

Informatikgeschichte im Informatikunterricht – Konzepte und Materialien

Alexander Best

Westfälische Wilhelms-Universität Münster
Fachbereich Mathematik und Informatik (FB 10)
Institut für Didaktik der Mathematik und der Informatik
Arbeitsbereich Didaktik der Informatik
alexander.best@uni-muenster.de

Abstract: In diesem Beitrag werden Konzepte der Geschichtsdidaktik auf die Informatiklehre unter Berücksichtigung bereits etablierter Methoden und Lehr-Lernformen transferiert. Es werden Konzepte und Materialien behandelt, um Themen der Informatikgeschichte für Lernende transparent und strukturiert für den Unterricht aufarbeiten zu können. Dazu werden exemplarisch historische Materialien als Gegenstände für den Informatikunterricht aufgegriffen und ihre didaktischen Potentiale herausgestellt.

1 Hintergründe

Die Informatik als wissenschaftliche Disziplin an deutschen Hochschulen weist eine Bestandszeit von rund 40 Jahren auf. Wie kaum eine andere Wissenschaft hatte und hat sie einen immensen Einfluss auf technologische, soziale und politische Veränderungen unserer Gesellschaft. Um die Genese heutiger Konzepte, wie z.B. soziale Netzwerke oder mobile Anwendungen erkennen zu können, bildet Geschichtsbewusstsein [Si01] einen Schlüsselfaktor. Aktuelle Thematiken und Erscheinungsformen innerhalb der Informatik stellen durch die Einbettung in den historischen Kontext für Lernende einen Zugang zur Vergangenheit und Zukunft der Disziplin und ihrer Inhalte bereit. Untersuchungen wie etwa [UO10] zeigen, dass SchülerInnen neuen Technologien mehrheitlich unkritisch oder gleichgültig begegnen. Sie werden als feste Bestandteile der Gesellschaft aufgefasst. Über Herkunft, Auswirkungen oder zukünftige Einsatzmöglichkeiten solcher Technologien findet – zumindest uninitiiert – keine Reflektion statt. Diese und andere Überlegungen haben zur Folge, dass sich die Didaktik der Informatik in den letzten Jahren vermehrt darum bemüht neue Konzepte zur Integration von informatikgeschichtlichen Inhalten in die Lehre auszuarbeiten, sowohl an Schulen wie auch an Hochschulen. Während über die Potentiale (und auch Nachteile) dieser Bemühungen bereits Erkenntnisse dargestellt wurden, fehlt es noch an konkreten Konzepten sowie an exemplarischen Lehr- und Lernmaterialien. Einige sollen im vorliegenden Beitrag verdeutlicht und aufbereitet werden.

2 Konzepte

Immanent bei der Integration von geschichtsspezifischen Inhalten in den Schulunterricht sind Konzepte, um diese in bestehende Lehrabläufe einfließen lassen zu können. Die bislang im Informatikunterricht eingesetzten Konzepte reichen hierfür nicht aus, da sie nicht für die Vermittlung von historischen Inhalten ausgelegt sind. Die Geschichtsdidaktik bietet sich aufgrund langjähriger Forschungsarbeiten und Evaluationen auf diesem Gebiet zur Untersuchung eines möglichen Transfers von Methoden auf die Informatiklehre an. Nachfolgend werden einige ausgewählte thematische Strukturierungskonzepte vorgestellt und für die Informatiklehre diskutiert. In Abschnitt 2.2 werden ausgewählte Lehr-Lernformen betrachtet und deren Vorteile für den Informatikunterricht.

2.1 Thematische Strukturierung

Das genetisch-chronologische Verfahren: Neue Erkenntnisse, Ansätze und Strategien werden stets von den vorherigen beeinflusst und haben zugleich Auswirkungen auf zukünftige Entwicklung. Dieser Umstand kann didaktisch aufgegriffen werden, um für die Lernenden zuvor zusammenhanglose Ereignisse temporal und kausal in Beziehung zu setzen. Besonders die Interdependenz zwischen Notwendigkeit, Umsetzung und Auswirkung konkreter Entwicklungen kann im Zuge dieses Verfahrens ansprechend verdeutlicht werden. Als Beispiel könnten nach Eulenhöfer [Eu98] der Plankalkül der 1940er Jahre, die Befehlspläne der 1950er und 1960er Jahre sowie das heutige Verständnis von Programmiersprachen dienen. Auch die Beziehungen zwischen Leibniz Aufforderung *Calculemus!*¹ und späteren Bemühungen auf dem Gebiet der Formalisierung und Automatisierung, wie sie Humbert [Hu09] vorschlägt, könnten in diesem Unterrichtsverfahren behandelt werden. Die chronologische Vermittlung dieser Themen verdeutlicht ihre Genese. Ohne diesen Umstand können die zuvor genannten Begriffe zwar thematisch erfasst werden, bleiben jedoch für die Lernenden jeweils autarke Einzelentwicklungen. Auch der Blick in die zukünftige Entwicklung des heutigen Verständnisses von Programmiersprachen bliebe somit verschlossen. Insbesondere die geschichtlichen Hintergründe der sogenannten „Softwarekrise“ eignen sich dafür, Aussagen über die Anforderungen an zukünftige Programmiersprachen treffen zu können. Die Genese des *software engineering* vom *writing* über *building* hin zum *growing* verdeutlicht, wie Konzepte hinsichtlich neuer Entwicklungen überdacht und angepasst werden müssen (s.a. [Th09]). Ebenfalls die Geschichte(n) konkreter Programmiersprachen und deren Zielsetzungen sowie Verbreitung zu behandeln, würde es den SchülerInnen ermöglichen die heutige Programmiersprachenlandschaft besser zu verstehen und Abschätzungen bezüglich Anforderungen an die Sprachen von Morgen treffen zu können. Hierzu gehört etwa die Tatsache, dass es niemals eine einzelne Programmiersprache für alle Anforderungen geben kann. Während Sprachen wie Java, C oder C++ eine hohe wirtschaftliche und auch teilweise universitäre Durchdringung erfahren haben, eignen sich für didaktische

¹ Hierin werden bereits um 1680 Ansätze zur kalkülhaften Beschreibung von Sachverhalten zwecks Abgleich unterschiedlicher Auffassungen erkennbar [Hu09].

Zwecke hingegen ältere, skriptbasierte oder grafische Programmiersprachen.² Während in der Literatur (s.o.) zumeist klassische Beispiele zu finden sind, können auch Themengebiete mit hoher Aktualität und einem starken Bezug zum Lebensumfeld der SchülerInnen in diesem Verfahren aufgegriffen werden. So erfordert die rapide Verbreitung von mobilen Endgeräten (*mobile computing*), die dezentrale Datenverarbeitung (*cloud computing*) oder die grundlegende Veränderung von Mensch-Maschine-Interaktion (Tastatur/Maus → *Touch*oberflächen) eine Neubewertung bestehender Programmier- und Sicherheitskonzepte. Für diese Entwicklungen gibt es in der Vergangenheit bereits vereinzelt Indikatoren, welche mit den SchülerInnen aufgearbeitet werden können. Zugleich lässt sich für die Zukunft prognostizieren, dass etwa die klassischen *Personal Computer* durch mobile Geräte verdrängt werden. SchülerInnen können erkennen, dass aktuelle Neuerungen Momentaufnahmen einer Entwicklung darstellen, welche in der Vergangenheit eingesetzt hat und sich in der Zukunft weiter an gesellschaftliche Ansprüche/Notwendigkeiten annähern wird. Auf weitere spezifisch unterrichtspraktische Beispiele wird in Abschnitt 3 Bezug genommen. Während die Lernenden bei diesem Verfahren auf die historische Wahrnehmung und Auswirkung von Ereignissen sensibilisiert werden, kann jedoch kein hoher Grad an Fachwissen vermittelt werden. Hierfür bieten sich die nachfolgenden Verfahren an.

Die Fallanalyse: Beim genetisch-chronologischen Verfahren ist es an ausgesuchten Stellen sinnvoll, mit Lernenden ein konkretes historisches Ereignis detaillierter zu betrachten. Hierfür bietet sich die Fallanalyse an. Sie wird ebenfalls in anderen Disziplinen, wie etwa der Wirtschaftswissenschaft, Soziologie oder Publizistik eingesetzt. Auch bei der Verbrechensbekämpfung kommen Fallanalysen zum Einsatz um Täterprofile zu erstellen. In allen Fällen wurde deutlich, dass die Verlagerung der Sichtweise vom *Absstraktum* hin zum *Konkretum* das Verständnis für die Materie vereinfachen und erhöhen kann. Insbesondere bei zeitgeschichtlichen Themen mit hohem Aktualitätsbezug und gesellschaftlicher Präsenz bietet sich dieses Verfahren an, da sich persönliche/emotionale Einstellungen der SchülerInnen zu einem solchen Thema positiv auf den Unterricht auswirken können. So würde sich beispielsweise bei der Behandlung von Datenschutzaspekten die Fallanalyse anbieten, um Erfahrungen der SchülerInnen zu sammeln und auf Erlebnisse einzugehen. Insbesondere bezüglich sozialer Netzwerke oder Passwortsicherheit können SchülerInnen i.d.R. konkrete Beispiele in den Unterricht einbringen. Ebenfalls die Ausführungen Brunnsteins [Br92] zu Computer-Unfällen eignen sich gut, um Aspekte der Softwareverifikation und -validierung in Form der Fallanalyse zu behandeln. Dabei kann sehr tiefgründig vorgegangen werden, da keine anderen Ereignisse thematisch/chronologisch zum Behandelten in Beziehung gesetzt werden. Ziel ist die Vermittlung eines hohen Grades an fachlichem Hintergrundwissen an die Lernenden. Die Fallanalyse kann weiterhin für den Themeneinstieg genutzt werden. Die SchülerInnen haben somit im Verlauf der Unterrichtsreihe einen fachlichen Bezugspunkt, zu dem neue und nicht derart detailliert betrachtete Inhalte in Beziehung gesetzt werden können.

² Interessant in diesem Zusammenhang sind Statistiken, wie etwa der *TIOBE Programming Community Index*, welche aktuelle und historische Daten bezüglich der Verbreitung von Programmiersprachen veröffentlichen. Siehe http://www.tiobe.com/index.php/tiobe_index

Die Konstellationsanalyse: Besonders die Geschichte der Informatik zeigt, dass keinesfalls von einem einzelnen Entwicklungsstrang innerhalb der Wissenschaft ausgegangen werden kann. So lassen sich nach Thomas [Th05] teilweise vollkommen unabhängig existierende Entwicklungen nachweisen, deren Zielsetzungen nur marginal voneinander abweichen. Ein prägnantes Beispiel für diesen Umstand seien die epochenübergreifenden und weltweiten Bemühungen auf dem Gebiet der Automatisierung. Ein einzelner stringenter Entwicklungsstrang vom Abakus bis in die Ära der Mikroprozessoren existiere nicht. Die Konstellationsanalyse – ursprünglich aus dem Ressourcenmanagement stammend [Gü08] – ermöglicht es dem Lehrenden unterschiedliche Entwicklungsstränge innerhalb der Informatik für die Lernenden zueinander in Beziehung zu setzen. Unterschiede, Gemeinsamkeiten, Ursprünge und gegenseitige Abhängigkeiten können aufgezeigt und – meist in Form einer Grafik – festgehalten werden. Diese könnte für das Beispiel „Programmierparadigmen“ – wie es Schubert/Schwill in [SS04] vorschlagen – die verschiedenen Lösungsstrategien für ein konkretes Problem zueinander in Beziehung setzen. Während bei imperativen Programmiersprachen die Frage nach dem „Wie?“ im Vordergrund steht, ist bei deklarativen Sprachen die Frage nach dem „Was?“ entscheidend. Zusätzlich könnte eine solche Grafik die bevorzugten Einsatzgebiete für deklarative und imperative Sprachen beinhalten. Dies verdeutlicht den SchülerInnen, dass die Anforderungen an Sprachen und deren Strategien vom Einsatzgebiet abhängig sind. Auch die Entwicklungsgeschichten konkreter Sprachen können hierzu Aufschlüsse geben. So zeigt etwa die Geschichte der Programmiersprachen, dass neue Entwicklungen auf älteren Konzepten aufbauten, jedoch konkrete Schwächen oder Inkonsistenzen verbessern sollten. Ebenfalls die Tatsache, dass imperative Sprachen und deklarative Sprachen häufig wechselseitig Konzepte voneinander übernahmen und in ihr jeweiliges Paradigma eingliederten, ist zu beobachten. Durch eine angemessene Aufarbeitung in grafischer Form können viele dieser historischen Entwicklungen und Abhängigkeiten der Programmierparadigmen für die SchülerInnen ansprechend visualisiert werden. Ein weiteres denkbares Themengebiet im Zuge der Konstellationsanalyse wäre die Gegenüberstellung von klassischen Enzyklopädien und dem Wikipedia Projekt. Es könnten die historische Ausgangssituation (im Falle der klassischen Enzyklopädien z.T. jahrhundertalt), die beiderseitige Entwicklung inklusive der Gemeinsamkeiten, Unterschiede und Wechselwirkungen, die Zielsetzungen sowie die Bedeutung beider Formen für die Gesellschaft heute betrachtet werden. Eventuell könnten daraus folgendermaßen Absätzungen hinsichtlich zukünftiger Entwicklungen herausgestellt werden.

*Das individualisierende Verfahren*³: Einzig der Mensch in seiner Rolle als Erschaffer von Geschichte ist Gegenstand dieses thematischen Strukturierungskonzeptes. Veränderungen innerhalb der Informatik oder das Aufkommen neuer Impulse sollen dem Lernenden anhand mehrerer Persönlichkeiten vermittelt werden. Wesentlich ist – genau wie bei den vorherigen Verfahren – die multidimensionale Sichtweise auf konkrete Ereignisse zu ermöglichen und zu schärfen. Die Lernenden sollen eine Beziehung zu den behandelten Personen aufbauen und Ereignisse der Informatikgeschichte somit personalisiert

³ Es lassen sich Parallelen zum Konzept des durch Thomas [Th05a] beschriebenen *history lifts* aus der Chemiedidaktik erkennen. Allerdings wird dieses Verfahren erst dann eingesetzt, „wenn der heuristische Weg der Erkenntnisgewinnung und der Problemlösung für den Schüler zu schwierig wird“. Das genetisch-chronologische Verfahren der Geschichtsdidaktik hingegen ist für den durchgängigen Unterrichtseinsatz ausgelegt.

nachvollziehen können. Sogenannte „versteckte Ereignisketten“ [Gü08], wie beispielsweise „... es kam zur Softwarekrise“ oder „... leitete die Ära der Mikroprozessoren ein“ sollen vermieden werden, da dies zur Anonymisierung von Geschichte führt. Das individualisierende Unterrichtsverfahren eignet sich zur Auseinandersetzung mit den Lebensgeschichten bedeutender Persönlichkeiten aus der Informatik und deren Errungenschaften. So bietet es sich bei der Behandlung von frühen Rechenmaschinen an, den Zugang über die Lebensgeschichte Konrad Zuses oder John von Neumanns zu wählen. Hierbei können den SchülerInnen die Hintergründe und Intentionen, welche zur Entstehung etwa der Z1 oder dem *Manchester Mark I* führten, aufgezeigt werden. Die Frage nach dem „Warum?“ kann von den SchülerInnen über die Lebensgeschichten zuvor genannter Persönlichkeiten selbst erschlossen werden. Handelt es sich bei den Lernenden um jüngere SchülerInnen, so hat sich in der Geschichtsdidaktik auch der Zugang zu Themen über fiktive Charaktere bewährt. Auch in diesem Verfahren bietet es sich an auf Personen einzugehen, welche eine hohe gesellschaftliche Präsenz aufweisen. So können beispielsweise über die Person Bill Gates oder Steve Jobs die Entwicklung von Betriebssystemen und die Veränderungen innerhalb der Softwarebranche der letzten 30 Jahre exemplarisch an die SchülerInnen weitervermittelt werden. Zudem könnten Personen wie Ken Thompson, Dennis Ritchie oder Linus Torvalds – den SchülerInnen mehrheitlich nicht bekannt – vorgestellt werden und Konzepte des Unix bzw. Linux Betriebssystems zu den zuvor genannten Betriebssystemen in Beziehung gesetzt werden, um Unterschiede und wechselseitige Einflüsse herauszustellen. Hier würde sich also eine Symbiose zwischen dem individualisierendem Verfahren und der Konstellationsanalyse anbieten.

Zur Umsetzung der in Abschnitt 2.1 aufgeführten thematischen Strukturierungskonzepte werden angemessene Lehr-Lernformen benötigt. Auch hier können aus der Geschichtsdidaktik Konzepte entlehnt werden, um die Lernmethoden des Informatikunterrichts zu erweitern.

2.2 Lehr-Lernformen

Die Quellenarbeit: Die Arbeit mit historischen Quellen ist die am meisten eingesetzte Lehr-Lernform innerhalb des Geschichtsunterrichts. Sie zeichnet sich dadurch aus, dass durch die Auswahl der Quellen zwar eine didaktische Reduktion erfolgt, jedoch i.d.R. keine didaktische Rekonstruktion.⁴ SchülerInnen sollen Kernaussagen und Intentionen von Quellen selbstständig erarbeiten. Zusatzinformationen können dabei weitergereicht werden, um *Termini technici* zu erläutern oder aus der Quelle selbst nicht ersichtliches Hintergrundwissen zu liefern. Ein Beispiel hierfür in der Informatik wären chiffrierte schriftliche Überreste der ENIGMA oder Briefe bzw. Aufzeichnungen bedeutender Persönlichkeiten wie etwa Edsger W. Dijkstra, Alan Turing, John von Neumann oder Kurt Gödel. Es scheint für die Informatik analog zur Geschichtswissenschaft sinnvoll, eine Unterscheidung zwischen zwei Grundtypen von Quellen vorzunehmen: Die der Traditi-

⁴ Ausnahmen können auftreten, falls es sich um eine sehr umfangreiche Quelle handelt. Wird die Quelle in logische Abschnitte aufgebrochen, kann es in der Unterrichtsplanung sinnvoll sein die Abfolge dieser Abschnitte abzuändern. Auch bei der Arbeit mit sehr umfangreichen bildlichen oder filmischen Quellen ist eine didaktische Rekonstruktion denkbar.

on und die des Überrestes. Wurde vom Autor einer Quelle die Intention verfolgt, der Nachwelt eine bestimmte Sichtweise zu vermitteln, spricht man von einer Tradition. Bei der Auseinandersetzung mit dieser Art von Quellen ist eine umfassende Quellenkritik notwendig, da die historische Faktizität durch persönliche Sichtweisen beeinträchtigt wird. Handelt es sich bei Quellen hingegen um zufällige Überlieferungen, denen ein rein dokumentierender oder praktischer Zweck zukam, so wird von einer Tradition gesprochen. Es wird keinerlei persönliche Sichtweise übermittelt, sondern die faktische historische Vergangenheit. Jedoch ist das Arbeiten mit Traditionen weitaus anspruchsvoller, da vom Autor keine Aufbereitung erfolgt ist und dies somit ausschließlich von der Lehrperson – ausgenommen es handelt sich um einen bereits ausgearbeiteten Unterrichtsentwurf – erfolgen muss. Exzerpte aus Konrad Zuses Publikation „Der Computer – Mein Lebenswerk“ würden anhand zuvor genannter Kriterien als Tradition behandelt werden müssen. Ein Dokument, wie die Patentanmeldung Z23624 der Prozessorarithmetik der Zuse Z1 vom 21.12.1936 würde hingegen in die Kategorie der Überreste fallen.⁵ In der Informatiklehre bietet es sich auch an, neben der Quellenarbeit im klassischen Sinn, neuere Formen und Einsatzgebiete im Unterricht zu erproben. Mit der Versionierung innerhalb der Softwareentwicklung stehen beispielsweise detaillierte Protokolle von der Entstehung bis zur Gegenwart zur Verfügung. Die Versionierungsprotokolle stellen somit ein historisches Dokument dar. Ebenfalls historischer Quelltext fällt unter diese Kategorie. Bei der Betrachtung von Quelltext vor den 1990er Jahren fällt auf, dass i.d.R. auf Anmerkungen verzichtet wurde und der Aspekt Effizienz der Les- und Wartbarkeit übergeordnet wurde. Die Folge ist, dass Quelltext aus dieser Zeit extrem schwer semantisch erfasst oder gar verändert werden kann. Die Quellenarbeit beschränkt sich jedoch nicht ausschließlich auf schriftliche Quellen, sondern beinhaltet auch Bilder (zu denen auch Fotografien gezählt werden) sowie filmisches Material. Im Geschichtsunterricht wird hierfür i.d.R. das Panofsky'sche Interpretationsschema [Pa67] – z.T. in abgewandelter Form – angewendet: Es erfolgt zunächst eine Bildbeschreibung. In dieser Phase sollen die SchülerInnen lediglich beschreiben, was sie sehen. Ziel ist es alle wesentlichen Elemente des Bildes mit den SchülerInnen herauszuarbeiten, welche später analysiert und interpretiert werden sollen. Anschließend erfolgt die Bildanalyse. Hierbei sollen die Bedeutungen der zuvor herausgestellten Elemente erfasst werden. Auch sollen SchülerInnen eine Bedeutungshierarchie für die Bildelemente erarbeiten. Während einige Elemente für die abschließende Interpretation von grundlegender Bedeutung sind, fällt anderen Elementen hingegen eine geringere Aussagekraft zu. In der abschließenden Bildinterpretation sollen die Bedeutungen der wesentlichen Bildelemente zueinander in Beziehung gesetzt werden, um eine zentrale Kernaussage über die intentionelle Absicht des Autors treffen zu können.

Visualisierung: Bei Visualisierungen ist es möglich durch Beteiligung der SchülerInnen Unterrichtsergebnisse festzuhalten, aber auch Unterrichtsgegenstände zu entwickeln. Dies wird in der Fachdidaktik Informatik beispielsweise durch *Mindmaps* praktiziert. Bei der Behandlung von geschichtsspezifischen Inhalten im Informatikunterricht besteht jedoch die Notwendigkeit neben kausalen Bezügen auch den zeitlichen Kontext abzubilden. Eine Möglichkeit dies zu tun sind Zeitleisten. Allerdings ist deren Einsatz in einigen Situationen nicht sinnvoll, etwa wenn eine sehr umfangreiche Zeitspanne mit sehr vielen

⁵ <http://www.zib.de/zuse/Inhalt/Texte/Chrono/30er/Pdf/0177.pdf>

Ereignissen dargestellt werden soll und falls zwischen den dargestellten Ereignissen vielschichtige kausale Bezüge existieren. Um diese Nachteile auszugleichen, empfiehlt die Geschichtsdidaktik den Einsatz eines sogenannten Geschichtsfrieses. Dabei handelt es sich um eine Vereinigung (im mathematischen Verständnis) zwischen der Zeitleiste und dem Schaubild. Anders als im Falle der Zeitleiste werden nicht lediglich stichpunktartige Einzelereignisse festgehalten, sondern ganze Textpassagen, Bilder o.ä. Auch gibt es keinerlei Beschränkung der dargestellten Dimensionen und somit können zuvor dargestellte zeitliche Abschnitte erneut unter anderen Aspekten oder mit einem höheren Ereignissaufkommen dargestellt werden. Der Geschichtsfries kann somit verwendet werden, um Unterrichtsinhalte eines ausgedehnten Zeitraumes – sogar eines gesamten Schuljahres [Gü08] – festzuhalten bzw. zu entwickeln. Für einen Geschichtsfries, welcher die 1940er bis 1970er Jahre abdecken soll, könnten beispielsweise folgende Elemente integriert werden:

- Fotografien von Großrechnern im Kontrast zu Fotos der frühen Heimcomputer der 1970er Jahre
- Statistiken über die Transistorzahlen der CPUs (Stichwort *Moore's Law*), Herstellungskosten und Leistungsfähigkeit (FLOPS)
- Bedeute Ereignisse für die Informatik innerhalb dieser Zeitspanne: Softwarekrise, ARPANet/Internet, proprietäre Software contra quellenoffener Code, etc.
- Programmiersprachen als Zeugnisse für historische Denkweisen

Das Schreiben: Dem Schreiben kommt in allen gesellschaftswissenschaftlichen Fächern eine zentrale Rolle zu. Wie bei kaum einer anderen Arbeitsform können SchülerInnen beim Schreiben nicht existierendes Wissen oder Halbwissen ihrerseits erkennen und somit Defizite selbstständig aufarbeiten. Weiterhin setzt das Schreiben über einen konkreten Unterrichtsgegenstand voraus, dass SchülerInnen ihr Wissen selbstständig strukturieren und so neue Zusammenhänge erkennen können. Besonders bei Siefkes [Si01] wird deutlich, dass das Schreiben auch in der universitären Lehre den Informatikstudenten große Probleme bereitet. So ergibt sich zumeist erst bei der Bachelor- und Masterarbeit die Konfrontation mit der Erstellung eines längeren Textes. In der schulischen Lehre kann das Schreiben von Texten zudem mit der Visualisierung verbunden werden, um Ausstellungen mit den SchülerInnen durchzuführen. Das Schreiben beschränkt sich also keineswegs auf die Einzelarbeit. Im Gegenteil bietet das gemeinsame Schreiben der SchülerInnen miteinander aber auch mit Einbindung des Lehrers Vorteile, wie etwa dem gesteigerten Motivationsfaktor, die Enthierarchisierung zwischen Lehrer und SchülerInnen und das Fördern von partnerschaftlicher und gruppenorientierter Arbeit. Beispiele sind Handbücher und Dokumentationen zu Projekten, aber auch Evaluationen oder Vorabanalysen. Gerade im Informatikunterricht besteht die Möglichkeit dem Schreiben (etwa von Aufsätzen im klassischen Sinn), fachspezifische und moderne Methoden hinzuzufügen. So stellen Blogs eine aktuelle und für SchülerInnen ansprechende Möglichkeit des Schreibens dar. In Blogeinträgen ist es mit geringem Aufwand möglich einen Verweis auf eine Quelle zu setzen. SchülerInnen können sich so mit dem Zitieren – sicherlich in einer stark vereinfachten Form – vertraut machen. Bezüglich modernen und fachspezifischen Methoden des Schreibens im Informatikunterricht gibt es sicherlich

noch eine Fülle weiterer zu untersuchender Varianten, wie etwa Podcasts, Webquests, etc.

Nachdem nun eine Auswahl an methodischen Konzepten zur Behandlung von geschichtsspezifischen Inhalten der Informatik vorgestellt wurde, sollen im folgenden Abschnitt einige Materialien für den Unterricht verdeutlicht werden. Die wichtigste Anforderung an diese Materialien ist die Möglichkeit historische Ereignisse und Entwicklungen mit heutigen Themen und aktuellen Bezügen der Informatik verbinden zu können. Materialien für den Unterricht aufzuarbeiten, welche dieser Prämisse entsprechen ist dabei keinesfalls trivial. So steht der Abakus für ein frühes Beispiel der Automatisierung und Formalisierung. Konkrete Bezüge zu aktuellen (kern)informatischen Themen und der Lebenssituation der SchülerInnen hingegen sind zunächst nicht offensichtlich.

3 Materialien

Soziale Netzwerke: Die Interaktion mit sozialen Netzwerken gestaltet sich extrem einfach und lässt SchülerInnen auf den ersten Blick vermuten, dass im Hintergrund von z.B. Facebook keine aufwendigen Algorithmen oder Datenstrukturen eingesetzt werden. Jedoch integrieren gerade soziale Netzwerke eine umfangreiche Palette an historischen Algorithmen und Datenstrukturen, um geringe Ladezeiten für Suchanfragen o.ä. zu ermöglichen. Werden Begriffe in Abfragen eingegeben, so erhält man eine sofortige Liste an möglichen Resultaten. Vergleicht man diese Nutzererfahrung nun mit Gegenbeispielen, so kann den SchülerInnen veranschaulicht werden, welchen Unterschied die Verwendung trivialer Algorithmen oder Datenstrukturen ausmachen kann. Ausgaben, wie „Bitte warten Sie bis Ihre Suchanfrage beendet wurde...“, werden von Benutzern heutzutage nicht länger toleriert. Viele Konzepte heutiger sozialer Netzwerke lassen sich bereits sehr früh erkennen. Das *Chatten* beispielsweise, wurde bereits in den ersten Unix-Varianten mit dem Programm *talk(1)* ermöglicht:

„The **talk** utility is a visual communication program which copies lines from your terminal to that of another user.“ – Auszug aus dem *talk(1) manual*

Mit dem Programm *mesg(1)* war es zudem möglich den Empfang von Nachrichten zu gestatten bzw. zu untersagen. Es wurden also bereits Aspekte zur Kontrolle der Privatsphäre berücksichtigt, welche in heutigen sozialen Netzwerken ein zentrales Thema darstellt. Ebenfalls die Nutzung von Mikroblogging-Diensten wie Twitter, wurde in Unix bereits mit dem Programm *finger(1)* und der Erstellung der Datei *~/plan* vorweggenommen. Ein weiterer Themenkomplex, welcher im Zuge von sozialen Netzwerken behandelt werden kann, ist die Datensicherheit. Die Geschichte der Datenverarbeitung ist von Ihren Anfängen bis in die heutige Zeit mit Fällen von Datendiebstahl und Datenmissbrauch gekennzeichnet. Soziale Netzwerke zeigen eine Vielzahl von Sicherheitsaspekten, welche aus Fehlern in der Vergangenheit diesbezüglich gezogen wurden. Aspekte wie SSL, die Hinterlegung von mehreren Email Adressen oder Identitätsverifikation sind nur einige Beispiele. Weiterhin gestattet es etwa Facebook jedem Benutzer, als

Entwickler aktiv zu werden. Dies könnte für SchülerInnen eine erste und didaktisch sinnvolle Einführung in das Programmieren sein. Der Aspekt der Objektorientierung wird von Anfang an fokussiert. Zudem ist es gleich zu Beginn und mit geringen Programmierkenntnissen möglich, grafische Applikationen zu schreiben. In Hochsprachen, wie etwa C oder Java, erfordert dies bereits professionelle Programmierkenntnisse. Soziale Netzwerke erscheinen somit als ein ergiebiger Pool an geschichtsspezifischen Unterrichtsmaterialien für den Informatikunterricht.

Computerspiele: Ein weiteres Beispiel für eine umfangreiche Materialsammlung zu informatikgeschichtlichen Unterrichtsinhalten bieten Computerspiele. Werden ältere Spiele aus den 1970er Jahren betrachtet, so können Algorithmen wie etwa Wegfindungsroutinen oder Algorithmen zur Generierung von zufälligen Ereignissen direkt durch die Betrachtung des Spielablaufes herausgearbeitet werden. Auch bezüglich der Hardware sind Computerspiele ein geschichtliches Zeugnis. So nutzten sie (und tun es noch) sehr stark spezifische Aspekte einer Rechnerarchitektur, um die zur Verfügung stehende Rechnerkapazität optimal zu nutzen. Nur wenige Computerspiele wurden mit dem Ziel der Portabilität entwickelt. Dies unterscheidet ältere Spiele von den aktuellen, welche auf einer Vielzahl von Plattformen (PC, Playstation 3, XBOX 360, etc.) ausführbar sein müssen. Zudem sind Computerspiele Zeugnisse der Softwareindustrie im Allgemeinen und spiegeln die jeweiligen Probleme aber auch Vorteile einer bestimmten Dekade wieder. Während Computerspiele in den 1970er und 1980er Jahren beispielsweise von Hobbyisten in einem Team von wenigen Personen entwickelt und vermarktet wurden, stellt die Entwicklung heutiger Computerspiele ein Projekt mit hunderten von Mitarbeitern und Produktionskosten im zweistelligen Millionenbereich dar. Das Computerspielmuseum in Berlin bietet sich zudem an, um im Rahmen eines Ausflugs die Geschichte der Computerspiele enaktiv und auch interaktiv erleben zu können. Auch lassen sich im Bereich des Computerspiele-Journalismus sehr viel stärker und früher als anderswo die Auswirkungen der Verlagerung von Print Medien auf digitale Medien beobachten. Die zukünftigen Veränderungen in Bezug auf das Lesen, können hier bereits mit den SchülerInnen abgeschätzt und in einen historischen Kontext eingeordnet werden.

4 Ausblick

Zur Behandlung von geschichtsspezifischen Inhalten im Informatikunterricht werden neue Konzepte benötigt, da die bislang existierenden hierfür nicht ausreichend sind. Jedoch lassen sich diese Konzepte teilweise aus den Didaktiken anderer wissenschaftlicher Disziplinen transferieren. Ein Beispiel hierfür sind die Didaktiken der Naturwissenschaften. Insbesondere bei Thomas [Th05] wird deutlich, dass im Chemieunterricht existierende Konzepte zur Behandlung von geschichtsspezifischen Inhalten auch auf den Informatikunterricht übertragen werden könnten. Auch die in diesem Beitrag vorgestellten Konzepte aus der Geschichtsdidaktik scheinen sinnvoll und zur didaktischen Verbesserung des Informatikunterrichts in Bezug auf geschichtsspezifische Lerninhalte beitragend. Konkrete Aussagen über die Vorteile/Einsatzmöglichkeiten von Konzepten anderer Disziplinen bei der Behandlung von Informatikgeschichte können jedoch nur im Kontext unterrichtspraktischer Erhebungen gemacht werden. Welche Konzepte und Methoden sich im Informatikunterricht bewähren, bleibt bis dahin abzuwarten. Eine

weitere Voraussetzung hierfür ist die Ausarbeitung eines Clusters an Unterrichtsmaterialien für die Informatikgeschichte. Diese Aufgabe ist keinesfalls trivial, da Materialien einer Vielzahl an Ansprüchen genügen müssen. Dazu zählen Faktoren wie etwa Enaktivität, ausreichende Quellenlage schriftlichen/visuellen Charakters, Möglichkeiten der didaktischen Reduktion/Rekonstruktion, Einbettung in einen historischen Kontext sowie die Verdeutlichung von Abhängigkeiten mit anderen historischen Ereignissen der Informatik und die Verknüpfung von historischen Materialien und Persönlichkeiten.

Literaturverzeichnis

- [Br92] Brunnstein, K.: Computer-Unfälle. LOG IN, 12. Jg. (1992), Heft 3, S. 17–23.
- [Eu98] Eulenhöfer, P.: Disziplingeschichte und die Disziplinierung der Geschichte. FIF Kommunikation, 15. Jg. (1998), Heft 2, S. 29–33.
- [Fo10] Fothe, M.: Kunterbunte Schulinformatik. Ideen für einen kompetenzorientierten Unterricht in den Sekundarstufen I und II. Berlin, 2010.
- [Gü08] Günther-Arndt, H.: Geschichtsmethodik. Handbuch für die Sekundarstufe I und II. 2. Auflage, Berlin, 2008.
- [Hu09] Humbert, L.: Ideengeschichte oder Archäologie. Geschichte der Informatik – das Unsichtbare ist der Kern. LOG IN, 29. Jg. (2009), Nr. 157/158, S. 20–24.
- [Pa67] Panofsky, E.: *Studies in Iconology. Humanistic Themes In the Art of the Renaissance*. Torchbook Ed., New York, 1967.
- [SS04] Schubert, S.; Schwill, A.: Didaktik der Informatik. Heidelberg u.a., 2004.
- [Sc09] Schwill, A.: Unterrichtshilfen in Informatik. LOG IN, 29. Jg. (2009), Nr. 160/161, S. 14–33.
- [Si01] Siefkes, D.: Schreiben und Geschichte als Zugang zur Informatik. FIF Kommunikation, 18. Jg. (2001), Heft 4, S. 11–13.
- [Th05] Thomas, M.: Vom Abakus bis Zuse. In: Friedrich, S. (Hrsg.): Unterrichtskonzepte für informatische Bildung. INFOS 2005; 11. GI-Fachtagung Informatik und Schule, 28.–30. September 2005 an der TU Dresden. Bonn, 2005, S. 185–196.
- [Th09] Thomas, M.: Skript zur Veranstaltung „Fachdidaktik Informatik – Gestaltung von Lehre und Unterricht“, SS 2009, Abschnitt 3.4.6 Software.
- [UO10] Unterbruner, U.; Otrell-Cass, K.: Wie sich Jugendliche Technik und neue Medien in einer Welt in 20 Jahren vorstellen: Ergebnisse der Studie Jugend-Zukunft-2008/09 mit Jugendlichen aus Österreich, Deutschland und Neuseeland. In: Zumbach, J.; Maresch, G. (Hrsg.): Aktuelle Entwicklungen in der Didaktik der Naturwissenschaften: Ansätze aus der Biologie und Informatik, Innsbruck u.a., 2010, S. 37–53.
- [We91] Weinberg, J.: Didaktische Reduktion und Konstruktion. In: Tietgens, H. (Hrsg.): Didaktische Dimension der Erwachsenenbildung, 1991, S. 130–150.