsgründen im Assembler zu erfolgen hat und mit großem Aufwand verbunden ist, dürften diese Sprachen in den nächsten Jahren für ein hinreichend großes Spektrum von Rechnermodellen nicht verfügbar sein.
4. Die Konzipierung einer allgemeinen prozeßorientierten Programmiersprache ist vielleicht möglich. Jedoch sind wegen des Umfangs der Sprache und wegen der Komplexität der Sprachelemente umfangreiche und komplizierte Übersetzungs-

programme erforderlich. Die Erstellung dieser Übersetzungsprogramme für die verschiedenen Rechnermodelle ist aus Zeit- und Kostengründen nicht sinnvoll.

Trotz der Kürze der verfügbaren Zeit möchte ich versuchen, plausibel zu machen, daß die beiden zuerst genannten Einwände unberechtigt sind und daß die restlichen Einwände mithilfe eines geeigneten Implementierungskonzepts entkräftet werden können.

Der Sprachentwickler hat beim Design einer höheren Programmiersprache die Aufgabe zu lösen, für einen bestimmten Problembereich gut übersetzbare Sprachelemente zu schaffen, mit denen die Anwender der Sprache die Ergebnisse ihrer Problemanalysen auf logischem Level und möglichst problemangemessen darstellen können. Beispielsweise sind in Programmiersprachen für den Compiler- und den Betriebssystembau Sprachelemente zum Anlegen und Bearbeiten von Kellerspeichern vorzusehen; in technisch-wissenschaftlichen Sprachen müssen Sprachelemente zur einfachen Deklaration und Verarbeitung tensorieller Größen geboten werden.

Je breiter das Einsatzspektrum einer Programmiersprache sein soll, desto niedriger wird im allgemeinen ihr Niveau sein und desto mehr Arbeit muß der Anwender der Sprache bei der Programmerstellung verrichten. Um einerseits ein breites Spektrum von Prozeß-Automationsaufgaben zu überdecken, andererseits auch dem Anwender mit geringer EDV-Erfahrung, der oft sein Automationsproblem besser als der EDV-Spezialist kennt, die Programmerstellung zu ermöglichen, wird bei der Entwicklung prozeßorientierter Programmiersprachen normalerweise ein mittleres Sprachniveau angestrebt.

Die bisher im Rahmen von Programmiersprachen formulierbaren Zusammenhänge sind aus in sich abgeschlossenen Denkmodellen hervorgegangen, die keinen unmittelbaren Bezug zur Umwelt hatten. Die Ausführung der Programme auf einem Rechenautomaten (einschließlich seiner Standard - Peripherie) lässt sich etwa mit der Lösung von Denkaufgaben (einschließlich der menschlichen Ein/Ausgabe) vergleichen.

Bleibt man in diesem - den Sachverhalt nur unvollständig widerspiegelnden - Bilde, so kann die Ausführung von Automationsprogrammen etwa mit der Abwicklung vorgezeichneter Verhaltensmodelle zur Steuerung des Körpers in der Umwelt verglichen werden.

In der Prozeß - Automation hat man es mit Problemstellungen zu tun, die im Rahmen von Rechenautomaten und ihrer Standard - Peripherie nicht lösbar sind. Um dem Verhalten des parallel und zwischen definierten Synchronisationspunkten weitgehend unabhängig zum Programm ablaufenden technischen Prozesses Rechnung zu tragen, werden Ausdrucksmittel zur prozeßgerechten Steuerung des Programmablaufs sowie zur Definition und zweckmäßigen Ausnutzung der eingesetzten Automationsmittel benötigt. Setzt man voraus, daß die erforderlichen Ausdrucksmittel bekannt sind, so ist es im Hinblick auf

- o die leichte Erlernbarkeit der Programmiersprache,
 - o den hohen Dokumentationswert der Automationsprogramme und
 - o die weitgehende Anlagenunabhängigkeit der Automationsprogramme
- sinnvoll, diese Ausdrucksmittel als genormte Elemente der Sprache zu bieten.

Durch die Bereitstellung von problemangemessenen Sprachelementen für die Prozeßprogrammierung wird der für die jeweilige Automationsaufgabe erforderliche Betriebssystemausbau auf Quellsprachenebene definierbar. Man kann sich Real - Zeit - Betriebssysteme aus einem invarianten und aus programmabhängigen Teilen aufgebaut denken. Der invariante programmunabhängige Teil enthält die Grundfunktionen, die zur Ausführung aller Automationsprogramme benötigt werden (Interrupt - Programm, Kern des Bedienungssystems, u. s. w.). Die programmabhängigen Teile werden von einem Betriebssystem - Composer aufgrund der in den Quellenprogrammen definierten Anforderungen mithilfe einer Bibliothek von Prozeduren und Makros aufgebaut.

Die Auffassung, daß Betriebssystem und Prozeßsprache nichts miteinander zu tun haben, erscheint danach nicht sinnvoll. Durch die Prozeßsprache wird ja gerade der sprachorientierte Teil des Betriebssystems definiert. Ob ein Programmierer den für die jeweilige Anwendung erforderlichen Betriebssystemausbau implizit über sein Quellenprogramm festlegen darf oder ob er, wie dies bisher gehandhabt wird, die jeweils erforderliche optimale Betriebssystem - Konfiguration beim Erstellen des Automationsprogramms selbst erkennen und zusammenstellen muß, ist letztlich eine Frage der Kostenwirksamkeit und der Praktikabilität.

Zu dem zweiten Einwand möchte ich bemerken, daß in den bekannten prozeßorientierten Programmiersprachen (insbesondere in PEARL) nur die logischen Grundelemente auf sprachlicher Ebene geboten werden, die zur Verknüpfung des Programm- und Prozeßgeschehens, für den Datenaustausch und zur Konfigurationsbeschreibung denkbar sind. Aus diesem Grund ist nicht zu erwarten, daß sich der

gebotene Sprachvorrat in abschbarer Zeit als zu eng erweist.

Was schließlich die beiden zuletzt genannten Einwände betrifft, so lassen sich diese durch geeignete Implementierungskonzepte, bei denen auf möglichst weitgehende Wiederverwendbarkeit der zu erstellenden Programmpakete geachtet wird, entkräften. Anlässlich der Fachgespräche über prozedurorientierte Prozeß - Programmiersprachen habe ich eine mögliche Vorgehensweise erläutert. An dieser Stelle möchte ich nochmals die meines Erachtens wichtigsten Teilkomponenten angeben.

Zur Erstellung wiederverwendbarer Systemprogramme wird:

1. eine niedere System - Programmiersprache benötigt, in der
 - o alle nicht laufzeitintensiven Systemprogramme (z. B. Compiler - Generator, Emulator - Generator, Betriebssystem - Composer, u. s. w.) in wiederverwendbarer Form und
 - o alle laufzeitintensiven Systemprogramme (z. B. Ablaufsteuerung, Zeit - Warteschlange, Interrupt - Programm) wenigstens auf formalem Niveauerstellt werden können.

2. zur Reduzierung des Anpassungsaufwands bei der Überführung des Systems auf verschiedene Typen von Rechenanlagen ein virtueller Rechenautomat benötigt, von dessen Code (Compiler - Zwischensprache) ausgehend man den Code heutiger und für die Zukunft zu erwartender Prozeßrechner mit geringem Aufwand erreichen kann.

Die Definition eines virtuellen Rechenautomaten kann in Zusammenhang mit der Strukturierung eines " idealen " Prozeßrechners gesehen werden und dürfte daher die Hardware - Entwicklung befruchten.

Zum Schluß möchte ich noch kurz auf den gegenwärtigen Stand der Entwicklung von sprachorientierten Betriebssystemen für prozeßorientierte Programmiersprachen von mittlerem Sprachniveau eingehen. Der Aufbau dieser Betriebssysteme ist heute für Einprozessoranlagen - insbesondere für Kernspeichersysteme - bekannt. Für Mehrprozessor- und Mehrrechnersysteme sind meines Wissens noch keine akzeptablen, d. h. hinreichend allgemeinen Konzepte, vorgelegt worden. Für die weitere Entwicklung wäre es nützlich, wenn folgende Teilgebiete genauer untersucht würden:

- o Aufbau von sprachorientierten Betriebssystemen für Mehrprozessor- und Mehrrechnersysteme.
- o Implementierungsmöglichkeiten für effektive Nachschuborganisationen bei Einsatz prozeßorientierter Programmiersprachen (automatisches und/oder halbautomatisches Roll-in/Roll-out von Programmen und Daten).

Anmerkung der Redaktion: Einige weitere an dieser Stelle gehaltene Referate werden in den nächsten PDV-Mitteilungen erscheinen.

3. Fachgespräch am 8. 10. 1971
über "Rechnerangepaßte Prozeßperipherie".

Teilnehmerliste

für das Fachgespräch "Rechnerangepaßte Prozeßperipherie" am 8. 10. 1971
im Kernforschungszentrum Karlsruhe

Name:

Institution:

Rehr	TH Aachen
Pavlik	Siemens AG, Karlsruhe
Kaltenecker	Siemens AG, Karlsruhe
Mielentz	BBC, Mannheim
Sinzinger	Siemens AG, Erlangen
Hofmann	Farbwerke Hoechst AG
Mesch	Universität Karlsruhe
Trebst	Universität Erlangen
Zabler	Universität Karlsruhe
Ammon	AEG-TELEFUNKEN, Frankfurt
Pießler	Hartmann u. Braun AG
Ankel	BASF
Eichner	Universität Stuttgart
Rohde	AEG-TELEFUNKEN, Konstanz
Kronmüller	Universität Karlsruhe
Didic	Unicomp/DFE, Blankenloch
Frech	Unicomp/DFE, Blankenloch
Haase	Unicomp/DFE, Blankenloch
Reh	ESG, München
Wiethoff	Hoesch Hüttenwerke AG, Dortmund
Busler	MBP, Dortmund
Klessmann	Hahn-Meitner-Institut, Berlin
Schmäing	BASF
Ossenberg	Fraunhofer-Gesellschaft
Dittmer	VEW, Dortmund
Brandes	Universität Karlsruhe
Lauber	Universität Stuttgart
Krüger	GfK/IDT

E r g e b n i s - P r o t o k o l l

zum Fachgespräch "Rechnerangepaßte Prozeßperipherie" am 8. 10. 1971
im Kernforschungszentrum Karlsruhe

1. Datenübertragung

Bis heute existiert eine gewisse Unverträglichkeit von Prozeßrechner auf der einen Seite und Meß- und Stellgliedern auf der anderen Seite. Beide Techniken haben sich zunächst weitgehend unabhängig voneinander entwickelt. Die Meß- und Stellglieder innerhalb der Meß- und Regeltechnik, der Digitalrechner zunächst als mathematisch-wissenschaftliches Instrument, der dann Ein- und Ausgänge erhielt, die mit von Fall zu Fall verschiedenen normierten Spannungen, eingepprägten Strömen, Eingangswiderständen usw. arbeiten. Für den Anwender ergibt sich das Problem, die beiden Gerätesysteme miteinander zweckmäßig zu koppeln. Hinzu kommt ein in der Meß- und Regeltechnik in dieser Form neuartiges Problem, nämlich die zuverlässige Signalübertragung über weite Strecken mit einer großen Anzahl von Eingängen und Ausgängen. Die Möglichkeiten, vor Ort Signale anzupassen, umzuwandeln und auf Signalleitungen zu schalten und in dieser Form an den entfernt stehenden Prozeßrechner weiterzuleiten, sind bisher kaum erforscht oder ausgenutzt.

Die Diskussion ergab deshalb, daß in weiten Bereichen Grundlagenuntersuchungen angestellt werden müssen. Zunächst einmal geht es um die Grundfrage, welche Übertragungsarten ein höchstes Maß an Sicherheit bieten und bei den geforderten Datenraten und bei vertretbaren Kosten am zweckmäßigsten sind. Hierbei wurden Stichworte wie analog/digital, parallele/serielle Meßwertübertragung, Stern/Ringleitung, rechnerabhängige oder rechnerunabhängige Steuerungen von Meßwertwandlern und Meßstellenumschaltern, sowie Abschirmungs-, Sicherheits- und Erdungsfragen erörtert. Hinzu kommen spezielle Fragen, wie bei einer Vielzahl von anzuschließenden Meßfühlern und Stellgliedern eine rasche Fehlererkennung und -Beseitigung durchgeführt werden kann. Weitere Fragen der Verwendung vorhandener Leitungssysteme, z.B. auch des Telefonnetzes wurden angesprochen. Welche Möglichkeiten ergeben sich, wenn Kleinrechner z.B. als zentrale oder dezentrale Verkehrsverteiler eingesetzt werden?

Als wichtig wurde die Schnittstellennormierung angesehen. Allerdings erscheint eine Normierung erst dann sinnvoll, wenn technisch ein gewisser

Reifegrad erreicht ist und hinreichende Erfahrungen vorliegen. Die Elemente des CAMAC-Systems sollen auf ihre wirtschaftliche und funktionelle Eignung als Teile der Datenübertragung untersucht werden.

Vorschlag:

Anfertigung einer Studie über Verfahren zur Signalübertragung und zum Schnittstellenproblem unter Beachtung der obengenannten Kriterien.

Vorentwicklung und Systementwicklung neuartiger Datenübertragungsverfahren und Einrichtungen.

2. Meßumformer:

Im Hinblick auf den Anschluß an den Digitalrechner mit seinen speziellen analogen und digitalen Eingängen und seiner Fähigkeit, Plausibilitätskontrollen und Fehlererkennungen durchzuführen, muß das Konzept der Meßumformer neu überdacht werden. Auf der einen Seite sollen sie Meßumformer zu konventionellen und in vielen Anlagen schon vorhandenen Geräten passen. Andererseits sollen die neuartigen Möglichkeiten und Anforderungen ausgenutzt und beachtet werden. Zum Beispiel kann es zweckmäßig sein, daß die Signale unmittelbar digital oder zumindest leicht digitalisierbar sind. Einrichtungen zur Fehleranzeige bzw. Plausibilitätskontrolle sollten eingebaut sein. Zum flexiblen Anschluß an verschiedene Rechnertypen oder auch Ringleitungen sollen die Signale einheitlich sein. Für Meßumformer verwendbare Netzversorgungen sind ebenfalls ein Gesichtspunkt der Entwicklung. Für den Einsatz von Prozeßrechnern in technischen Systemen hat die Frage der Betriebssicherheit Vorrang. In diesem Zusammenhang sollten an die Meßumformer höhere Anforderungen gestellt werden, als das bisher der Fall ist. Die Umweltbedingungen, unter denen die Meßumformer arbeiten, müssen stärker beachtet werden und es müssen besonders robuste betriebssichere Konstruktionen verfügbar sein. Da die Auswertung der Meßsignale im Rechner im allgemeinen eine sehr viel höhere Auflösung erlaubt, als das bei den konventionellen Meßeinrichtungen üblich ist, müssen gegebenenfalls die Signale erst für die Verarbeitung aufbereitet werden oder auch genauere Meßumformer entwickelt werden.

Vorschlag:

Studie über Anforderungen an Meßumformer und Schnittstellen.

Bestandsaufnahme der im Einsatz befindlichen und erforderlichen Meßumformer.
Einordnung in Klassen.

Aufzeigen allgemeiner Entwicklungstendenzen.

Angaben über voraussichtliche Marktentwicklungen.

Entwicklung rechnerangepaßter und neuartiger Meßumformer.

3. Sichere und redundante Strukturen

Dieser Gesichtspunkt trat bei der Diskussion aller Fragen auf, man kam zu dem Ergebnis, daß man deshalb dieses wichtige Thema im Rahmen einer besonderen Grundsatzstudie untersuchen sollte. In diesem Zusammenhang kamen noch einmal die grundsätzlichen Probleme der Datenübertragung, der Fehlererkennung und -Beseitigung, einer sicheren Inbetriebnahme und eines sicheren Betriebes unter harten Umweltbedingungen zur Sprache. Weitere Aufgaben zu diesem Thema sind:

- Standardisierung von Schnittstellen
- Dezentralisierung von Funktionen
- Back up
- Systematik der Interface-Strukturen
- Erfahrungsaustausch der Anwender.

Vorschlag:

Studie über die Möglichkeit, sichere, redundante Strukturen aufzubauen.

Aufzeigen von Möglichkeiten, in möglichst zweckmäßiger Weise Fehlerdiagnosesysteme einzusetzen.

Untersuchung der Möglichkeit einer weitgehenden Datenreduktion vor Ort.

4. Schnittstellen

Auch dieses Problem trat bei der Besprechung der einzelnen Komponenten immer wieder auf, so daß es als gesonderter Diskussionspunkt festgehalten werden soll. Hierbei geht es um Schnittstellen zwischen Rechnern, zwischen

Rechnern und standardisierter Peripherie oder anderen peripheren Geräten und zwischen peripheren Geräten und dem Prozeß. Ein besonderes Problem sind die unterschiedlichen Schnittstellen der einzelnen Hersteller. Ziel einer sinnvollen Entwicklung müßte die Schaffung von einigen wenigen "Klassen von Schnittstellen" sein. Hierbei sollen bereits vorhandene Normen und Standards berücksichtigt werden.

Vorschlag:

Anfertigung einer Studie zum allgemeinen Problem der Schnittstellen.

Untersuchung vorhandener Interface-Einrichtungen zur Koppelung unterschiedlicher Schnittstellen.

5. Planung und Auslegung

Die Planung ganzer Prozeßsysteme aus Prozeßbeschreibungen, Meßumformern, Datenübertragungssystemen, Programmsystemen, Rechnersystemen usw. ist nur möglich, wenn über die verfügbaren Meßumformer und Datenübertragungssysteme klare Unterlagen vorliegen. Diese Aufgabe ist relativ komplex, weil viele Möglichkeiten des Aufbaus einer Anlage denkbar sind. Eine grundsätzliche Entscheidung ist z.B.

 ober einer schnellen Datenübertragung und Auswertung im zentralen Rechner oder einer Auswertung und Datenreduktion in einem kleinen Rechner vor Ort und einer langsamen Übertragung zum zentralen Rechner der Vorzug zu geben ist.

Zu Entscheidungen dieser Art sind u.a. notwendig:

- Eine Gliederung des Gerätesystems mit einer übersichtlichen Definition der Funktion einzelner Geräte und Baugruppen und der Informationsübertragung an den verschiedenen Nahtstellen.
- Untersuchung der Wirtschaftlichkeit
- Untersuchung der Sicherheit (erhöhte Sicherheit kann trotz höherer Kosten erhöhte Wirtschaftlichkeit bedeuten, wenn ein Ausfall des Systems mit hohen Kosten verbunden ist).

Vorschlag:

Als Beitrag zum Entwurf ganzer Systeme sind grundlegende Studien über Anforderungsprofile, Entscheidungstechniken, Entwurfskriterien anzufertigen. Insbesondere sind für bestehende charakteristische Fälle Lösungen zu analysieren und neuartige Lösungen zu entwerfen, zu realisieren und zu erproben.

gez. Stams

gez. Kaltenecker

4. Fachgespräch am 22. 10. 1971
über "Kommunikationssysteme in Prozeßwarten".

Teilnehmerliste

für das Fachgespräche "Kommunikationssysteme in Prozeßwarten" am
22.10.1971 im Kernforschungszentrum Karlsruhe

Name:

Institution:

Syrbe	Fraunhofer-Gesellschaft, Karlsruhe
Geiser	Fraunhofer-Gesellschaft, Karlsruhe
Hofmann	Farbwerke Hoechst AG
Müller	KFA, Jülich
Dietsch	Erdölchemie, Köln
Zander	Hahn-Meitner-Institut, Berlin
Wölcken	Strahlencentrum Gießen
Nagy	Nixdorf Computer AG, Berlin
Miels	Nixdorf Computer AG, Berlin
Kimmel	Forschungsinstitut für Anthropo- technik, Meckenheim
Eberspächer	IBM
Klessmann	Hahn-Meitner-Institut, Berlin
Schmidt	MBP, Dortmund
Eichner	Universität Stuttgart
Arp	Universität Stuttgart
Tradowsky	GfK/LEM
Martin	BBC, Mannheim
Gehbauer	Universität Karlsruhe
Krüger	GfK/IDT
Kiersch	BBC, Mannheim
Schief	Fraunhofer-Gesellschaft
Baumann	Siemens AG, Karlsruhe
Baum	TH Aachen
Ammon	AEG-TELEFUNKEN, Frankfurt
Didic	Unicomp/DFE, Blankenloch
Zabler	Universität Karlsruhe
Hoermann	TU München
Sinzinger	Siemens AG, Erlangen
Ent	Daimler Benz AG
Kießling	Daimler Benz AG
Kaltenecker	Siemens AG, Karlsruhe

Feig

Frech

Hofmann

Benz

Salamon

Koehler

Schenk

Ottes

Stams

Eckert

GID, Aschaffenburg

Unicomp/DFE, Blankenloch

Siemens AG

Siemens AG

AEG-TELEFUNKEN

Hartmann u. Braun

DFVLR, Braunschweig

GfK/LEM

GfK/PDV

GfK/PDV

E r g e b n i s - P r o t o k o l l

zum Fachgespräch "Kommunikationssysteme in Prozeßwarten" am 22.10.71
im Kernforschungszentrum Karlsruhe

1. Mensch-Maschine-Schnittstelle

Trotz fortschreitender Automatisierung wird der Bedienungsmann von technischen Prozessen nicht aussterben. Insbesondere aus wirtschaftlichen Gründen wird er für Überwachungs- und Bedienungsfunktionen unentbehrlich sein. Da die Planung technischer Prozesse nur mit begrenztem Aufwand durchgeführt werden kann, sind unvorhergesehene Betriebszustände und Störungsfälle nicht auszuschließen. Für solche Fälle stellt der Mensch das wirtschaftlichste Element der Sicherung einer Anlage dar. Infolge der wachsenden Komplexität technischer Prozesse besteht bei Vernachlässigung der Mensch-Maschine-Schnittstelle die Gefahr, daß sie zum schwächsten Glied des Prozesses werden kann.

Durch den Einsatz von Prozeßrechnern zur Prozeßlenkung sind neue Geräte und Methoden zur Gestaltung der Mensch-Maschine-Schnittstelle in Prozeßwarten gegeben. Diese umfassen sowohl die Darstellung der für den Bedienungsmann notwendigen Prozeßinformationen als auch die Eingabe der Eingriffe des Bedienungsmannes in den Prozeßablauf. Um diese neuen technischen Möglichkeiten günstig, d.h. an den Eigenschaften des Menschen orientiert, einsetzen zu können, sind folgende Problemkreise zu untersuchen:

1.1. Anforderungen an die Mensch-Maschine-Schnittstelle

a) Anforderungen des technischen Prozesses an den Bedienungsmann.

Die Aufgaben des Bedienungsmanns sind unter Berücksichtigung des Automatisierungsgrades für die einzelnen Betriebsphasen zu beschreiben. Ferner ist zu prüfen, inwiefern verfahrenstechnische Gegebenheiten eine Rolle spielen (z.B. Raffinerie-Kraftwerk).

b) Anforderungen des Bedienungsmanns an die Prozeßwarte

Das Ziel der Wartengestaltung ist die Anpassung der Anzeige- und Bedienelemente an den Menschen. Hierzu sind Kenntnisse der Eigenschaften und der Struktur der Informationsverarbeitungsprozesse des Menschen erforderlich. Um zu übertragbaren Aussagen zu gelangen, sind analytische

Modelle der menschlichen Informationsverarbeitung zu entwickeln.

Zur Begründung dieser Vorgehensweise kann auf das Beispiel der Modellbeschreibung des Menschen als Regler verwiesen werden, die einen solchen Entwicklungsstand erreicht hat, daß sie z.B. bei der Auslegung der dynamischen Eigenschaften von Flugzeugen ein nützliches Hilfsmittel darstellt.

1.2. Erarbeitung von Regeln für die Gestaltung der Mensch-Maschine-Schnittstelle

Aufgrund der unter 1.1. genannten Untersuchungen können Regeln für die anthropotechnische günstige Wartengestaltung ermittelt werden, wobei Ein- und Ausgabeelemente für den Dialog zwischen Mensch und Maschine behandelt werden müssen.

2. Geräteentwicklung

Die Diskussion ergab, daß das Angebot an technischen Kommunikationsmitteln groß und in ständiger Weiterentwicklung ist, daß im Rahmen dieses Projektes man sich vor allem darauf konzentrieren sollte, die verfügbaren Geräte zum Anschluß an Prozeßrechner und zum Einsatz in Prozeßwarten anzupassen sowie die Zuverlässigkeit zu erhöhen. Zu beachten sind neben der Zuverlässigkeit: Informationsreduktion, Standardisierung von Schnittstellen und die Datenübertragung. Die allgemeine Entwicklung der Kommunikationsmittel soll beobachtet, der Einsatz in Prozeßwarten geprüft, vorbereitet und erprobt werden.

3. Rechnergestützte Warten

Die unter 1 und 2 entwickelten Ergebnisse sind Unterlagen für die Gestaltung rechnergestützter Bedienpulte und Warten. Die vorliegenden Erfahrungen, Aufgabenstellungen und Anforderungen müssen gesichtet, ausgewertet und nach Anforderungsklassen geordnet werden. Neuartige Möglichkeiten müssen prototypisch entwickelt, realisiert und erprobt werden.

4. Bilddarstellung

Bilddarstellungen ermöglichen eine besonders effektive dynamische Darstellung von Prozeßvorgängen. Es gibt verschiedene Möglichkeiten

zur Gestaltung des Bildaufbaues. Es sollen Untersuchungen darüber angestellt werden, welche Möglichkeiten hinsichtlich Aufgabenstellung und Wirtschaftlichkeit zweckmäßig sind. In diesem Zusammenhang wurde darauf hingewiesen, daß demnächst u.U. völlig neuartige Möglichkeiten der dynamischen Darstellung von Prozeßabläufen zur Verfügung stehen.

5. Programmierung von Prozeßrechnern zur Warterversorgung

Die Programmierung von Dialogsystemen, von graphischen Darstellungen und von Bedienabläufen in Prozeßwarten ist z.Zt. sehr aufwendig. Dieses Problem läßt sich nur dadurch lösen, daß man geeignete problemorientierte Programmiersprachen und standardisierte Programmpakete zur Verfügung stellt. Beide müssen unterstützt werden durch geeignete Funktionen des Betriebssystems und die Geräte selber. Auf diesem Gebiet laufen verschiedene Arbeiten. Siemens berichtet über Arbeiten an einer Bildbeschreibungssprache. Es wurde berichtet über die Entwicklung von Bildprogrammiersprachen, die als besondere Eigenschaft den Bildaufbau unter Zuhilfenahme des Bildschirms haben. Es sind die Anforderungen von der Bedienung her an die Bildprogrammiersprachen, an die Ein- und Ausgabe und an die Datenverwaltung zu klären. Davon ausgehend soll die Entwicklung und Erprobung von Compilern behandelt werden.

gez. Stams

gez. Syrbe

5. Fachgespräch am 4./5.11.1971

über "Systemanalyse, Systemsynthese, ~~System~~simulation".

Teilnehmerliste

für das Fachgespräch "Systemanalyse, Systemsynthese, Systemsimulation"
am 4./5. 11. 1971 im Kernforschungszentrum Karlsruhe

Name:

Institution:

Isermann	Universität Stuttgart
Schief	Fraunhofer-Gesellschaft
Adamczyk	Forschungsvereinigung Programmiersprachen, Aachen
Didic	Unicomp/DFE, Blankenloch
Hirschberg	Siemens AG, Karlsruhe
Sinzinger	Siemens AG, Erlangen
Kollmann	Hartmann u. Braun, Frankfurt
Wirsing	Gutehoffnungshütte
Mesch	Universität Karlsruhe
Schmidt	TU München
Scharf	MAN
Unbehauen	Universität Stuttgart
Ent	Daimler Benz AG
Müller	Rohde u. Schwarz, Karlsruhe
Frederking	Messerschmidt-Bölkow-Blohm
Berndorfer	Messerschmidt-Bölkow-Blohm
Rueff	GID, Aschaffenburg
Martens	Rohde u. Schwarz, München
Ammon	AEG-TELEFUNKEN, Frankfurt
Syrbe	Fraunhofer-Gesellschaft
Latzel	BEC, Mannheim
Thomas	DFVLR, Braunschweig
Budde	Forschungsvereinigung Programmiersprachen, Aachen
Binge	IBM, Sindelfingen
Hoermann	TU München
Goos	Universität Karlsruhe
Nagy	Nixdorf Computer AG, Berlin
Sell	Nixdorf Computer AG, Berlin
Krüger	GfK/IDT
Stams	GfK/PDV
Eckert	GfK/PDV

Schreitmüller	TH Aachen
Hotes	SCS, Hamburg
Jung	I.R.A., Frankfurt
Martin	BBC, Mannheim
Schuchmann	Farbwerke Hoechst AG
Neunheuser	SCS, Hamburg
Müller	Erdölchemie, Köln
Pott	Hartmann u. Braun, Heiligenhaus
Nolte	Gutehoffnungshütte
Tenten	KFA, Jülich
Eichner	Universität Stuttgart
Göhring	Universität Stuttgart
Reh	ESG, München
Eichenauer	ESG, München
Popović	Farbenfabriken Bayer
Scheurer	AEG-TELEFUNKEN
Reichert	Rohde und Schwarz, Karlsruhe
Rehr	TH Aachen
Essel	TH Aachen
Pavlik	Siemens AG, Karlsruhe
Klapdor	BBC, Mannheim
Balling	Hartmann u. Braun, Minden
Rohde	AEG-TELEFUNKEN, Konstanz
Lotte	MAN
Schmidt	MBP, Dortmund
Hofmann	Siemens AG
Kreuter	Siemens AG
Lauber	Universität Stuttgart
Bautz	ARG-TELEFUNKEN, Berlin
Gerresheim	PSI, Berlin
Zander	Hahn-Meitner-Institut, Berlin
Schwerdtner	IBM
Schmidt	Messerschmidt-Bölkow-Blohm
N.N.	Dornier
N.N.	MAN

E r g e b n i s - P r o t o k o l l

zum Fachgespräch "Systemanalyse und Prozeßführung" am 4./5.11.1971
im Kernforschungszentrum Karlsruhe

Die Auswertung des Einsatzes (Konkurrenzfähigkeit der Hersteller) und damit auch der Nutzung (Konkurrenzfähigkeit der Anwender) von Prozeßrechnern hängt nicht nur von der Leistungsfähigkeit der Geräte und der Leistungsfähigkeit der Programmierung ab, sondern auch von der Bereitstellung geeigneter Einsatzmethoden. Letztere wurden unter dem Stichwort "Systemanalyse, Prozeßführung" zusammengefaßt. In diesem Fachgespräch wurde dieses Gesamtgebiet aufgeteilt in drei Untergebiete

Prozeßanalyse, Aufgabenbeschreibung: zur Vorbereitung des Einsatzes,
Lösungsmethoden, Auswahlkriterien: zur Synthese der Einsatzmethode,
Simulation, Test: zur Überprüfung und Erprobung der Einsatzmethode.

1. Prozeßanalyse, Aufgabenbeschreibung

Die durch Prozeßrechner gegebenen Möglichkeiten der Prozeßlenkung lassen sich nur dann wirkungsvoll und wirtschaftlich ausnutzen, wenn man über ein zutreffendes Modell des zu steuernden Prozesses verfügt. Dieses Modell wird sowohl im Planungsstadium zur Auslegung der DVA benötigt als auch bei Optimierungs- und Adaptionsaufgaben während der Prozeßsteuerung. In der Vergangenheit wurden verschiedene Methoden zur Prozeßanalyse und Prozeßidentifikation vorgeschlagen. In der Praxis konnten bisher nur relativ wenige Methoden wirklich hinreichend erprobt werden. Deshalb ist eine besonders wichtige Forderung die Anwendung der bereits bekannten Methoden und Überprüfung der Leistungsfähigkeit. An Methoden wurden im einzelnen genannt:

- Bestimmung der Prozeßvariablen und ihrer funktionellen Abhängigkeiten aus der Kenntnis der Technologie des Prozesses,
- On-line-Messungen mit Off-line-Identifikation eines Prozesses.
- On-line-Identifikation, d.h. Verifikation und Entwicklung eines Modells durch on-line-Prozeßsteuerung und Messungen
- Prozeß-Simulation

Weitere Aufgaben im Rahmen der Prozeßanalyse und Aufgabenbeschreibung sind:

- Aus der Prozeßanalyse und -Aufgabenbeschreibung sollen Anforderungen an Rechner, Prozeßperipherie, Kommunikationseinrichtungen und Programmiersysteme hergeleitet werden.
- Ein Katalog von Prozeßmodellen mit Anwendungstypen und Anwendungsmethoden, mit Bewertungen und Vergleichen (Modelldatenbank).
- Projektierungshilfsmittel, Methoden zur formalisierten Beschreibung von Prozeßkomponenten.
- Entwicklung einer Entwurfsmethodik.
- Ausfalldiagnose, Auswertung von Meßwerten zur selbsttätigen möglichst auch frühzeitigen Erkennung von Ausfallursachen.

Bei allen genannten Aufgaben sollen folgende Gesichtspunkte beachtet werden:

- Schwerpunkt nichtlineare Prozesse, verteilte Parameter und Mehrgrößenregelung
- Grundlegendes Kriterium für jeden Prozeßrechnereinsatz ist die Wirtschaftlichkeit
- Sicherheit und Redundanz in sämtlichen Ebenen, Benutzerfreundlichkeit

2. Lösungsmethoden, Auswahlkriterien

Beim bisherigen Einsatz von Prozeßrechnern zur Prozeßlenkung wurden bereits einige Methoden entwickelt und zum Einsatz gebracht:

- Digitale Vielfachregelung

Einsatzbeispiel: Energieversorgung, Werkzeugmaschinensteuerung.
Die Probleme wurden ganz eng praxisbezogen gelöst.

Forderung:

Aufstellung eines Katalogs von Algorithmen mit Bewertung und Vergleich, Fortsetzung der Erprobung.

- Optimierungsmethoden

In der Literatur sind eine Vielzahl von Optimierungsmethoden beschrieben. Sie wurden bisher noch nicht umfangreich getestet.

Forderung:

Katalog mit Leistungsvergleich und Kriterien. Entwicklung von vereinfachten Methoden für mittlere und kleinere Rechner Typen.

- Adaptive Methoden

Die adaptive Prozeßführung kann wesentliche technische und wirtschaftliche Fortschritte bringen, sie setzt jedoch die On-line-Identifikation voraus. Beispiele sind die adaptive Regelung der Drehmaschine und die Anwendung bei der Papierherstellung.

Forderung:

Weiterentwicklung dieser Möglichkeit anhand von ausgewählten praktischen Beispielen. Leistungsvergleich mit parameterunempfindlichen Verfahren.

- Prädiktive Regelung

Durch geeignete Programme ist die vorausschauende Berücksichtigung von Störsignalen möglich. Hieraus lassen sich eine bessere Regelung durch Vorhersagen über das zukünftige Verhalten eines Prozesses ableiten. Das läßt auch eine Vorhersage von Ausfällen einzelner Prozeßkomponenten zu.

Forderung:

Die Verfahren und Methoden zur Vorhersage stochastischer Signale, Regelungsverfahren und Lokalisierung von Fehlern sollen erforscht und für spezielle Beispiele erprobt werden.

- Mehrgrößenregelung

Die Mehrgrößenregelung ohne Prozeßrechner ist im wesentlichen nur bei herkömmlichen linearen Systemen möglich. Systeme mit verteilten Parametern und/oder nichtlineare Systeme lassen sich mit den gängigen Methoden der Regelungstechnik nur noch schwer oder nicht mehr beherrschen. Dies könnte ein besonders wichtiges Anwendungsgebiet für Prozeßrechner werden. Bereits bei der Identifikation derartiger Systeme ist der Prozeßrechner wegen der großen Fülle des Datenmaterials unbedingt erforderlich. Noch mehr natürlich bei der praktischen Ausführung von Mehrgrößenregelungssystemen.

3. Simulation, Test

Bisher ist es üblich, komplette Systeme mit Rechner, Prozeßperipherie und Prozeß aufzubauen und dann mit Programmentwicklung und Test zu beginnen. Das ist schwerfällig, unwirtschaftlich und riskant. Dieses Verfahren könnte entscheidend verbessert werden, wenn man den Prozeßrechner und die einzelnen Prozeßkomponenten simulieren könnte.

Forderung:

Aufbau von Emulationsrechenverfahren mit folgenden Möglichkeiten:

- Simulation von beliebigen Rechnerkonfigurationen.
- Aufbau von Dialog- und Protokollsystemen zur Verfolgung, Überwachung und Steuerung der Simulation, insbesondere zur Änderung von Anwenderprogrammen

- Simulation von Prozeßkomponenten
- Anschluß nichtsimulierter Komponenten
- Verfahren zum schrittweisen Aufbau von Anlagen
- Verfahren zum Aufbau dieser Simulationsprogramme
- Erprobung der Emulationstechnik in Form von Fallstudien
- Untersuchung der Wirksamkeit und Wirtschaftlichkeit des Verfahrens
- Einsatz dieses Verfahrens zum Vergleich verschiedener Rechnerstrukturen für bestimmte Aufgaben.

Dozent Dr.-Ing.R.Isermann
Fachgebiet Regelungstechnik
im Fachbereich Energietechnik

Universität Stuttgart

7 Stuttgart -1

Keplerstrasse 17

4.11.71

EINSATZ VON PROZESSRECHNERN ZUR PROZESSANALYSE

(Referat zum Fachgespräch 2.DV-Programm am 4.11.71)

Die Automatisierung von technischen Prozessen spielt sich bekanntlich in mehreren Ebenen ab, Bild 1. In der untersten Ebene werden bestimmte Prozessgrößen geregelt (bzw. gesteuert). In der zweiten Ebene wird der Prozess überwacht und optimiert. Wenn mehrere Prozesse zusammenwirken, dann müssen diese in einer dritten Ebene koordiniert werden. Schliesslich kann man in einer vierten Ebene das Management aller Prozesse zusammenfassen, wo die Anpassung des technischen "Gesamtprozesses" z.B. an den Markt, an die Planung und an die Verfügbarkeit von Rohprodukten und Personal erfolgt.

Prozessrechner werden zur Zeit hauptsächlich in den beiden untersten Ebenen eingesetzt, gelegentlich auch in der dritten Ebene. Aufgaben der vierten Ebene werden heute nur teilweise und dann off-line mit Digitalrechnern ausgeführt.

Zur sinnvollen Lösung vieler Aufgaben bei der Automatisierung und insbesondere zur zweckmässigen Programmierung von Prozessrechnern braucht man mathematische Modelle für das Verhalten der Prozesse. Diese mathematischen Modelle müssen für die Anwendung in der Ebene 1 das dynamische Verhalten des Prozesses beschreiben. Für die anderen Ebenen 2,3 und 4 sind meist nur Modelle für das statische Verhalten erforderlich.

Zur Ermittlung dieser mathematischen Modelle gibt es nun zwei verschiedene Wege, die theoretische und experimentelle Analyse, Bild 2.

Bei der theoretischen Analyse wird das Modell berechnet. Man beginnt mit vereinfachenden Annahmen über den Prozess, stellt dann die Bilanzgleichungen für die gespeicherten Massen, Energien und Impulse auf, führt physikalische Zustandsgleichungen und bei irreversiblen Prozessen die phänomenologischen Gesetze (z.B. Wärmeleitung, Diffusion, chemische Reaktion) ein. Wenn mehrere irreversible Prozesse auftreten, können zusätzlich Entropiebilanzgleichungen verwendet werden. Diese Gleichungen bilden dann ein simultanes Differentialgleichungssystem, das, wenn es sich für bestimmte Ein- und Ausgangsgrößen des Prozesses explizit lösen lässt, auf ein theoretisches Modell des Prozesses mit bestimmter Struktur und bestimmten Parametern führt. Vielfach ist dieses Modell umfangreich und kompliziert. Dann muss es für weitere Anwendung vereinfacht werden. Man erhält dann ein vereinfachtes mathematisches Modell.

Bei der experimentellen Analyse wird das mathematische Modell des Prozesses durch Messungen ermittelt. Man geht hierbei stets von a-priori-Informationen über den Prozess aus, die z.B. aus der theoretischen Analyse oder aus vorhergegangenen Messungen gewonnen werden. Ist z.B. bekannt, dass der Prozess lineares oder näherungsweise lineares Verhalten besitzt, dann verwendet man bei unbekannter Struktur des Prozessmodells die direkten Identifikationsmethoden, bei bekannter Struktur die Modell-Identifikationsmethoden. Bei ausgeprägt nichtlinearem Verhalten muss die Struktur zur Identifikation des Prozesses meist als bekannt vorausgesetzt werden.

Man erhält dann als Ergebnis ein experimentelles Modell, das eventuell ebenfalls vereinfacht werden muss, wenn es zu umfangreich ist.

Die Wahl des resultierenden Prozessmodells hängt vom Anwendungszweck ab. Soll das Prozessmodell z.B. dazu verwendet werden, um einen Prozess bezüglich seines dynamischen Verhaltens günstig auszulegen, bzw. zu konstruieren, dann wird man das theoretische Modell vorziehen, da es den funktionalen Zusammenhang zwischen

den physikalischen Daten des Prozesses und seinem dynamischen Verhalten enthält. Soll das Prozessmodell dagegen z.B. zur Anpassung eines Regelalgorithmus für verschiedene Betriebspunkte verwendet werden, dann zieht man i.a. das experimentelle Modell vor, wenn es das momentane dynamische Verhalten genauer beschreibt.

Im ersten Fall liegt der Schwerpunkt auf der theoretischen Analyse. Man verwendet dann die experimentelle Analyse bei komplizierten Prozessen zur Nachprüfung der Genauigkeit des theoretisch ermittelten Modells (Bild 2: → Vergleich: wenn Fehler → Änderung der Annahmen oder der Vereinfachung).

Im zweiten Fall dagegen liegt der Schwerpunkt auf der experimentellen Analyse. Man verwendet jedoch vorteilhaft möglichst viel a-priori-Information aus der theoretischen Analyse. Aus diesen Beispielen ist zu sehen, dass eine zweckmässige Prozessanalyse durch eine geeignete Kombination theoretischer und experimenteller Verfahren erfolgt.

Der Anwendungszweck des Prozessmodells bestimmt ferner die erforderliche Genauigkeit des Modells, welche wiederum auf die theoretische oder experimentelle Analyse zurückwirkt, Bild 3. Prozessanalyse ist im allgemeinen ein iterativer Vorgang.

Prozessrechner können sehr vorteilhaft zur Ermittlung von mathematischen Modellen für das dynamische und statische Verhalten des Prozesses auf experimentellem Wege, also zur Prozessidentifikation, eingesetzt werden, z.B. zur Synthese eines Regelalgorithmus oder zur Optimierung des Prozesses oder einfach zum Studium des Prozessverhaltens.

Die Auswahl der Identifikationsverfahren hängt wiederum vom Anwendungszweck des identifizierten Modells ab. Abhängig vom Anwendungszweck lassen sich verschiedene Forderungen an das Identifikationsverfahren stellen, Bild 4. Während zum Studium des Prozessverhaltens die Messung am offenen Regel-

kreis, ein nichtparametrisches Modell und eine off-line Auswertung ausreicht, ist für eine selbstadaptive Regelung die Messung im geschlossenen Regelkreis, ein parametrisches Modell und on-line Auswertung erforderlich.

In den letzten Jahren sind nun sehr viele Identifikationsverfahren vorgeschlagen worden. Diese lassen sich, Bild 5, in direkte Messverfahren (Auswertung durch Fourieranalyse und Korrelation) und Modell-Messverfahren (Modellabgleich- und Parameterschätzverfahren) unterteilen. Dabei nimmt die erforderliche a-priori-Information, Bild 5, von oben nach unten zu.

Bisher sind nur sehr wenige dieser Verfahren auf Prozessrechnern eingesetzt worden. Es fehlt insbesondere ein Leistungsvergleich der verschiedenen Identifikationsverfahren untereinander und Erfahrungen beim praktischen Einsatz an technischen Prozessen. Besonders die Parameterschätzverfahren, die zur Synthese von Regelalgorithmen gut geeignet sind, sind bisher in bezug auf Speicherplatzbedarf und Rechenzeit zu aufwendig. Ferner steht die Identifikation nichtlinearer und zeitvarianter Prozesse ganz am Anfang einer Entwicklung.

Aus diesen Betrachtungen zur Prozessanalyse ergeben sich folgende Aufgaben:

EXPERIMENTELLE ANALYSE:

1. Praktischer Einsatz von Identifikationsverfahren an industriellen Prozessen. Entwicklung prozessrechnergeeigneter Verfahren.
2. Identifikation von Signalen industrieller Prozesse (Störsignal-Identifikation)

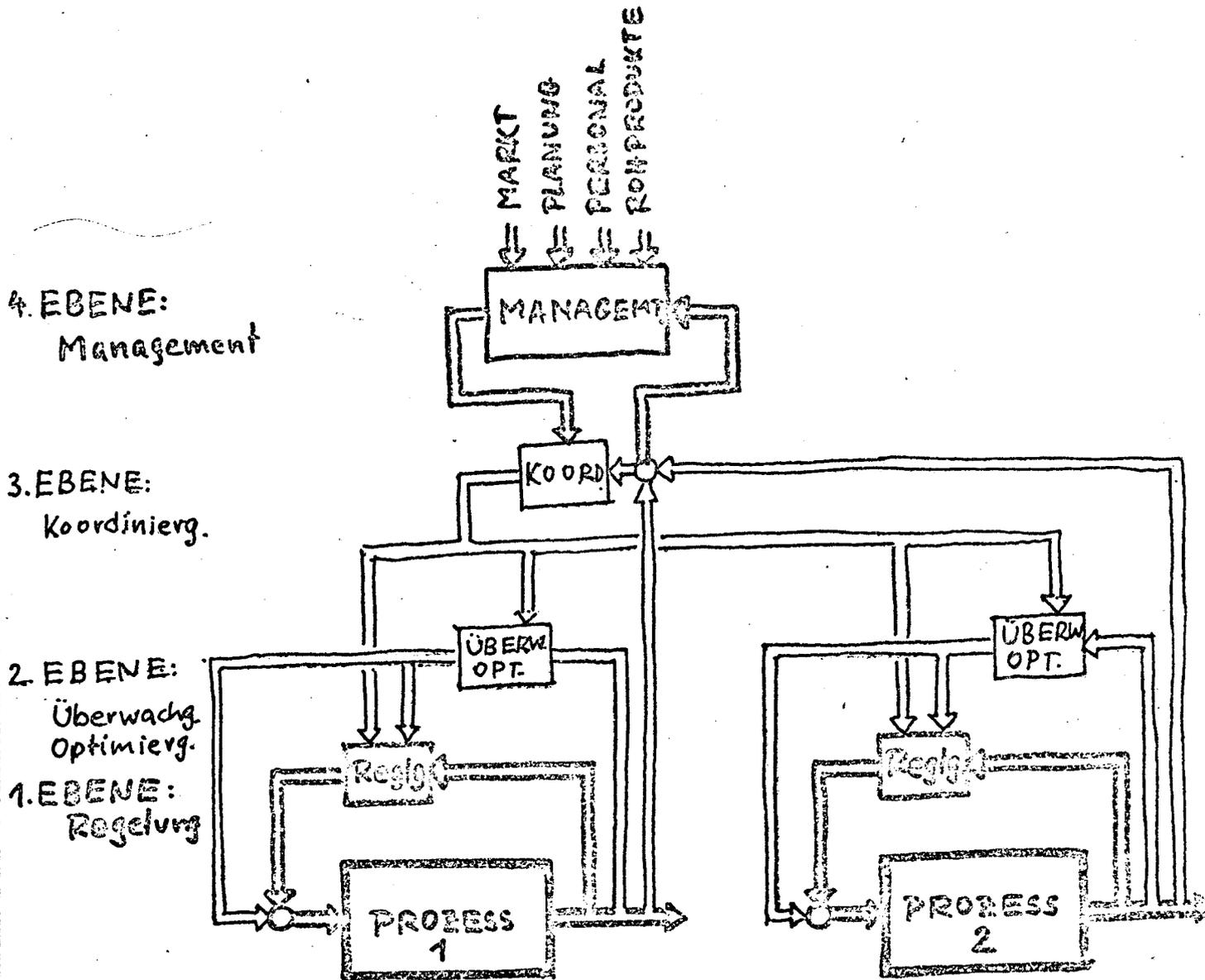
THEORETISCHE ANALYSE:

3. Aufstellen von Prozessmodellen für einzelne technische Prozesse
4. Sammlung von Prozessmodellen, die häufig verwendet werden, in einem Prozessmodell-Katalog
5. Theorie des statischen und dynamischen Verhaltens technischer Prozesse

Dabei sind alle Aufgaben in bezug auf den praktischen und wirtschaftlich vertretbaren Einsatz von Prozessrechnern zu bearbeiten.

LITERATUR:

- /1/ Aström, K.; Eykhoff, P.: Survey on Identification and Parameter-Estimation. IFAC-Symposium Prag 1970
- /2/ Isermann, R.: Neue Ergebnisse bei der Identifikation von Prozessen. Regelungstechnik 18 (1970) S. 508/512 und IFAC-Automatica Vol.7 (1971) S. 191/197
- /3/ Isermann, R.: Experimentelle Analyse der Dynamik von Regelsystemen (Identifikation I)
Bibliographisches Institut, Mannheim, 1971
- /4/ Isermann R.: Theoretische Analyse der Dynamik industrieller Prozesse (Identifikation II)
Bibliographisches Institut, Mannheim, 1971



PROZESSFÜHRUNG IN MEHREREN EBENEN

PROZESS - ANALYSE

THEORETISCHE ANALYSE

EXPERIMENT. ANALYSE

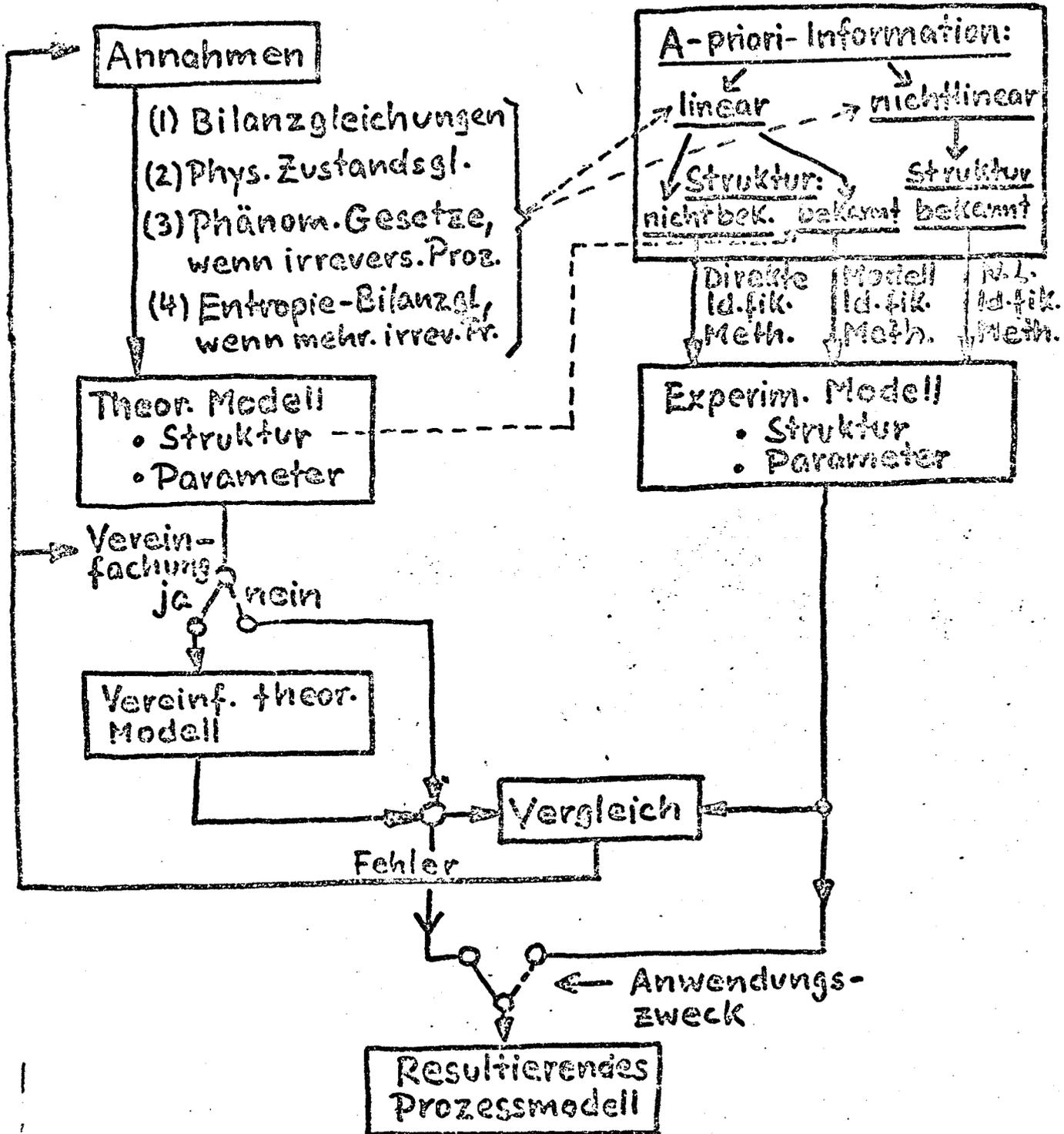
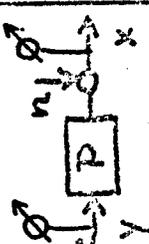
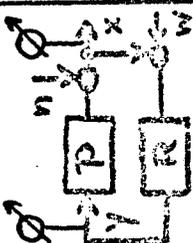
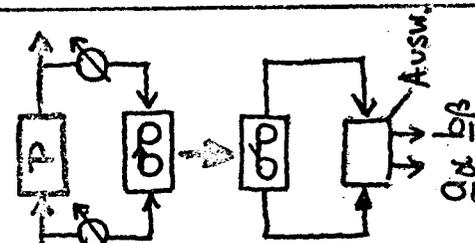
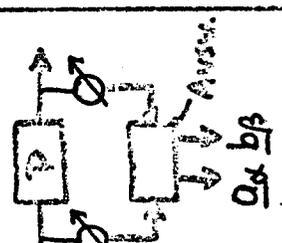


Bild 2

ZWECK der Identifikation	MESSUNG am		MODELL		AUSWERTUNG	
	off. R. Kreis	geschl. R. Kreis.	nichtparam.	parametr.	off-line	on-line
			$F(i\omega) = \frac{\omega F \alpha}{\vdots \vdots \vdots \vdots}$	$F(i\omega) = \frac{\sum b_n (i\omega)^n}{\sum a_n (i\omega)^n}$		
1. Vgl. theor. u. exp. math. Modell	X	(X)	X	—	X	—
2. Synthese einer Regelg. od. Steuerung	X	(X)	(X)	X	X	X
3. Adaptive Regelg.	—	X	(X)	X	—	X

DIREKTE M.V.	MESSVERFAHRS.	KENNZEICHNUNGEN	SIGNALE	nichtparametrisches Modell	parameter
Fourieranalyse		a) linear b) zeitinvar.	- nicht per. - periodisch	$F(i\omega)$, $0 < \omega < \omega_m$ <small>APPROX.</small>	$F(i\omega) = \frac{\sum b_k (i\omega)^k}{\sum a_k (i\omega)^k}$
Korrelation		a) linear b) zeitinvar.	- stochest. - pseudost.	$g(t)$ $F(i\omega)$ <small>APPROX.</small>	" "
Modellabgleich (analog)		a) Struktur b) linear od. nichtlinear	beliebig	—	$\sum_{k=1}^m a_k x^{(k)} = \sum_{k=1}^m b_k y^{(k)}(t)$
Parameterschätzung (digital)		a) Struktur b) linear v. Per. abh. c) zeitinvar.	beliebig	—	$\sum_{i=1}^m a_i x(k-i) = \sum_{i=1}^m b_i y(k-i) + e(k)$ $A(z^{-1}) x(k) = B(z^{-1}) y(k) + e(k)$ $G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}$
Maximum likelihood bzw. verallgem. Kleinstquadr. Fehler		a) Struktur b) e linear v. Per. abh. c) zeitinvar. d) $e(k)$: normalverteilt	beliebig	—	" "

Prozeßrechnereinsatz im Bereich der Fertigungstechnik

Bericht aus dem Laboratorium für Werkzeugmaschinen und Betriebslehre der Technischen Hochschule Aachen.

Dr.-Ing. H.J. Schreitmüller

Dipl.-Ing. K. Essel

Dipl.-Ing. W. Rehr

1. Einleitung

Der noch vor kurzem als Schlagwort benutzte Begriff "automatische Fabrik" kann heute bereits in einigen Teilbereichen der Fertigung als verwirklicht angesehen werden. So sind z. B. auf dem Gebiet der Energie- und Verfahrenstechnik Anlagen im Einsatz, deren Erzeugungsprogramm unter dem Einsatz elektronischer Datenverarbeitungsanlagen vollautomatisch erstellt wird.

Diese völlig neuen Möglichkeiten sind durch die stürmische Entwicklung der Computer und Computer-Techniken, die das wirtschaftliche Verarbeiten und Aufbereiten großer Datenmengen erlauben, in den letzten Jahren in alle Bereiche der Produktionstechnik vorgedrungen.

In Verbindung mit dem vermehrten Einsatz von numerisch gesteuerten Werkzeugmaschinen im Bereich der Fertigung sowie der marktwirtschaftlichen Lage, gekennzeichnet durch die sprunghaft ansteigenden Löhne bei zunehmendem Arbeitskräftemangel, kann heute bereits mit Sicherheit gesagt werden, daß die Firmen, die diese Computertechniken konsequent auszunutzen verstehen, die besten Voraussetzungen für eine leistungsfähige und konkurrenzfähige Produktion mitbringen werden.

Vom Standpunkt der Unternehmen hat dieser Automatisierungsprozeß das Ziel, Personalkosten durch Kapitalkosten in der Weise zu ersetzen, daß die Herstellungskosten pro Einheit verringert werden oder eine höhere Qualität der Produkte erreicht wird.

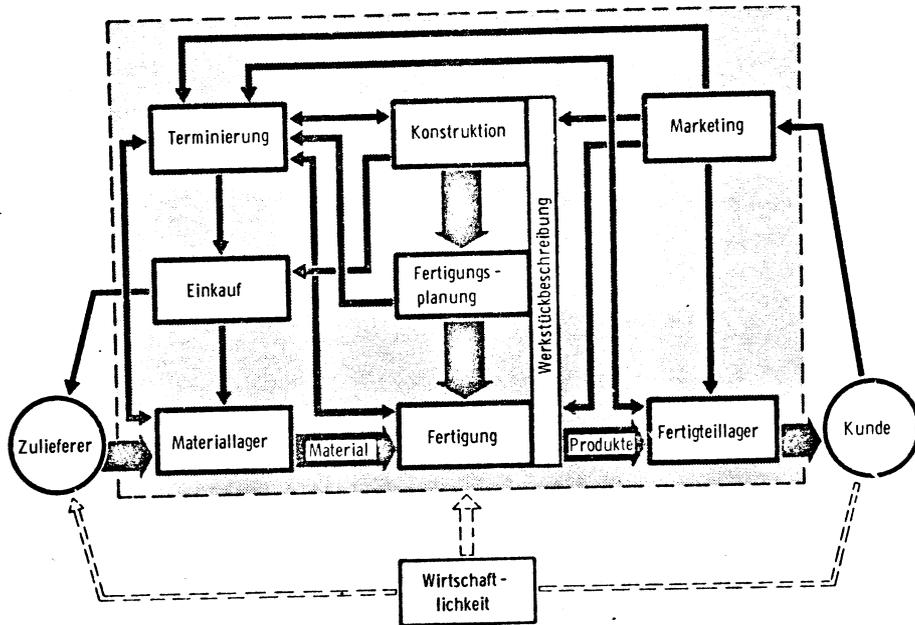


Bild 1: Verknüpfung der Unternehmensbereiche

Unter Berücksichtigung dieser Gesichtspunkte ist es sinnvoll, einen Betrieb in eine horizontale und vertikale Struktur zu unterteilen. (Bild 1). Die horizontale Ebene dokumentiert den Materialfluß vom Anlieferer bis zum Kunden.

Der vertikale Informationsfluß umfaßt die Konstruktion, Fertigungsplanung und die Fertigung selbst. Mit Erfolg werden bereits EDV-Anlagen in vielen Bereichen der Produktion voll eingesetzt und genutzt. In der Fertigung als Knotenpunkt der Produktion wurden bisher lediglich Einzelprobleme für den Computer so aufbereitet, daß sie mehr oder weniger losgelöst vom Gesamtkonzept einer Vollautomatisierung betrieben werden können.

2. Integrierte Fertigung

Ziel der gesamten Bestrebungen muß es jedoch sein, eine automatisierte und integrierte Fertigung, wie an einem Beispiel im Bild 2 dargestellt ist, zu erarbeiten. Nur so kann die Fertigung nach einer "automatischen Fabrik", die alle Bereiche der Produktion, insbesondere auch die Fertigung umfaßt, realisiert werden.

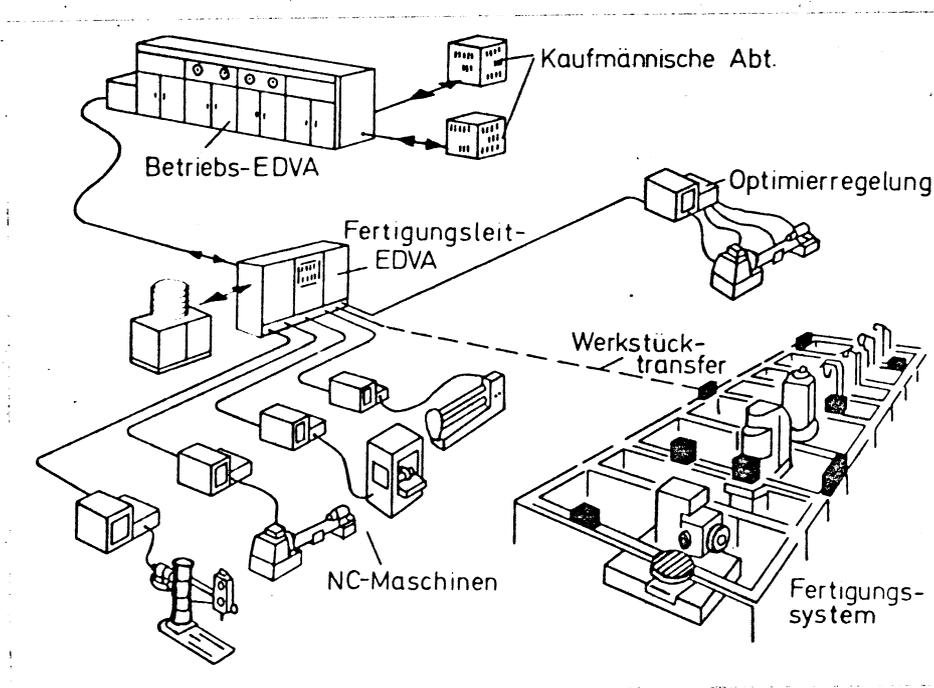


Bild 2: Automatisierung der Fertigung durch Rechnereinsatz

Der Betriebsrechner, im Bild oben links, an den alle Abteilungen des Betriebes angeschlossen sind, übernimmt die Durchführung der Planungsaufgaben. Die entsprechend erstellten Daten werden an den Fertigungsleitrechner übergeben, der mit großen externen Speichergeräten, wie zum Beispiel dem Plattenspeicher, ausgerüstet ist. An den Fertigungsleitrechner wird einmal das Fertigungssystem angeschlossen, das für jeden einzelnen Arbeitsschritt die entsprechenden Steuerinformationen erhält.

Die NC-Maschinen andererseits werden vom Fertigungsleitreehner über Erfassungsgeräte gesteuert, in die auch wiederum Betriebsdaten für den Rechner eingegeben werden können. Bei den on-line gesteuerten Maschinen mit adaptiver Regelung übernimmt der Rechner die Verarbeitung der während des Bearbeitungsvorganges erfaßten Meßgrößen und die entsprechende Steuerung der Maschine nach optimalen Zielfunktionen.

Besonders die Lösung der Probleme in diesem technischen Bereich hat gezeigt, daß durch die Erstellung und durch das Austesten neuer Programme hohe Kosten verursacht werden. Daraus läßt sich die Forderung nach generellen, von bestimmten Gewohnheiten einzelner Firmen unabhängigen Lösungen ableiten. Ein Beispiel hierfür ist die fertigungs-technisch-orientierte Programmiersprache EXAPT, die in Deutschland in enger Zusammenarbeit mit Forschungsinstituten und der Industrie entstand.

3. Problemstellungen

Das folgende Bild 3 zeigt schwerpunktartig die aktuellen Problemstellungen auf, die einer Lösung bedürfen.

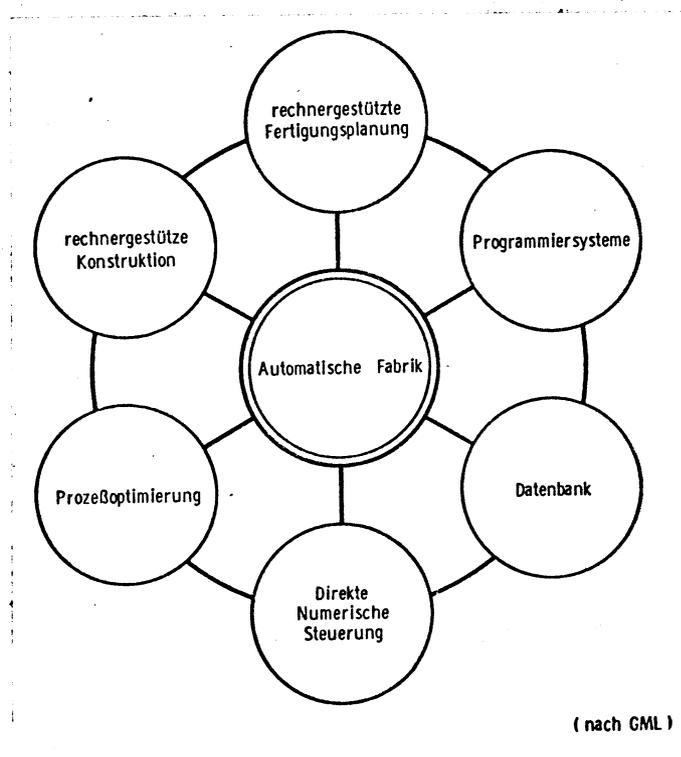


Bild 3: Aktuelle Entwicklungsschwerpunkte zur Realisierung der automatischen Fabrik

Die Notwendigkeit zur Realisierung dieses Projekts ist von den Hochschulen wie auch der Industrie erkannt worden. So befassen sich heute bereits eine Anzahl von Forschungsstellen mit einzelnen der hier angesprochenen Problemkreise. Zu erwähnen sind die Arbeiten im Rahmen der Erstellung einer Datenbank für Schnittwerte, die im Laboratorium für Werkzeugmaschinen und Betriebslehre der TH Aachen seit einigen Jahren mit Unterstützung der Industrie durchgeführt werden. Andere verschiedene Einzelvorhaben wie die rechnergestützte Fertigungsplanung, die Erstellung von Programmiersystemen, die direkte Numerische Steuerung von Werkzeugmaschinen und die Prozeßoptimierung wurden bereits in Teilbereichen in Angriff genommen.

4. Direkte Steuerung von Werkzeugmaschinen

In USA und Japan aber auch in Deutschland wurde gleichzeitig in jüngster Zeit eine Anzahl Systeme direkt von einem Rechner gesteuerter Werkzeugmaschinen entwickelt. Sie wurden in Anlehnung an directly numerically controlled als DNC-Systeme bekannt. Man versteht darunter Steuerungssysteme, bei denen die Versorgung einer Anzahl von Steuerungen (z.B. Werkzeugmaschinensteuerungen) durch Steuerinformationen schritthaltend mit dem Bearbeitungsprozeß über die direkte Kopplung mit einem Rechner erfolgt. Im allgemeinen werden hierfür Prozeßrechner eingesetzt, da diese über die notwendigen leistungsfähigen Ein- und Ausgabemöglichkeiten digitaler und analoger Signale verfügen.

Obwohl durch die Einführung der numerischen Steuerung (NC) und die Entwicklung fertigungstechnisch orientierter Programmiersprachen ein hoher Automatisierungsgrad in der Einzel- und Kleinserienfertigung erreicht wurde, erwartet man von der DNC-Steuerung weitere Vorteile. Folgende Gesichtspunkte müssen dabei hervorgehoben werden:

Verbesserung des Informationsflusses

Verkürzung der Rüstzeiten

Höhere Maschinenauslastung

Erhöhung der Transparenz des Betriebsgeschehens durch

Betriebsdatenerfassung

Möglichkeit des Aufbaus von Rechnerhierarchien

Im Bild 4 ist das im Laboratorium für Werkzeugmaschinen und Betriebslehre der TH Aachen entwickelte DNC-System mit konventionellen NC-Steuerungen, Rumpfsteuerungen und Steuerrechner (CNC) dargestellt.

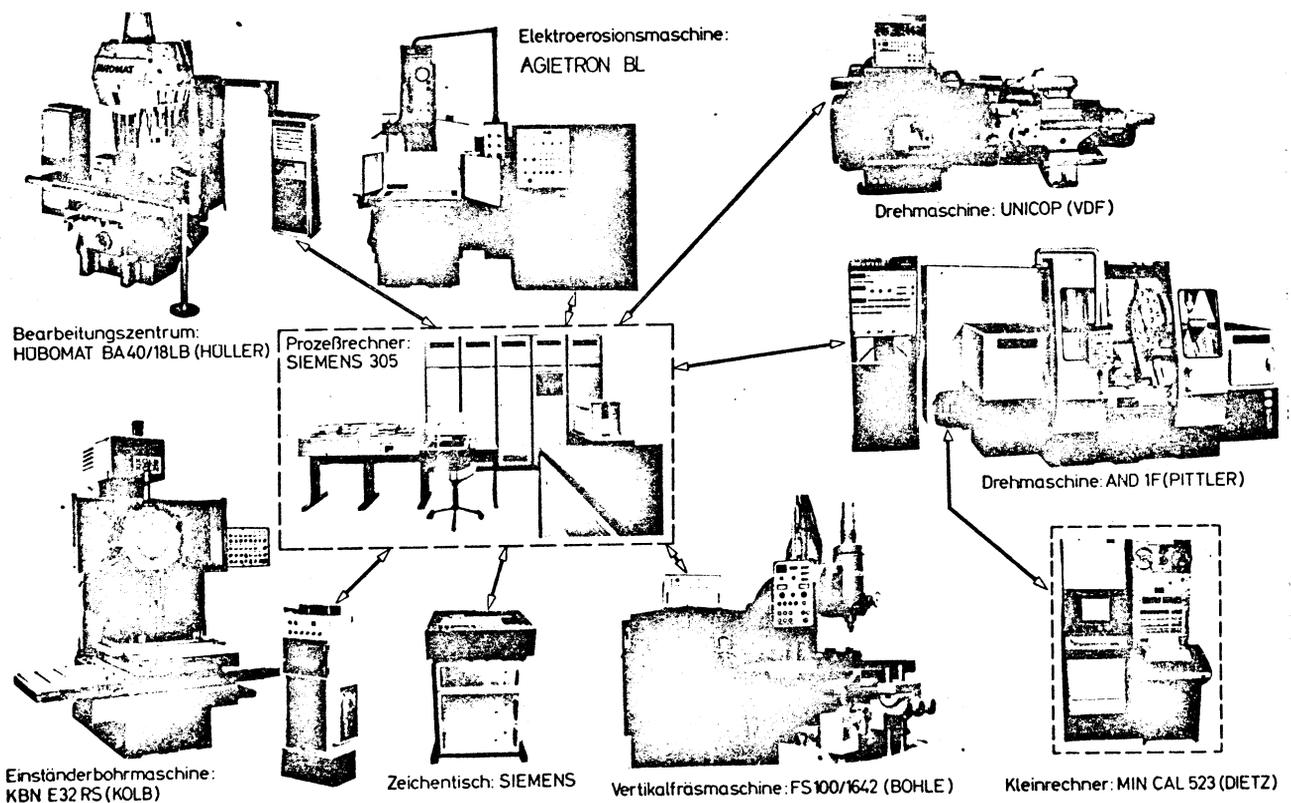


Bild 4: Direkte numerische Steuerung von Werkzeugmaschinen (DNC).

Die Entwicklung dieser hochwertigen Steuereinheiten erlaubt einerseits den wechselseitigen Datenaustausch zwischen Rechner und Steuerung, andererseits aber auch das Abrufen von Zustandsdaten der Steuerung, der Maschine und des Bearbeitungsprozesses. Eine Überwachung des Systems "Werkzeugmaschine" einschließlich Steuerung, Werkzeug und Werkstück kann mit einem solchen System verwirklicht werden. Die rechnergekoppelten Steuerungen sind vorzüglich geeignet, in Verbindung mit Sensoren charakteristische Prozeßdaten aufzunehmen und dem Rechner zur Verfügung zu stellen.

Die Komponenten eines DNC-Systems sind im Bild 5 zusammengefaßt dargestellt.

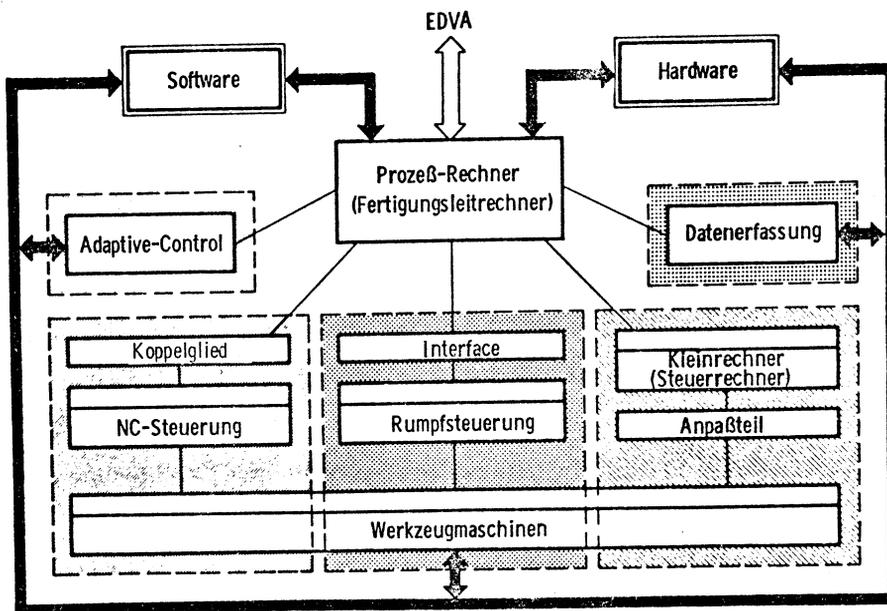


Bild 5: Komponenten eines DNC-Systems

Neben der direkten Steuerung von Werkzeugmaschinen mit einem Prozeßrechner besteht die Möglichkeit, durch den Einsatz von Adaptive-Control-Systemen (AC) eine optimale Führung des Bearbeitungsprozesses und damit eine Steigerung der Produktivität der Fertigungseinrichtungen zu erreichen.

Die zu den AC-Systemen zählenden Optimierungregelungen haben zum Ziel, den Bearbeitungsprozeß im Hinblick auf eine das gewünschte Fertigungsziel charakterisierende Größe zum bestmöglichen Ablauf zu führen. Die Vorgabe der Sollwerte oder ggf. eine Veränderung der Reglerparameter erfolgt dabei aufgrund der Ergebnisse einer Optimierungsrechnung. Diese Rechnungen sind im allgemeinen recht umfangreich und können praktisch nur mit Hilfe von Rechnern durchgeführt werden.

Als Beispiel einer Prozeßerkennung und anschließenden Prozeßoptimierung sei die im vorgestellten DNC-System integrierte Elektroerosionsmaschine genannt. Nach anfänglichen off-line Untersuchungen zur Identifikation der den Abtrag beeinflussenden Prozeßparameter konnten durch den Prozeßrechnereinsatz die Zusammenhänge zwischen den Parametern Abtragsleistung, Wirkverhältnis, mittlere Arbeitsspannung, Tastverhältnis, Impulsstrom und Spüldruck nachgewiesen werden. Eine Bewertung der Einflußgrößen auf die Zielgröße Abtragsleistung legte eine Teiloptimierung mit nur zwei Meßgrößen und einer Stellgröße nahe. Anhand eines empirischen Prozeßmodells, dem der Zusammenhang zwischen Wirkverhältnis und mittlere Arbeitsspannung zugrunde liegt, konnte ein stabiler Ablauf des Prozesses und eine teilweise Steigerung des Abtrages um 20 % gegenüber dem Handbetrieb erreicht werden.

Das Blockschaltbild der rechnergeregten Erosionsanlage ist in Bild 6 dargestellt.

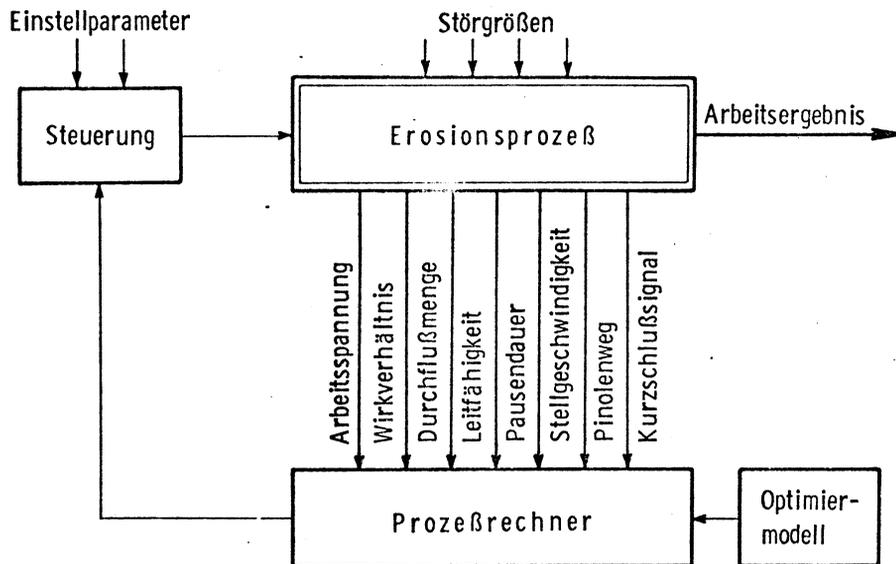


Bild 6: Optimierung des Erosionsprozesses

5. Entwicklung von Optimierregelungen

Im Bereich der Fertigungstechnik steht die Entwicklung von Systemen zur automatischen Führung und Überwachung von Fertigungsprozessen mit Hilfe eines Prozeßrechners (AC-Systeme) noch in den Anfängen.

Die Vielzahl nichtlinearer Einflußgrößen auf den Bearbeitungsprozeß hat es bisher verhindert, durch funktionale Verknüpfung der Bearbeitungsparameter den Prozeßverlauf zuverlässig vorherzubestimmen. Eine wesentliche Leistungssteigerung ist durch den Einsatz von Adaptive-Control-Systemen (AC) zu erzielen, die durch die automatische Änderung der Parameter des Fertigungssystems den Bearbeitungsprozeß nach vorgegebenen Kriterien beeinflussen und zum optimalen Ablauf führen. (Bild 7).

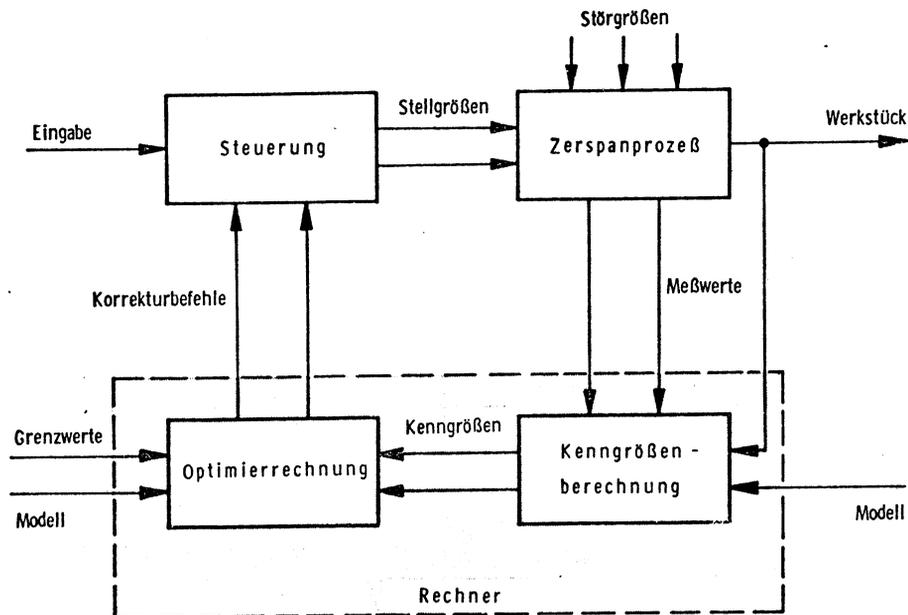


Bild 7: Blockschaltbild einer Optimierregelung

Grundlage der Optimierung bilden aktuelle Prozeßdaten, die im Rechner zunächst in aussagefähige Kenngrößen zur Bestimmung des aktuellen Gütegrades verarbeitet werden. Die anschließende Optimierrechnung führt zu einer Korrektur der Prozeßparameter hinsichtlich des optimalen Arbeitspunktes des Systems. Die Durchführung der umfangreichen Optimierrechnungen kann entweder der Fertigungsleitreehner eines DNC-Systems oder ein Optimierreehner übernehmen.

Von den Instituten der "Hochschulforschergruppe Adaptive Control (HFGAC)", die sich mit dem Einsatz von DV-Anlagen zur Prozeßsteuerung und Optimierung intensiv befassen und schon wesentliche Grundlagenarbeit auf diesem Gebiet geleistet haben, ist die Durchführung eines gemeinsamen Forschungsprojektes auf dem Gebiet der Prozeßoptimierung geplant.

Der Gegenstand dieses bereits in allen Einzelheiten umrissenen Forschungsvorhabens, das dem Projektstab des 2. DV-Förderungsprogrammes als Arbeitsunterlage vorliegt, ist die systematische Untersuchung der Voraussetzungen und Möglichkeiten für den Einsatz der Systeme zur Prozeßlenkung und die Verwirklichung derartiger Systeme im Bereich der Fertigungstechnik. Um eine praxisbezogene Forschungstätigkeit zu gewährleisten, wurde das Forschungsprojekt in Zusammenarbeit mit dem Verein Deutscher Werkzeugmaschinenfabriken e. V. (VDW) und zwei namhaften Steuerungs- und Rechnerherstellern, den Firmen Siemens AG und AEG-Telefunken festgelegt.

Im Rahmen der Untersuchungen sind die im nächsten Bild 8 dargestellten Teilaufgaben für die Verfahren

Drehen/Bohren

Fräsen/Bohren

Schleifen

einer aussagefähigen, praxisnahen Lösung zuzuführen.

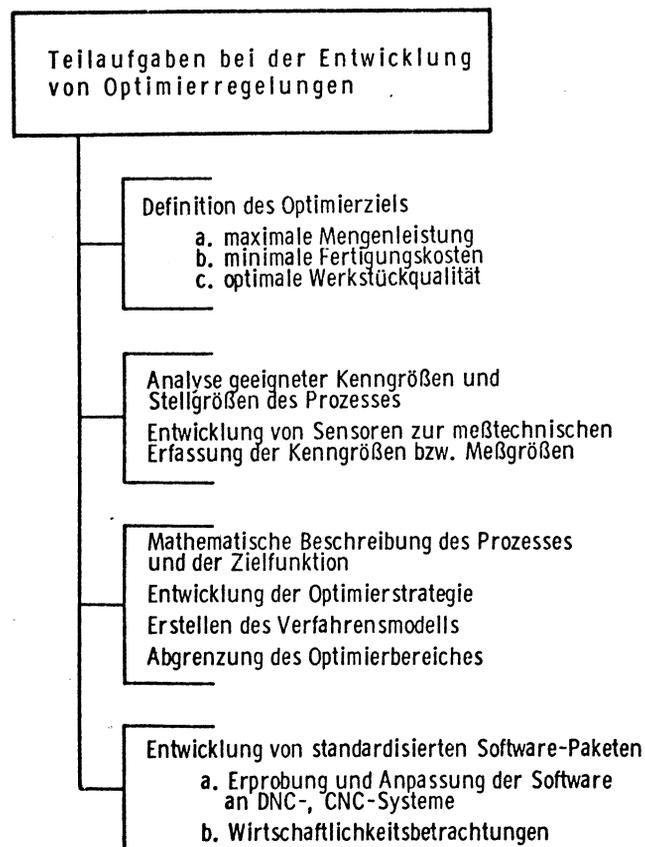


Bild 8: Teilaufgaben bei der Entwicklung von Optimierregelungen

Das Optimierziel beeinflusst maßgeblich die Grundfunktionen, Identifikationen, Entscheidungsprozeß und Modifikation eines AC-Systems. Neben der Analyse geeigneter Kenngrößen bzw. Meßgrößen ist die Entwicklung von Sensoren als wichtiges Aufgabengebiet anzusehen. Auf dem Gebiete der Prozeßidentifikation liegen bereits aussagefähige Teilergebnisse vor, die jedoch auf das gesamte AC-System sinnvoll abgestimmt und zum Teil erweitert werden müssen. Den Schwerpunkt der Entwicklungen von Optimierregelungen bildet der eigentliche Entscheidungsprozeß. Dieser Teilbereich erstreckt sich zunächst auf die mathematische Beschreibung des Prozesses und der Zielfunktion, die die Grundlage für das zu erstellende Verfahrensmodell, die Rechneralgorithmen, bilden. Die Optimierstrategie und die Abgrenzung des Optimierbereiches stellen dabei wesentliche Komponenten dar. Das Ergebnis des Entscheidungsprozesses gibt die erforderliche Modifikation der Einstellparameter des Zerspanprozesses an, um die Bearbeitung entsprechend der vorgegebenen Zielsetzung durchzuführen.

Die Entwicklung von standardisierten Software-Paketen stellt im Rahmen des geplanten Forschungsprojektes eine wichtige Teilaufgabe dar.

Auf Grund der besonderen Erfahrungen, die an verschiedenen Forschungsstellen auf dem Gebiete der Prozeßoptimierung innerhalb der Fertigungstechnik gewonnen wurden, ist folgende Aufgabenteilung im Rahmen des geplanten Forschungsvorhabens bei der Entwicklung von Optimierregelungen vorgesehen. (Bild 9).

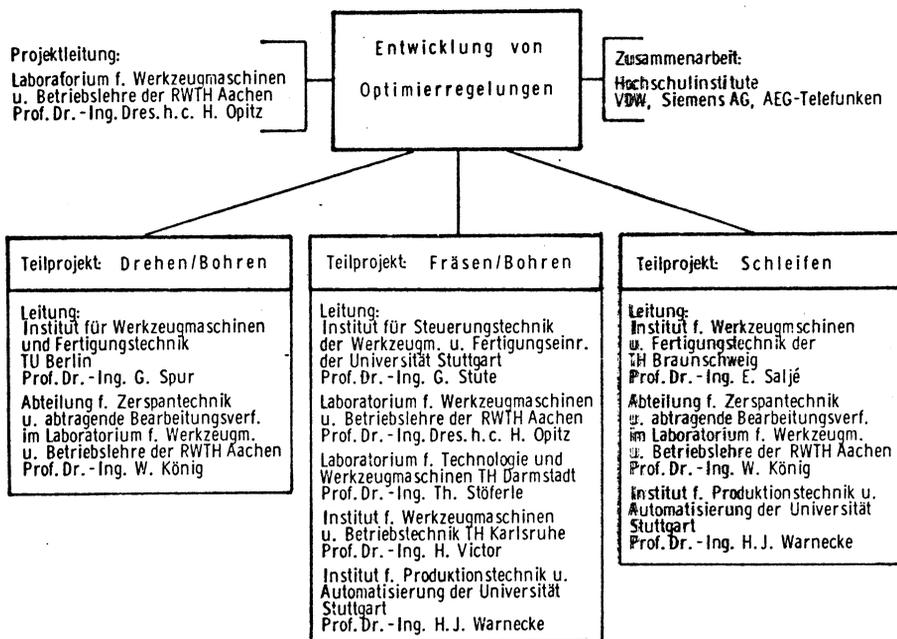


Bild 9: Vorgesehene Aufgabenunterteilung bei der Entwicklung von Optimierregelungen im Bereich der Fertigungstechnik

Die Gesamtleitung wurde von den beteiligten Instituten dem Laboratorium für Werkzeugmaschinen und Betriebslehre der RWTH Aachen, Direktor Prof. Dr. -Ing. Dres. h.c. H. Opitz, übertragen. Die Koordinierung der Arbeiten innerhalb eines Teilprojektes wird jeweils von dem angegebenen Institut vorgenommen. Die Untersuchungen und Entwicklungen werden in engem Kontakt nach Absprache mit dem Verein Deutscher Werkzeugmaschinenfabriken e. V. (VDW) und den Firmen Siemens AG und AEG durchgeführt, sodaß eine praxisbezogene Entwicklungsarbeit auf dem Gebiet der Optimierregelungen im Bereich der Fertigungstechnik gewährleistet ist.

Möglichkeiten und Grenzen des Prozeß- rechnereinsatzes in der automatischen Fertigung

Die Lage auf dem Arbeits- und Absatzmarkt zwingt die Unternehmen zur ständigen Rationalisierung der Fertigung. Im Bereich der Massenfertigung sind die Rationalisierungsmöglichkeiten im Fertigungsbereich durch die Konstruktion und den Einsatz von Einzweckautomaten weitgehend ausgeschöpft. Diese Maßnahmen in diesem Bereich werden sich auf die der Fertigung vorgelagerten Bereiche sowie auf die Entwicklung neuer Fertigungsverfahren sowie den Einsatz neuer Werkstoffe konzentrieren.

Im Bereich der Einzel- und Kleinserienfertigung läßt sich der Fertigungsbereich selbst durch den Einsatz flexibler Automaten (NC-Maschinen) weiter automatisieren. Insbesondere die Verkettung derartiger Fertigungsanlagen wird zu einer erheblichen Produktivitätssteigerung führen. Maschinen- und entwicklungsmäßig sind die Voraussetzungen zur Verkettung der Fertigungseinrichtungen durch Prozeßrechner und den Werkstückfluß automatisierende Förderanlagen vorhanden. Allerdings erfordern diese Maschinen so hohe Investitionen, daß eine vollständige Auslastung der Fertigungseinrichtungen gewährleistet sein muß. Dabei bereitet insbesondere die Organisation der Fertigung und die damit verbundenen Entwicklungen der entsprechenden Software erhebliche Probleme.

Gesamtmarktpolitisch läßt sich klar erkennen, daß diese Problematik nur in Gemeinschaftsarbeit aller Interessensgruppen bearbeitet werden kann. In anderen Ländern - USA und Japan - werden DNC-Systeme aufgrund staatlicher Aufträge und Subventionen entwickelt, die einen ständig wachsenden Entwicklungsvorsprung dieser Länder zur Folge haben.

Die Folgen lassen sich aus den derzeitigen Verteilungen der Haupt- und Nebenzeiten von 38 % Hauptzeit zu 62 % Nebenzeit bei der konventionellen Fertigung ableiten. Die Nebenzeiten werden auf ca. 20 % zu reduzieren sein und damit beim Einsatz verketteter Fertigungseinrichtungen Hauptzeitanteile von ca. 80 % bewirken. Die DNC-Fertigungssysteme höherer Ordnung erfordern eine entsprechende umfangreiche Planung bezüglich des Datenflusses innerhalb und außerhalb des Systems. Der innere Datenfluß kann durch Prozeßrechner gesteuert werden. Hierbei werden im ersten Ausbau des Systems eine Anzahl von Werkzeugmaschinen über einen Prozeßrechner mit Steuerinformationen versorgt. Ein vollautomatisch geregelter Materialfluß ist jedoch erst in einem weiteren Schritt zu erreichen, wenn die zeitliche und funktionsmäßige Koordinierung der Einzelprozesse (Bearbeitung, Transport, etc.) automatisiert ist und die entsprechende Planungs-Software vorliegt. Für diesen erweiterten Aufgabenbereich kann der Prozeßrechner nur dann sinnvoll eingesetzt werden, wenn Algorithmen für die Verarbeitung der aus dem Produktionsprozeß anfallenden Daten in allgemeingültiger Form definiert sind.

Diskussionen mit führenden Herstellern von Prozeßrechnern sowie deren Anwendern führten zu dem Vorschlag, in einer Fallstudie den Gesamtbereich der Automatisierungsmöglichkeit des Fertigungsprozesses durch den Einsatz von Prozeßrechnern zu untersuchen und damit einen Weg zu einer gemeinsamen Erstellung der kostspieligen Software aufzuzeigen. Schwerpunkte der Studie, die von neutraler Stelle in enger Zusammenarbeit mit Industrie und Hochschulen durchgeführt werden sollte, sind die Untersuchungen und Abgrenzungen der Einsatzbereiche von Prozeßrechnern in der Fertigung sowie die Untersuchung bestehender Systeme.

Folgende Ziele sollten mit der Studie erreicht werden:

1. Nachweis, daß eine Optimierung des Fertigungsprozesses durch den Einsatz von Prozeßrechnern zur globalen Steuerung verketteter Fertigungseinrichtungen möglich und wirtschaftlich vertretbar ist,
2. Ausarbeitung eines Projektplanes zur Realisierung der erforderlichen Software.
Dies schließt die Grobkonzeption
 - der erforderlichen Planungslogiken
 - der mathematischen Modelle
 - des Aufbaus der Datenstrukturen sowie
 - die Abgrenzung einzelner Modulprogramme ein.
3. Vermeidung kostspieliger Parallelentwicklungen und Ausnutzung bereits bestehender Erfahrungen.
4. Der Aufbau einer Arbeitsgruppe, in der gleichermaßen Industrie und Hochschule das vorgeschlagene Projekt bearbeiten und ggf. Marktänderungen anpassen.
5. Aufzeigen von Möglichkeiten, die Ergebnisse des ausgearbeiteten Projektes auf die konventionelle Fertigung zu übertragen.

Möglichkeiten der Systemsimulation zur Kostensenkung in der Fertigung

NC-Maschinen bieten aufgrund ihrer Flexibilität Voraussetzungen zur Automatisierung der Einzel- und Kleinserienfertigung. Das setzt jedoch die Verfügbarkeit eines Datenträgers voraus, der alle für einen reibungslosen Arbeitsablauf auf der Maschine erforderlichen Angaben enthält.

Die Erstellung des Datenträgers - üblicherweise als NC-Programmierung bekannt - ist das Ergebnis eines bis ins kleinste Detail geplanten *Fertigungsprozesses*. Die dabei anfallenden Tätigkeiten sind mit äußerster Sorgfalt durchzuführen, da nicht nur die Wirtschaftlichkeit der Bearbeitung auf den kapitalintensiven NC-Maschinen, sondern aufgrund der direkten Koppelung der Fertigung an die Arbeitsvorbereitung auch die Sicherheit der Fertigung entscheidend beeinflusst wird.

Ein derzeit übliches Verfahren der Kontrolle der Steuerlochstreifen ist der Test auf der Werkzeugmaschine. Dabei läuft der programmierte Bearbeitungsprozeß in der geplanten Dauer ab. Unter Berücksichtigung der Tatsache, daß das Teilespektrum in der Einzel- und Kleinserienfertigung sehr häufig wechselt und der Bearbeitungsprozeß sich häufig über Stunden ausdehnt, ist dieses Verfahren wegen des Anstiegs der unproduktiven Zeiten auf den teuren Fertigungseinrichtungen wirtschaftlich nicht mehr zu vertreten. Eine wirtschaftlichere Lösung des Problems ist in der Entwicklung eines EDV-Systems zu sehen, das, ausgehend von den Daten des Steuerlochstreifens, den Ablauf auf der NC-Maschine simuliert. Das System muß in der Lage sein, Fehler des Datenträgers festzustellen, die dann durch entsprechende Maßnahmen korrigiert werden können.

Diese Simulation des Arbeitsablaufes soll zunächst auf Einzelmaschinen beschränkt bleiben. Die Sicherheit des Arbeitsablaufes auf Einzelmaschinen ist jedoch bereits heute als wesentliche Voraussetzung auch zum wirtschaftlichen Einsatz verketteter Fertigungssysteme anzusehen. Das Simulationsmodell für Einzelmaschinen bildet

gleichzeitig die Grundlage des stufenweisen Ausbaus zum Simulationsmodell für Fertigungssysteme.

Der Aufbau von Simulationsmodellen für folgende Fertigungseinrichtungen ist vorgesehen:

- Bohrmaschinen
- Bohr- und Fräswerke
- Drehmaschinen (1 Support)
- Drehmaschinen (mehrere Supporte)
- Fräsmaschinen
- Bearbeitungszentren

Teile der Einzelmodelle werden in jedem Gesamtmodell wieder vorkommen, so daß bereits im Planungsstadium auf einen modularen Systemaufbau Rücksicht genommen wird. Die zu erarbeitenden Modelle sind Basissoftware der Prozeßrechner zur Steuerung verketteter Fertigungseinrichtungen.

Betreff Bedeutung und Anwendung der Simulation

1. Bedeutung und Anwendung der Simulation in der BBC-Praxis

In diesem Kurzbeitrag soll die besondere Bedeutung der Simulation für die Systemanalyse, Synthese, Prüfung und Inbetriebnahme von prozeß-rechnergeführten Anlagen stärker hervorgehoben werden, als das aus der Themenübersicht für die PDV-Tagung am 4./5.11.1971 hervorgeht. Es werden hierbei nur die wichtigsten Gesichtspunkte aufgeführt.

- a) Bei der Analyse vermittelt die Simulation a-priori-Kenntnisse über das Systemverhalten. Die hierdurch gewonnenen Ergebnisse sollten möglichst durch Messungen an vergleichbaren realen Anlagen ergänzt werden, um die Güte des Simulationsmodells festzustellen. Die Prozeßanalyse muß sehr sorgfältig durchgeführt werden, da die nachfolgende Synthese von diesen Ergebnissen ausgeht und bei guter Systemkenntnis oft vereinfacht werden kann.

- b) Im Rahmen der Synthese verschafft die Simulation einen Einblick in das Verhalten des Gesamtsystems (Strecke mit Regeleinrichtung). In diesem Stadium kann das theoretische Gesamtkonzept unter Einwirkung der später zu erwartenden realen Störsignale erprobt werden. Voraussetzung hierfür ist die vorangegangene sorgfältige Analyse der Führungs- und Störungssignale.

- c) Nach Erstellung der Hardware und Software werden diese Teile sukzessive in den Simulationsablauf einbezogen. Das hat folgende Vorteile:
 - Test der Hard- und Software,
 - genauerer Test des Gesamtkonzeptes, da die Einzelkomponenten bei der Simulation oft nur näherungsweise dargestellt werden;
 - Inbetriebnahme-Vorbereitung durch Ermittlung der Parameterempfindlichkeit, Schulung der Inbetriebnehmer usw.

Betreff Bedeutung und Anwendung der Simulation

- d) Hilfsmittel hierzu sind Analog-, Digital- und Hybridrechner. Die in diesem Förderungsrahmen interessierenden digitalen Systeme sind der Prozeßrechner und das hybride Rechnersystem.

Die digitale Simulation wird hauptsächlich für umfangreiche und besonders nichtlineare Prozesse sowie bei der Forderung nach guter Dokumentationsmöglichkeit eingesetzt. Im Interesse der Übersichtlichkeit und Handhabbarkeit eines zu erstellenden Modellkataloges sollte die Vielzahl der angebotenen benutzerfreundlichen "Blockorientierten Programmiersprachen" zur Simulation (z.B. CSMP, MIDAS), drastisch reduziert werden. Besonders ist darauf zu achten, daß diese Programmiersysteme auch auf Prozeßrechnern anwendbar sind. Deshalb wurde von uns das digitale Simulationssystem DIGSI erstellt, das auf einem 32 K - Prozeßrechner laufen kann.

Hybride Rechnersysteme werden eine wachsende Bedeutung für die Prozeßsimulation erlangen:

Auf dem Analogrechner ist der analoge Prozeß nachgebildet, dessen Steuerung und Regelung - wie später im echten Betrieb - der Prozeßrechner übernimmt. Auf diese Weise läßt sich die gesamte Software einschließlich der Anwenderprogramme zum Ansprechen der peripheren Geräte und des back-up sowie diese Hardwareteile selbst prüfen.

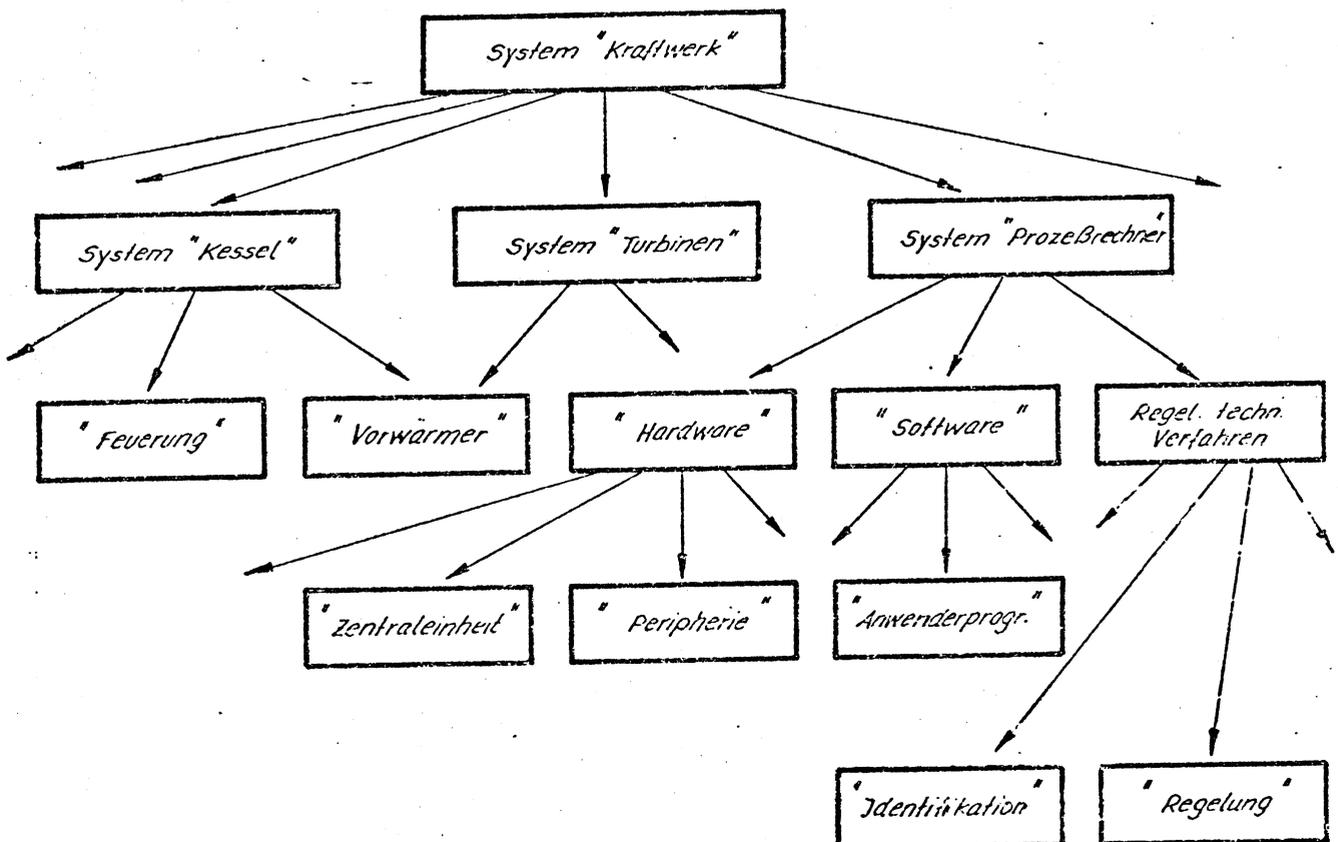
Voraussetzung, daß die Simulation ein wirksames Hilfsmittel mit kurzer Zugriffszeit darstellt, ist jedoch, daß die im Rahmen der "Systemanalyse" erarbeiteten Modelle katalogisiert mit Leistungskriterien und Rechnerprogramm vorliegen.

Betreff Bedeutung und Anwendung der Simulation

Bemerkungen zu "Systemanalyse"

Nach unserer derzeitigen Erfahrung sollten als Ergebnis der Systemanalyse folgende Ziele angestrebt werden:

- a) Ein Modellkatalog mit Leistungskriterien unter besonderer Beachtung der nichtlinearen Elemente. In diesem Katalog müssen die Modelle "technischer Bausteine" enthalten sein. Diese Bausteine sind Untersysteme (z.B. Rohrleitung, Wärmeaustauscher, Destillationskolonne oder auch einzelne Böden einer solchen Kolonne, schwingungsfähige mechanische Systeme). Jedes größere System läßt sich - entsprechend der folgenden Abbildung - in solche Untersysteme aufspalten und, nachdem solch ein Katalog vorliegt, zu jedem Zeitpunkt bausteinartig wieder daraus zusammensetzen.



Betreff Bedeutung und Anwendung der Simulation

b) Verfahrenskatalog mit Programmbibliothek

Sowohl zum Aufbau der Modelle als auch zur off-line und on-line Identifikation und zur späteren Synthese sind zugriffsbereite handhabbare Verfahren erforderlich. Diese müssen in ähnlicher Weise vorliegen, wie das schon jahrelang in Rechenzentren der Fall ist: als Softwareprogramm-Bibliothek mit einer kurzen Programmbeschreibung. Dieser letzte Schritt von der theoretischen Behandlung bis zur anwendungsfreundlichen Aufbereitung wurde bisher nur in Ausnahmefällen gemacht und ist ein wesentlicher Grund dafür, daß nur ein minimaler Anteil der theoretisch erarbeiteten Verfahren praktisch genutzt wird. In dem Verfahrenskatalog sollten alle Verfahren einschließlich ihres Rechnerprogramms und zugehöriger Beschreibung enthalten sein; besonderer Wert ist dabei auf beschreibende Leistungsangaben (z.B. Voraussetzungen, Gültigkeit des Verfahrens, Näherungen usw.) zu legen, die einen Leistungsvergleich ermöglichen und damit dem Benutzer die erforderlichen Anwendungshinweise geben.

Erst nach Schaffung dieser Voraussetzungen ist daran zu denken, auf dieser Stufe einer industriellen Gesamtabwicklung auch den kritischen Punkt der Wirtschaftlichkeit befriedigend zu lösen. Die Arbeitsweise würde dann etwa folgendermaßen aussehen:

- Begonnen wird mit Hilfe der Modelle aus dem Katalog, die auf dem Rechner zu dem gewünschten System kombiniert werden. Die damit mögliche Simulation liefert a-priori-Kenntnisse über das Systemverhalten (dieses Modell-Systems).
- Mit Hilfe der off-line und on-line Identifikationsverfahren werden die Simulationsergebnisse mit denen vergleichbarer realer Anlagen verglichen; durch Modellverbesserung wird die erforderliche Übereinstimmung hergestellt und somit das Modell "sicherer gemacht."
- Nach Absicherung des Gültigkeitsbereiches dieses Modells lassen sich auf dem Rechner alle möglichen Betriebszustände - auch die Gefahrensituationen - risikolos erproben.
- Hiernach kann mit der Synthese begonnen werden.

Betreff Bedeutung und Anwendung der Simulation

Bemerkungen zu "Rechnersprachen"

Grundvoraussetzung für die Erstellung eines übersichtlichen und anwendbaren Modell-Kataloges ist die Verwendung weniger - möglichst nur einer - "Blockorientierter Programmiersprachen". Hierbei ist jedoch von vornherein darauf zu achten, daß diese Sprache relativ einfach Modifikationen gestattet und mit dem Kernspeicher eines Prozeßrechners auskommt. Dasselbe gilt für den Themenkreis der "Prüfsprachen".

Betreff Bedeutung und Anwendung der Simulation

Bemerkungen zu "Synthese"

Ein sehr wichtiges Kapitel dieses Themenkreises sind die Punkte "Ausfall-Analyse" (Ausfall-Vorhersage), "Redundanz" und "Sicherheit". Diese Probleme sollten als ein Themenkreis betrachtet werden, da sowohl für die Rechner als auch für die zu regelnden und zu steuernden Systeme noch weitgehend die Grundlagen der Sicherheitsphilosophie erarbeitet werden müssen. Besonders wichtige Anwendungen sind z.B. Kraftwerke und Verkehrssysteme.

Betreff Bedeutung und Anwendung der Simulation

Vorschläge für die Intensivierung des Kontaktes "Hochschule-Industrie"

Um sicherzustellen, daß im Rahmen dieses Förderungsprogrammes handhabbare und damit für die Industrie überhaupt interessante Lösungen erarbeitet werden, sollte diese Entwicklung gemeinschaftlich von Forschungsinstituten und der Industrie durchgeführt werden. "Gemeinschaftlich" bedeutet, daß nicht einzelne eng umrissene Aufgaben von der Industrie an Forschungsinstitute gegeben werden, die mit einem Berichtswerk abschließen, sondern daß die gesamte Entwicklung in Form eines Teams - welches nicht unbedingt räumlich zusammensitzen muß - bearbeitet wird. Von der Institutseite kämen in diesem Fall die theoretischen Grundlagen, die Industrie setzt die Anforderungen (Pflichtenblatt) und sorgt in jedem Stadium der Entwicklung für die Überprüfung an konkreten Einsatzfällen und für eine anwenderfreundliche Aufbereitung. Wir haben auf diesem Gebiet schon erste Erkenntnisse gesammelt und sind gern bereit, über die Form dieser Zusammenarbeit und die Gestalt der Dokumentation unsere Erfahrungen und Gedanken beizutragen.

Ein Haupthindernis ist der mangelnde Informationsfluß zwischen den Instituten und der Industrie; wir schlagen deshalb vor:

In bestimmten Abständen trifft sich ein nicht zu großer Kreis aus industriellen Entwicklungsleitern und Institutsleitern zu zwei- bis dreitägigen Aussprachetagen. In diesem Rahmen sollten die Institutsleiter in Kurzreferaten die an ihren Instituten durchgeführten Arbeiten praxisverständlich erläutern und die Industrie-Vertreter ihre Probleme schildern. Durch diese Referate und die dabei möglichen Diskussionen wissen alle interessierten Stellen, welche theoretischen Lösungen wo erarbeitet werden und wo welche Probleme besonders akut sind.

