

Technische Universität Berlin

**Forschungsberichte
der Fakultät IV –
Elektrotechnik und Informatik**

Wozu Informatik?

Theorie zwischen Ideologie, Utopie und Phantasie

Materialien zu einer Arbeitstagung in Bad Hersfeld März 2002

Herausgeber:

Frieder Nake
Universität Bremen
Fachbereich Mathematik/Informatik

Arno Rolf
Universität Hamburg
Fachbereich Informatik

Dirk Siefkes
Technische Universität Berlin
Institut für Softwaretechnik
und Theoretische Informatik

Bericht 2002 - 25

ISSN 1436-9915

Technische Universität Berlin
Fakultät IV - Elektrotechnik und Informatik
Franklinstr. 28/29 - D - 10587 Berlin

Inhalt

Allgemeine Papiere

Einleitung	3	Frieder Nake, Arno Rolf, Dirk Siefkes
Eröffnung der Tagung	5	Dirk Siefkes
Konturen einer Theorie der Informatik	6	Dirk Siefkes

Arbeitsgruppe Kulturelle und Geschlechterperspektiven auf die Informatik

Aufruf	11	Corinna Bath
Positionspapiere		
Gender als Phantasie oder Potential einer Disziplin?	13	Cecile K.M. Crutzen
Professionalisierung – Problemlösung oder vorzeitiger Abbruch der Diskussion?	17	Eva Hornecker
Klare Strukturen sind effizient, Verwirrungen zeigen Möglichkeiten auf	20	Barbara Kleinen
Genderforschung am Beispiel der Informatisierung der Neurowissenschaften	22	Britta Schinzel
Klassifizierungen des Lebendigen	27	Jutta Weber
Bericht	32	Corinna Bath, Jutta Weber

Arbeitsgruppe Theorie der Anwendungen der Informatik

Aufruf	41	Gerhard Wohland
Positionspapiere		
Unser aller Profession gib uns heute ... oder die Frage nach einer mäeutischen Informatik	42	Peter Bittner
Über den notorisch unproduktiven Umgang mit Computersystemen	46	Peter Brödner
Historische, wissenschaftstheoretische und kulturelle Wurzeln der angewandten Informatik	50	Christian Dahme
Anwendungsspiele. Eine Antitheorie der Anwendungen	54	Martin Fischer
Wer nicht nach der Softwarenutzung fragt, bleibt dumm	56	Bernd Pape
Informatiksysteme in Organisationen	60	Arno Rolf
Theorie der Anwendung oder Anwendung der Theorie	61	Peter Schuck
Fragen an die Theoretiker, aus der Praxis eines internen Dienstleisters heraus formuliert	64	Kai Seim
Ist jemand verwundert? Neue Arbeitsmethoden und ihre Folgen	66	Jörg Werda
Bericht	68	Peter Brödner, Kai Seim, Gerhard Wohland

Arbeitsgruppe Informatik als Hybridwissenschaft

Aufruf	85	Dirk Siefkes
Positionspapiere		
Wissens-Verarbeitung in Informatik und Pädagogik	88	Johannes Busse
Mut zum Nachdenken	89	Paul Drews
Was sind Anwendungen der Informatik? Was könnte eine Theorie dieser Anwendungen sein?	91	Jochen Ludewig
Informatik als Hybrid-Wissenschaft? Am Beispiel der Entwicklung wissensbasierter Systeme	93	Frank Puppe
Ökumenische Studien der Informatik	99	Jochen Schneider
Vorstellung der Arbeitsgruppe	101	Dirk Siefkes
Informatik als Methodendisziplin – am Beispiel interdisziplinärer Arbeit mit der Soziologie	102	Rüdiger Valk
Informatik ist Kulturwissenschaft (ja und?)	106	Ulrike Wilkens
Bericht	112	Dirk Siefkes et al.

Liste der TeilnehmerInnen	127	
----------------------------------	-----	--

Einleitung

Frieder Nake, Arno Rolf, Dirk Siefkes

Die Informatik ist als wissenschaftliche Disziplin Ende der sechziger Jahre von Menschen gebildet worden, die vor allem aus Mathematik und Elektrotechnik kamen. Heute versteht sie sich gern als Ingenieurwissenschaft, als Vertreterin einer "definierenden Technologie". Gleichzeitig ist deutlich geworden, dass sie – schneller und direkter als andere (Ingenieur-)Wissenschaften – von Entwicklungen in tendenziell allen sozialen und kulturellen Bereichen ebenso geprägt wird, wie sie diese beeinflusst.

Mathematische Theorie, Ingenieurmethoden und Entwicklungspraxis reichen nicht aus, um diesen lebhaften Wechselwirkungen von Wissenschaftlichkeit und Gesellschaftlichkeit zu begegnen. Vor mehr als zehn Jahren (1988) hat sich daher ein Arbeitskreis des Fachbereichs "Informatik und Gesellschaft" der Gesellschaft für Informatik, gefördert als Diskursprojekt des BMFT, unter Leitung von Wolfgang Coy der Frage gestellt, wie eine "Theorie der Informatik" aussehen könnte, die mit den vielfältigen Bezügen zwischen Informatik und anderen, insbesondere geistes- und sozialwissenschaftlichen Disziplinen besser umzugehen erlaubt (Coy 1989). Auf drei Arbeitstagen in Bederkesa wurde die Diskussion in größerem Kreis geführt. 1992 erschien als Bestandsaufnahme der Band "Sichtweisen der Informatik" (Coy et al. 1992), der die Vielfalt von Zugängen zur Informatik mit Anspruch auf theoretische Begründung betont.

Diese Tradition haben wir etwa ein Dutzend Jahre danach mit einer kleinen Reihe von Arbeitstagen wieder aufgegriffen. Als Vertreter einer Disziplin, die in der Öffentlichkeit meist als allumfassende Technologie wahrgenommen wird, wollen wir unser Selbstverständnis als Informatiker konstruktiv-kritisch reflektieren. Den Anfang machten wir mit:

Informatik: Aufregung zu einer Disziplin
Arbeitstagung Heppenheim, April 2001

Ausgangspunkt war die Aufregung um Informations- und Kommunikationstechnologie und die sich daran knüpfenden Diskussionen um unsere Disziplin. In drei Arbeitsgruppen haben wir Informatik als kulturelle Entwicklung, als technische Semiotik und als gestalterische Aufgabe betrachtet und die Ergebnisse im Plenum geteilt (Nake et al. 2001). Die Arbeit war fruchtbar und die Diskussionen aufregend, so dass zum Abschluss Einigkeit bestand, das Unternehmen fortzusetzen.

Uneinig waren wir uns über einen neuen Namen. 'Theorie', insbesondere im Singular, scheint vielen vorbelastet als 'nicht Praxis', 'weltfern' oder 'regulierend'. Uns Veranstaltern war aber wichtig, "Theorie der Informatik" als Fokus und Orientierung der Arbeitstagung erhalten bzw. noch genauer herausgestellt zu sehen. Mit diesem Ziel haben wir zu folgender Unternehmung eingeladen:

Wozu Informatik? Theorie zwischen Ideologie, Utopie, Phantasie
Arbeitstagung Bad Hersfeld, März 2002

Es gab wieder drei Arbeitsgruppen:

AG Kulturelle und Geschlechterperspektiven auf die Informatik
Corinna Bath, Uni Bremen; Jutta Weber, TU Braunschweig
Ist die Informatik eine Normalisierungswissenschaft? Inwieweit verfestigt sie soziale Normen oder erzeugt globale Klassifikationen? Können Gendertheorien dem entgegenwirken und dazu beitragen, eine Vielfalt in informationstechnischen Systemen zu implementieren?

AG Theorie der Anwendungen der Informatik
Gerhard Wohland, Diebold Deutschland
Ist Informatik so sehr Anwendung, daß Theorie mit Praxis zusammenfällt? Ist das "Wissenschaft als unmittelbare Produktivkraft"? Wo ist dann der Ort der Reflexion?

AG Informatik als Hybridwissenschaft
Dirk Siefkes, TU Berlin
Mit Formalismen hybridisieren wir Mensch und Maschine. Wird eine objektorientierte und internetbasierte Informatik eine "Technik/Struktur/Humanwissenschaft"? Ist sie universal, fraktal oder gar keine Wissenschaft?

Grundlage der gemeinsamen Arbeit waren neben einführenden Texten nicht Vorträge, sondern Positionspapiere, mit denen sich die Teilnehmer in eine der drei Gruppen einordneten und die allen vorher zugänglich gemacht wurden. Nach Abschluß der Tagung erstellten die Verantwortlichen der Arbeitsgruppen zusammen mit TeilnehmerInnen Berichte über die Arbeit und ihre Ergebnisse. Das

vorliegende Heft enthält diese einführenden Texte, Positionspapiere und Berichte. Wir wollen damit die bisherige Arbeit dokumentieren und einem größeren Kreis zugänglich machen.

Eine dritte Tagung findet demnächst statt:

Informatik zwischen Konstruktion und Verwertung.
Kritisches Selbstverständnis aus theoretischer Reflexion
Arbeitstagung Bad Hersfeld, April 2003

Es werden zwei Arbeitsgruppen angeboten:

AG Informatik im interdisziplinären Kontext.
Wie wird Identität konstruiert?
Corinna Bath, Ralf Streibl, Universität Bremen
Ulrike Wilkens, Hochschule Bremen

AG Theorie der Informatikanwendungen in Wertschöpfungsprozessen
Peter Brödner, Institut Arbeit und Technik, Gelsenkirchen
Kai Seim, T-Systems, Darmstadt
Gerhard Wohland, Diebold, Leinfelden-Echterdingen

Die theoretische Reflexion soll uns zu einem kritischen Verständnis der Informatik in ihrer ganzen Breite von universitärer Wissenschaft bis zu verwertbarer Technologie führen.

Literatur

Wolfgang Coy: *Brauchen wir eine Theorie der Informatik?* Informatik Spektrum 12:5, 1989, S. 256-266.
Wolfgang Coy, Frieder Nake, Jörg-Martin Pflüger, Arno Rolf, Jürgen Seetzen, Reinhard Stransfeld, Dirk Siefkes (Hrsg.): *Sichtweisen der Informatik*. Vieweg 1992.
Frieder Nake, Arno Rolf, Dirk Siefkes (Hrsg.): *Informatik - Aufregung zu einer Disziplin. Tagung Heppenheim 2001*. Universität Hamburg, FB Informatik, Bericht 235, 2001.
<http://tal.cs.tu-berlin.de/siefkes/Heppenheim>

Eröffnung der Tagung

Dirk Siefkes

Theorie der Informatik? Die Bezeichnung kommt im offiziellen Titel der Tagung nicht vor; auch 2001 in Heppenheim haben wir sie vermieden. Sie hat sich bei mir aber festgesetzt, ist mir natürlich geworden: Wir arbeiten hier an einer Theorie der Informatik. "Theoriearbeit", sagt Gerhard Wohland.

Als Wolfgang Coy 1988 das Diskursprojekt "Theorie der Informatik" ins Leben rief, war der Name kämpferisch gewählt: Theoretische Informatik ist nur Mathematik, die Informatik braucht mehr Theorie. In einer Gruppe von acht bis zehn Personen haben wir uns regelmäßig getroffen, und es gab drei Arbeitstagungen in Bederkesa. Im Ergebnisband "Sichtweisen der Informatik" haben wir das Programm formuliert ("Grundlagen") und die Sichtweisen Arbeit, Kultur und Verantwortung ausgearbeitet.

Im vergangenen Jahr haben Frieder Nake, Arno Rolf und ich – damals Mitglieder im Theorie-Kreis – in Absprache mit Wolfgang Coy und Wilfried Brauer mit der Arbeitstagung "Informatik – Aufregung zu einer Disziplin" einen zweiten Anlauf gemacht. Aus Anlass der aktuellen Aufregung um I&K-Technologie in Deutschland haben wir gefragt: Was tun wir Informatiker eigentlich? In drei Arbeitsgruppen haben wir die Informatik als technische Semiotik, als gestalterische Aufgabe und als kulturelle Entwicklung betrachtet und Muster in der Vielfalt gesucht.

Bei der Abschlussitzung gab es Ärger: Zu wenig Anwendungsbezug, zu akademisch, zu männlich – drei ältere Professoren, die die Richtung vorgeben. Als Folge kam es zu den beiden ersten Arbeitsgruppen dieser Tagung: "Kulturelle und Geschlechterperspektiven" (Corinna Bath, Jutta Weber) und "Theorie der Anwendungen" (Gerhard Wohland). Die dritte "Informatik als Hybridwissenschaft" biete ich an, weil Arno Rolf und Frieder Nake zu tief in anderen Projekten stecken; aber auch, weil ich am meisten lerne, wenn ich etwas tue.

Am meisten Ärger gab es um den Namen. Die einen oder anderen sagten: Theorie ist bloß

- Ideologie: Wir tun etwas; die stellen Regeln auf, wie es sein soll.
- Utopie: Wir verfolgen Ziele; die schwärmen davon, wohin es führen könnte.
- Phantasie: Wir wissen Bescheid; die reden darüber und haben keine Ahnung.

Ein Logiker, ein Marxist und ein Luhmann-Vexierter, die sich gemeinsam für Theorie begeistern – wohin soll das führen?

Wenn man die Tür aber fragt: Wozu tut Ihr das? Was wollt ihr damit? Welchen Sinn ergibt es für Euch und andere? Dann werden sie je nach Eigenart redselig oder schweigsam. Dann entwickeln sie, in Gedanken oder im Reden, Theorie. Theoria war im alten Griechenland der Bericht derer, die zu den Heiligen Spielen gesandt worden waren. Theoretiker waren Enthusiasten. Sie wollten ihre Begeisterung mit denen teilen, die zu Hause saßen, träge oder frustriert.

Unsere Tagungen sind keine Heiligen Spiele. Wir Teilnehmer sind keine Abgesandten. Wir tun unsere Arbeit als Informatiker wie die zu Hause Gebliebenen. Wir sind hier, weil wir Sinn in unserer Arbeit suchen und die Begeisterung und den Ärger, den wir daraus ziehen, verstehen und kommunizieren wollen.

Theorie soll Sicherheit geben, aber nicht unverrückbare von außen wie eine Ideologie, sondern Gewissheit über den Sinn unseres Tuns.

Theorie soll Richtungen geben, aber nicht eindeutige oder ins Jenseits führende wie eine Utopie, sondern vielfältige hin zu anderen Menschen und Disziplinen.

Theorie soll uns Fragen stellen lassen, aber nicht irgendwelche phantastischen, sondern konkrete, die neue Wege öffnen.

So soll unsere Theorie zwischen Ideologie, Utopie und Phantasie hin und her pendeln: ein bißchen von jedem, aber keines ganz.

Konturen einer Theorie der Informatik

Nach der Tagung: Gemeinsames in den Arbeitsgruppen

Dirk Siefkes

AG Theorie der Anwendungen

Arbeit hat einen formalen und einen kreativen Anteil. Der formale besteht aus sich wiederholenden, daher festgeschriebenen, daher formalisierbaren, daher programmierbaren Vorgängen; der kreative kommt überraschend, kann daher nicht auf den Computer gebracht werden. Diese Unterscheidung soll nicht einen Gegensatz aufbauen, sie weist auf eine Wechselwirkung hin: Der formale Anteil von Arbeit besteht aus Tätigkeiten, die wir gewohnt sind und daher ohne viel Nachdenken immer wieder auf dieselbe Weise tun. Kreativ werden wir, wenn ungewohnte Umstände auftreten - aber nur, wenn die Routine so weiter läuft, dass wir Muße zum und Freude am Denken haben. Angst tötet Kreativität. Langeweile auch: Wenn formale Arbeit so festgelegt ist, dass sie keine Gedanken oder Gefühle anregt, ist sie tot. Kreative Arbeit entsteht also aus formaler, wenn die Umstände günstig sind. Umgekehrt entsteht formale aus kreativer Arbeit, weil wir aus neuen Tätigkeiten Gewohnheiten machen, die alten Routinen ändern oder ersetzen. Kreativität, die nicht in Ordnung mündet, erzeugt Chaos.

Diese Wechselwirkung kann sich am ehesten in kleinen Gruppen aufrechterhalten. In überschaubaren Umgebungen können wir uns auf Gewohnheiten und Einfälle der Beteiligten verlassen. Die Wechselwirkung zwischen beidem hält die Gruppe am Leben, d.h. gleichzeitig beständig und sich verändernd. Auch als Einzelne sind wir immer auf andere bezogen, zumindest in Gedanken. Ohne menschliche Umgebung entwickeln wir uns auf Dauer nicht weiter. Und in großen Organisationen sind alle Vorgänge so festgelegt, dass sie sich nur schwerfällig verändern. Große Organisationen leben aus der Fähigkeit ihrer kleinen Gruppen, im Sinne der Organisation, aber nicht strikt nach ihren Regeln zu arbeiten. ("Paradox der Organisationstheorie")

Informatikanwendungen dürfen daher in großen Organisationen Arbeit nur so weit ersetzen, dass in den kleinen Gruppen die Wechselwirkung zwischen kreativen und formalen Tätigkeiten nicht gestört wird. In tayloristischen Organisationen wird die Kreativität aus der Basis (Produktion) in die Spitze verlegt. Diese starre hierarchische Struktur ist Voraussetzung für den Einsatz von EDV und wird durch ihn verstärkt. In modernen Organisationen arbeiten kleine Randgruppen relativ autonom, offen für die Kommunikation untereinander und nach außen. Konventionelle EDV, die der Arbeit angepaßt wird und sie damit fixiert und hierarchisiert, lähmt ihre Kreativität und bringt das Unternehmen in die Havarie. Gesucht ist moderne EDV, die kreative Arbeit lokal unterstützt.

AG Kulturelle und Geschlechterperspektiven

Beim Handeln verwirklichen wir Ziele. Ziele geben Richtungen vor, nicht einzelne Schritte. Was wir tun, ist genauso durch seinen Ursprung bestimmt: im Kleinen durch unser Wissen und Werten, unsere Fähigkeiten und Gewohnheiten; im Großen durch Traditionen und Theorien, Artefakte und Institutionen unserer Kultur. Diese Orientierungen bestimmen das Verständnis des Ziels wie die Wahl der Mittel und schreiben sich so bewußt oder unbewußt in das Ergebnis unserer Arbeit ein. Von dort beeinflussen sie wieder unsere Orientierungen und Ziele. So entwickelt sich unsere Kultur in Wechselwirkung zwischen realen und mentalen Gegebenheiten. In diesem Sinn sind auch in Technik und Wissenschaft die Charakteristika unserer Kultur eingeschrieben. Technik und Wissenschaft stehen nicht neben der Kultur. Sie entwickeln sich nicht immanent, sondern in und durch die Kultur und die Kultur mit ihnen.

Besonders tief in jeder Kultur sind die Beziehungen der Geschlechter verankert. Geschlechterverhältnisse sind nicht einfach natur- oder gottgegeben, sondern beruhen auf Zuschreibungen – Vorstellungen von Frauen und Männern, die sich mit der Kultur, wenn auch langsam, verändern. Ohne solche Zuschreibungen gäbe es keine Kultur. Je bewußter wir uns ihrer sind, desto besser verstehen wir ihre Entwicklung. – Besonders tief mit unserer Kultur ist die Informatik verquickt. Sie beeinflusst alle Bereiche und wird daher mit Recht eine "definierende Technologie" genannt. Und sie ist spezifisch für unsere auf Technik und Formales fixierte Kultur und nur in ihr denkbar. Wir können die eine nicht ohne die andere verstehen.

Wenn uns Informatik und Geschlechterverhältnisse am Herzen liegen, sollten wir also nach geschlechtlichen Zuschreibungen in der Informatik und nach informatischen Zuschreibungen in den Geschlechterverhältnissen suchen und ihre Beziehungen aufklären. Es gibt keine männliche und weibliche Informatik, der wir das Geschlecht austreiben müßten – so wenig, wie wir Männern oder Frauen ihre Liebe zur Informatik aus- oder eintreiben müssen. Aber die Informatik konserviert kulturelle Zuschreibungen,

je

mehr wir unsere Augen davor schließen. Und unbewußte Einschreibungen konservieren die Informatik.

Zum Beispiel werden in unserer Kultur Technik und Gesetze als männlich und überlegen angesehen, die soziale Sphäre mit ihren unregelmäßigen Beziehungen als weiblich und untergeordnet. Mit Computern bringen wir Maschinen und Formalismen ins menschliche Leben ein. Mit der unbewußten geschlechtlichen Konnotation machen wir aus Softwareentwicklung einen Kriegszug: Informatiker besiegen die nach Hilfe rufende und gleichwohl widerständige Praxis, wie Männer die hilflosen und gleichwohl widerspenstigen Frauen bezwingen. Sehen wir Frauen und Männer als Partner an, die mit ihren Stärken und Schwächen aufeinander angewiesen sind, verstehen wir auch die Wechselwirkung zwischen Informatiksystemen und sozialen Situationen besser, können besser mit ihr umgehen. Soft skills, die z.B. für Partizipation und Teamarbeit in der Systementwicklung benötigt werden, sind dann nicht "wei(bli)che" Fähigkeiten, sondern selbstverständliche Grundlage und Ausdruck einer fruchtbaren Arbeitskultur.

AG Informatik als Hybridwissenschaft

Nach unterschiedlichen Definitionsversuchen zu Anfang besteht jetzt weitgehend Einigkeit, dass die Informatik eine Ingenieurwissenschaft sei: Sie befaßt sich mit Entwicklung und Einsatz von Computersystemen. Die technische Entwicklung ist sicher Aufgabe der Informatik, wenn auch der universitäre Anteil umstritten ist. Immer deutlicher wird aber, dass Erfolg oder Mißerfolg beim Einsatz stark von der Kenntnis des Bereiches abhängt. Mit jedem Bereich, in den Informatikanwendungen Einzug halten, erhebt sich daher neu die Frage, ob die zuständigen wissenschaftlichen Disziplinen ihren Anteil an die Informatik abzugeben oder umgekehrt die Informatik die neu entstehenden technischen Gebiete abzuspalten habe. Der Kampf um die Grenzen der Disziplin wird dadurch erschwert, dass das oben benannte Feld uneinheitlich und inhaltlich kaum zu charakterisieren ist. Die ältesten Teile, Theoretische und Technische Informatik, sind ihren Ursprungsdisziplinen Mathematik und Nachrichtentechnik methodisch enger verbunden geblieben als der Informatik selbst. Frühere Kernbereiche wie Betriebssysteme und Compilerbau sind längst in einer Vielzahl anderer spezieller Systemtypen untergegangen. Softwaretechnik hat sich als zentraler Bereich der Informatik etabliert, ist aber ohne Bezug zu formalen Methoden und technischen Gegebenheiten nicht zu betreiben. Für den Kampf um Ressourcen in zum Sparen gezwungenen Hochschulen ist also die Vorstellung der Disziplin als eines Territoriums mit festen Grenzen, die es zu verteidigen gilt, ungeeignet.

Wir betrachten stattdessen die Informatik als Kultur. Kulturen haben meist kein festes Gebiet inne; sie leben zwischen und mit anderen Kulturen. Sie ziehen ihre Stärke nicht daraus, dass sie sich abgrenzen oder gar bekämpfen, sondern dass sie kooperieren. Sie gewinnen und erhalten ihre Identität aus Traditionen, die sich in Riten und Mythen, in Sitten und Festen äußern. Diese Traditionen werden mündlich und schriftlich weitergegeben, aber reproduzieren sich vor allem im Gebrauch; ohne praktizierende Menschen werden sie zu Museumsstücken. Daher sind Kulturen regional unterschiedlich ausgeprägt, abhängig von Umgebung und Geschichte; Teilgebiete können sich voneinander stärker unterscheiden als von denen benachbarter Kulturen. So entwickeln sich Kulturen, manchmal schnell, oft überraschend; so bleiben sie überlebensfähig.

In dieser Vorstellung braucht die Informatik einen festen Bestand von Methoden, Inhalten und Lehrgegenständen, um überleben zu können. Diese Kanons sind aber nicht ewig, sondern reproduzieren sich im Gebrauch durch die beteiligten Menschen und verändern sich dabei. Es gibt wie in Kulturen keinen hochrangigen Kern, umgeben von Randgebieten, die weniger wichtig wären. Theoretische und Technische Informatik sind Teilbereiche der Informatik, die ihre Methoden aus den Ursprungsdisziplinen beziehen, aber ihre Identität aus der Informatik. Löste man sie aus der Informatik heraus, verlören sie Antrieb und Richtung ihrer Entwicklung. Neue Teilgebiete wie wissensbasierte Systeme, Datenschutz, CSCW, Neue Medien in der Ausbildung, Internet oder Sozionik sind dagegen anderen Disziplinen wie Psychologie, Jura, Arbeitswissenschaften, Pädagogik, Organisationstheorie oder Soziologie ebenso nahe wie der Informatik. Sie leben aus der Verbindung zu jeweils mindestens zwei Disziplinen und können in verschiedenen Hochschulen der einen oder der anderen zugeordnet sein. Die Informatik entwickelt sich in solchen regionalen Kooperationen nachhaltiger als durch die globale Akquisition neuer Kolonien. Auch Gebiete wie Informatikgeschichte, Mensch-Maschine-Kommunikation oder Theorie der Formalisierung können von Informatikern allein genauso wenig betrieben werden wie von Historikern, Psychologen oder Philosophen. Geschichte z.B. als Generationenfolge von immer schnelleren Rechnern oder Hierarchien von Programmiersprachen langweilt Studenten mit Recht; Geschichten von Pionieren erfreuen, aber befehlen sie nicht. In Texten von Informatikern nach kulturellen Orientierungen zu suchen, kann Studenten wie Dozenten, Wissenschaftlern wie Anwendern einen neuen Zugang zur Informatik eröffnen: Eine Ingenieurdisziplin, die nicht nur Probleme löst oder schafft, sondern Teil der Kultur ist, in der sie selber leben. Eine Wissenschaft, die ihre Verantwortung nicht leugnet oder verdrängt, sondern als Herausforderung annimmt.

Arbeitsgruppe

Kulturelle und Geschlechterperspektiven auf die Informatik

Kulturelle und Geschlechterperspektiven auf die Informatik

Aufruf zur Arbeitsgruppe

Corinna Bath, Universität Bremen

Ist in der Informatik die Rede vom Geschlecht, so wird diese meist als reine Frauenfrage interpretiert. Maßnahmen, um eine stärkere Beteiligung von Frauen am Fach zu erreichen, sind zwar noch immer notwendig, möglicherweise sogar nötiger denn je, dennoch dürfen Analysen zum Geschlechter-Informatik-Verhältnis nicht auf dieser Ebene stehen bleiben. Gehen Ansätze hierzu darüber hinaus, so fragen sie zumeist nach Unterschieden: Haben Frauen einen anderen Zugang als Männer zu Informatik und Technik? Würden sie Technologien anders oder besser gestalten? Oder wäre gar eine feministische Informatik möglich? Von der darin vorausgesetzten Zweigeschlechtlichkeit ausgehend ließe sich im Kontext dieser Tagung beispielsweise diskutieren, ob Theorien der Informatik auf die Geschlechter unterschiedlich eingehen müssten.

Statt jedoch nach polarisierten Differenzen zwischen Frauen und Männern, zwischen dem Weiblichen und dem Männlichen zu suchen und damit letztendlich die bestehende Geschlechterordnung zu zementieren, soll in dieser Arbeitsgruppe Vielfalt in den Mittelpunkt gestellt werden - eine Vielfalt von Menschen, die als InformatikerInnen Systeme gestalten oder als NutzerInnen mit diesen Technologien um- gehen, und eine Vielfalt von Mensch-Maschine-Verhältnissen, die sich weder in die Dichotomie von Sozialem und Technischem noch in die häufig gegenübergestellten Kategorien von Entwicklern und NutzerInnen einpassen läßt.

Theoretisierungen der Informatik und insbesondere Bemühungen, das Fach auf seine vermeintlichen, verschüttet gegangenen oder bewusst ausgeblendeten Vorannahmen zu hinterfragen, können eine Analyse der Verstrickungen und des gegenseitigen Hervorbringens von "Technik" und "Geschlecht" nicht umgehen. Die Genderforschung hat in den letzten zwei Jahrzehnten vielfältige und komplexe Vorschläge zur Konzeptualisierung von Geschlecht gemacht. Seither kann Mannsein und Frausein nicht mehr als stabile, lebenslange Größe, die sich etwa am "körperlichen Geschlecht" ablesen ließe, gedacht werden. Vielmehr geht es darum, die komplexen und situierten Herstellungsprozesse von Geschlecht, das 'Doing Gender', genauer zu verstehen, es zu rekonstruieren und dekonstruieren.

Wahrnehmungen und Darstellungen von Weiblichkeit oder Männlichkeit sind nicht nur geprägt von je spezifisch kulturellen Kodierungen, sie können ohne den Rückbezug auf bestehende Vorstellungen nicht stattfinden. Der alltägliche Prozess des Zitierens von Geschlechtlichkeit verfestigt jedoch auch damit einhergehende gesellschaftliche Normen. Er stellt Normalität her - eine Normalität, in der die Annahme von genau zwei Geschlechtern, denen jede Person eindeutig und für immer zugeordnet werden kann, sich ebensowenig hinterfragen lässt wie Attributierungen, Zuschreibungen und Stereotypisierungen, die dazu dienen, das soziale Bezugssystem der Zweigeschlechtlichkeit sicherzustellen.

Die Rekursionsschleifen, Zirkel oder Spiralen, die in unseren Lebens- und Arbeitsprozessen dafür sorgen, dass etwas als selbstverständlich erscheint, sind nicht allein in Bezug auf die Kategorie "Geschlecht" von Bedeutung. Sie können für jede kulturell geprägte und hervorgebrachte Kategorie in den Blick genommen werden, die uns als vertraut gilt. Doch auch in der Frauen- und Geschlechterforschung waren die vielschichtigen Verwicklungen und Überlagerungen von "Geschlecht", "Klasse", "Ethnie", "Alter" und weiterer Dimensionen bereits früh Gegenstand. Seit langem weisen Untersuchungen auf historisch-kulturell gewachsene Prägungen und Unterschiede, d.h. auf die lokale Situiertheit von sozialen Klassifikationsmerkmalen hin. Diese werden von den aktuellen Globalisierungstendenzen teils gegenläufig durchkreuzt. Vor dem Hintergrund der gegenwärtigen politischen Entwicklungen etwa verlangt die Frage, was als "arabische Herkunft" und "islamischer Fundamentalismus" in der "zivilisierten Welt" gedeutet, konstruiert und in Zukunft gelten wird, eine hohe Sensibilität.

Aus dem Blickwinkel der Informatik erscheint vor allem relevant, welche scheinbaren "Gewissheiten" – Kategorien, Denkweisen, Anschauungen – in informationstechnischen Systeme eingeschrieben werden und durch welche Prozesse diese materiell-symbolischen Verwicklungen des Technischen entstehen. Denn sobald kulturelle Einschreibungen implementiert sind und zum Einsatz kommen, werden sie wiederum als soziale Strukturen und Machtverhältnisse wirksam. Technologien erzeugen gesellschaftliche und lokale Effekte, die dazu beitragen, Normalitäten und soziale Normierungen aufrechtzuerhalten. Sie können jedoch auch dazu genutzt werden, Verschiebungen von Bedeutungen und von strukturell-hierarchischen Gefügen oder Verunsicherungen des Alltäglichen zu produzieren.

Dies setzt jedoch eine bewußte Reflexion nicht explizierter Vorannahmen im Prozess der Technikgestaltung voraus.

Die Untersuchung der Zirkel von Technikgenese und -wirkung, in denen sich das Soziale manifestiert, es zitiert und neu hervorgebracht wird, ist innerhalb der Informatik der Angewandten Informatik bzw. dem Bereich "Informatik und Gesellschaft" zugeordnet. Die Problemstellung wird aus einer breiteren kultur- und gesellschaftswissenschaftlichen Perspektive in der Wissenschafts- und Technikforschung betrachtet. Während sich in der Angewandten Informatik der Fokus mehr und mehr auf jeweils einen spezifischen Anwendungskontext des Einsatzes von Informationstechnologien zu richten scheint, demgegenüber ein grundsätzliches Hinterfragen des eigenen Tuns oder die gesellschaftliche Einordnung der Systeme und der Disziplin insgesamt in den Hintergrund gerät, ist die Wissenschafts- und Technikforschung in Deutschland bisher kaum präsent. Ihre Erkenntnisse werden in der Informatik ebenso unzureichend wahrgenommen, wie Geschlechterperspektiven innerhalb der dominanten Diskurse unberücksichtigt erscheinen.

In der Arbeitsgruppe möchten wir bestehende blinde Flecken und Rezeptionssperren überwinden und kritische Positionen der Technikgestaltung, der Genderforschung sowie der Wissenschafts- und Technikforschung produktiv miteinander verknüpfen. Die komplexe Verwobenheit von Kultur, Geschlecht und Technik soll aus unterschiedlichen Perspektiven der bzw. auf die Informatik beleuchtet werden. Ziel wird es sein, Thesen zu möglichen Sichtweisen von Informatik und der Gestaltung von Informationstechnologien zu entwickeln, die der Vielfalt von Menschen gerecht werden oder zumindest näher kommen können.

Dazu wollen wir zunächst informatische Beispiele zusammentragen und vorstellen, um sie auf geschlechtlich stereotypisierende und anderweitig klassifizierende Einschreibungen zu befragen. Es geht also darum, uns einen Einblick in den gegenwärtigen Stand der Informatik zu verschaffen. Was tun wir heutzutage? Welche materiell-symbolischen Artefakte stellen InformatikerInnen momentan her? Auf welchen sozialen Vorannahmen setzen wir/ sie dabei auf? Wie schreiben sich Geschlecht und andere kulturelle Selbstverständnisse in informatische Konzepte und informationstechnische Systeme ein? Wie läßt sich der Prozess der Vergegenständlichung sozialer Strukturen und Normen in Informationstechnologien anhand der ausgesuchten Beispiele besser verstehen?

Im Anschluss möchten wir diskutieren, wie sich die identifizierten oder offensichtlichen normierenden Einschreibungen verhindern, durchbrechen oder zumindest in Hinblick auf ihre Wirksamkeit abschwächen ließen. Wie können wir uns technische Systeme vorstellen, die gegenüber kultureller und geschlechtlicher Vielfalt sensibel sind? Welche Kategorisierungen und Klassifizierungen sollten dabei vornehmlich aufgelöst oder verschoben werden? Wie könnten diese Artefakte entstehen? Welche Strategien und Herangehensweisen wären für ihre Entwicklung nötig? Wie könnte also Technik so gestaltet werden, dass sie nicht zur Stereotypisierung und Normierung beiträgt, dass kulturelle Selbstverständnisse sich nicht in ihr vergegenständlichen und diese damit normalisieren?

Abschließend möchten wir uns nach Möglichkeit mit den Implikationen dieser Überlegungen für die Informatik als Disziplin auseinandersetzen. Welche der diskutierten Problembereiche und Fragestellungen sollten Gegenstand der Informatik sein? Inwieweit wäre es angebracht, hier punktuell oder übergreifend inter- und transdisziplinär zusammenzuarbeiten? Welche Disziplinen oder fachlichen Perspektiven sind dabei besonders grundlegend? Inwieweit sind und sollten Aspekte dieser Thematik im Curriculum integriert sein? Von welchen Erfahrungen der Institutionalisierung – z.B. gute Beispiele, Lehrkonzepte, theoretische Ansätze – kann hierzu bereits berichtet werden?

Als Beiträge wünschen wir uns konkrete Beschreibungen und Analysen wie auch theoretische Betrachtungen. Gefragt sind Fallbeispiele, die kulturelle und geschlechtliche Einschreibungen in aktuelle informationstechnische Artefakte und Konzepte veranschaulichen und als Ausgangspunkt der Diskussionen dienen können. Darstellungen von Einschreibungs- und Wirkungsprozessen, die sozial-kulturelle Einbettungen von Softwaresystemen tiefergehend untersuchen, sind ebenso willkommen wie Vorschläge zu einer Gestaltung von Technik, die die Zweigeschlechtlichkeit und andere soziale Klassifizierungsmerkmale verwerfen. Dabei können eigene Erfahrungen und Erkenntnisse der informatischen Praxis, sozial- und kulturwissenschaftliche Beobachtungen oder auch Theorieansätze die Ausgangspunkte der Betrachtung sein. Nicht zuletzt werden wir uns deshalb auch über Reflexionen der hier umrissenen Thematik freuen, die einen kultur- und geschlechterkritischen Beitrag zur Informatik entwickeln oder auch eine An-Sicht der Informatik darstellen können.

Gender als Phantasie oder Potential einer Disziplin?

Cecile K.M.Crutzen, Open Universiteit Nederland, Heerlen

Von dieser Phantasie fühlt sie sich aufgesogen. Sie sieht ihre Identifikation von einem Phantasieszenario vereinnahmt, in dem sie keinen Platz findet, finden will oder finden kann, und dessen gegebene Positionen (Opfer oder Täter) sie nicht einnehmen will.” [Lauretis97, S. 120]

Genderladung, was ist das?

Eine Gender-Perspektive auf die Disziplin Informatik suggeriert eine Repräsentation von Informatik aus einer Position der außerhalb der Informatik, wobei mit einer speziellen Brille die Gender-Aspekte des Objekts "Informatik" sichtbar gemacht werden. Gender ist aber keine Eigenschaft oder eine Sammlung typischer Fähigkeiten von repräsentierenden und interpretierenden Akteuren im Außenraum vom oder im zu betrachtenden Objekt "Disziplin Informatik" und dessen Produkten. Im Gegenteil, Gender ist ein integrierter Prozeß, der sich im Handeln von menschlichen und nicht-menschlichen Akteuren in der Informatik manifestiert. Jede solche Konstruktion und jede Performance von Gender, auch die in der Informatik, ist situiert, weil Gender seine Bedeutung in der Interaktion von Menschen bekommt und durch das Handeln von Menschen wahrnehmbar wird. Menschen "haben" keinen Gender, sie "tun" Gender [Butler90, S.140]. In ihrem Tun und Unterlassen erzeugen sie ein Gewebe von Bedeutungen, worin implizite und explizite Gegensätze untereinander verknüpft werden [Meijer91, S.109]. Bedeutungen wie "technologisch" und "sozial", "Entwerfen" und "Benutzen", "männlich" und "weiblich" kommen in solchem Gewebe vor und werden an Handlungen konnotiert werden. Durch wiederholtes Interpretieren und Repräsentieren von Handlungen wird auch in einer Disziplin wie der Informatik eine Genderladung konstruiert, in der implizite und explizite Verbindungen mit dem Geschlechtsunterschied gemacht und gefestigt werden. Die so erzeugten Gegensätzlichkeiten und ihre Verbindungen sind meistens symbolisch und haben nicht wirklich etwas mit dem biologischen Geschlecht zu tun. In solchem Gewebe von Gegensätzen manifestiert sich Gender oft als zwei Fäden "Männlichkeit" und "Weiblichkeit", die immer wieder durch dieses Gewebe von Bedeutungen gezogen werden und das Gewebe zusammenhalten.

Die Bewertung von Genderladung

<i>(...)</i>	<i>Ich bin Seinteil</i>
<i>Solange ich webe lebe ich</i>	<i>Und während ich es webe webt es mich.</i>
<i>Webe ohne Taktik Kalkül Weben</i>	<i>Macht' ich es fertig</i>
<i>blind wie das Leben im Frühling</i>	<i>Macht' es fertig mich</i>
<i>Und immer</i>	<i>Darum</i>
<i>auf der Jagd nach dem</i>	<i>aufreißen alles</i>
<i>was ich noch nicht webte</i>	<i>mein Ich aufreißen zerpfücken zerfasern</i>
<i>Immer größer die Gier</i>	<i>und neu zusammenzwirnen</i>
<i>nach dem Nicht-Gewebten</i>	<i>immer wieder neu (...)</i>
<i>je mehr ich schon webte</i>	
<i>Und jedes Weben ist Vernichtung</i>	
<i>des Nicht-Gewebten</i>	
<i>(...)</i>	

Aus dem Gedicht 'Peneleope am Webstuhl'
von Ulla Hahn [Hahn97, S. 38-39].

Eine Genderladung ist nicht positiv oder negativ zu bewerten. Sie ist immer da, weil Gender ein Prozeß ist, der in der Interaktion zwischen menschlichen und nicht-menschlichen Akteuren stattfindet, und in der Interaktion das Gewebe von Bedeutungen benutzt wird und sich ändert durch das Tun und Unterlassen selbst:

Information technology consists of knowledge and social order – and of relations between women and men – built from the starting points of women and men. (...) When men are in a majority, information technology textuality is born more from the starting points of men and reflects more of men's live. [Vehviläinen97, Part One, p. 15]

Einzig zu bewerten ist, ob diese Genderladung stabil oder zu ändern ist und wie sie benutzt wird. Ist sie ein Potential das, das zum Phantasieren über das Ändern des Verhaltens anregt? Oder ist die Genderladung so stabil, daß sie mögliches Handeln, das ändert, verdeckt. Gender als Prozeß kann somit eine "Grundlage bilden für Identitätsbildung" als auch "eine Inszenierung des Begehrens"¹ aller Akteure auch in der Domäne Informatik. So eine Bewertung der Genderladung, ihre Nutzung und die Effekte der Nutzung, kann man die "Genderperspektive" einer Interaktion, einer Gemeinschaft, einer Disziplin nennen.

Ein Ansatz zur Bewertung der Genderladung in der Informatik

Bei einer stabilen Genderladung werden die Gewebe von Gegensätzen aus der Gewesenheit der Erfahrung immer wieder ähnlich konstruiert und mit einer als Gegensatz konstruierten Geschlechtlichkeit verknüpft. Durch Wiederholung von Handlungen und Bedeutungskonstruktionen dieser Handlungen, ohne Hinterfragen und ohne Bezweifeln, wird die Konstruktion der Genderladung und Gender selbst unsichtbar und wird das Handeln selbst als genderneutral qualifiziert. Jede Gemeinschaft und jede Disziplin, die denkt, genderneutral zu handeln, oder dem nachstrebt, hat in ihrer Gewesenheit nur eine stabile Genderladung konstruiert und als selbstverständlich im Handeln integriert. Ist es nicht so, daß es gerade die Informatik ist, die das Wiederholen selbst zur alles verdeckenden Methode gemacht hat? Gehört es nicht zum selbstverständlichen Handeln innerhalb der Domäne Informatik, Gegensätzlichkeiten zu stabilisieren, indem in vielen Methoden der Analyse und der Realisierung "re-use" und Klassifikation befürwortet wird? Durch Produkte wie UML wird, durch die Fokussierung auf Möglichkeiten von "re-use", das Phantasieren als eine Inszenierung des Begehrens unterschlagen:

(...) object-oriented techniques allow us to reuse far more than code. We can reuse requirements, analysis, design, test plans, user interfaces and architecture. In fact, virtually every component of the software engineering life cycle can be encapsulated as a reusable object. [Yourdon/Argila96, p. 6]

Generalisierung und Spezialisierung befördern die Verfilzung von existierenden Geweben, weil sie auf der Konstruktion von Ähnlichkeit basieren und somit die Folge eine erzwungene Gleichstellung ist. Herrscht nicht genau in der Informatik eine Angst vor Ambiguität und dem Unbestimmten? Wird in der Informatik die Genderladung selbst nicht verdeckt, indem Menschliches analysiert und repräsentiert wird mittels Methoden, die ausschließlich zur Herstellung von mathematischen Maschinen gedacht waren. Diese Kolonialisierung der Analysephase durch das Maschinelle (Syntaktische) erzeugt eine Gleichstellung von Maschine und Mensch, und damit "verschwindet"² auch die Genderladung in der Interaktion.

Aber solange die Genderladung in der Informatik durch an der Oberfläche liegende Symptome noch wahrnehmbar ist, bietet sie vielleicht doch noch eine Öffnung, um das Selbstverständliche zu unterbrechen:

The abiding gendered self will then be shown to be structured by repeated acts that seek to approximate the ideal of a substantial ground of identity, but which in their occasional discontinuity reveal the temporal and contingent groundlessness of this 'ground'. The possibilities of gender transformation are to be found precisely in the arbitrary relation between such acts, in the possibility of a failure to repeat, a deformity, or a parodic repetition that exposes the phantasmatic effect of abiding identity as a politically tenuous construction. [Butler90, p. 141]

Aber solche oberflächlichen Symptome machen die Benutzung der Gewebe mit ihrer Genderladung auch gefährlich, denn diese wird in der Informatik oft gebraucht, um mögliche Ansätze des Änderns zu unterdrücken oder in eine bestimmte Richtung zu forcieren. "Benutzerfreundlich" wird definiert als bequem und gleichgesetzt mit "technologischem Fortschritt" [Markussen95, pp. 158-160]. Eine Bequemheit, die patriarchal entschieden wird, indem bedeutungskonstruierendes Handeln aus dem kritischen Raum zwischen Interpretation und Repräsentation weggenommen wird. Der Verbleib in diesem Raum wird minimalisiert.³ Die damit gewonnene Zeit wird gebraucht, die Benutzer immer wieder

¹ Teresa de Lauretis sieht Phantasie nicht als Gegensatz zur Realität, sondern siedelt Phantasie in der Bedeutungsverschiebung zwischen dem Subjektiven jedes Individuums und dem Gesellschaftlichen an. [Lauretis97, S.99]

² Dieses "verschwinden" bedeutet "nicht mehr wahrgenommen werden". Denn man könnte sogar sagen, dass durch die Gleichstellung Mensch und Mann die Verbindung "Technologie und Mann" bekräftigt wird.

³ Sehr illustrativ ist das folgende Zitat, das in irgendeiner Form in sehr vielen Lehrbüchern über die Objektorientierte Methode vorkommt:

As systems become more complex however, it becomes increasingly difficult to explain behavior in an unambiguous manner. (...) one of the reasons for this ambiguity is the inherent ambiguity in any natural language. (...) We can think of natural language as a set of atomic elements assembled by SRS (Software Requirements Specifications) writers into documents. Due to the relatively

zu überschütten mit Handlungen die sie nicht-verstehend ausführen müssen oder die durch Andere ausgeführt werden, ohne daß die Benutzer selber dieses verlangen.⁴

Die Genderladung zur Konstruktion zu nutzen, benötigt einen dekonstruktiven Vorgang. Denn ohne Zerfaserung des Gewebes der Gegensätze wird in jede Neukonstruktion das alte Gewebe wieder hineingeflochten. Während einer Dekonstruktion können Öffnungen für neue Verknüpfungen entstehen. Für einen solchen kritisch-analytischen Dialog sollte man sich in der Informatik geeignete Partner suchen, die noch nicht vollständig in der Informatik sozialisiert sind oder auf dem Weg sind, ihre Sozialisierung zu hinterfragen. Genderstudies wäre als solcher Partner eine geeignete Wahl.

Beide Disziplinen, Informatik und Genderstudies, sind gleich alt, aber ihr Werdegang ist sehr unterschiedlich. Die Informatik ist gestartet mit einer Kenntnis- und Methodenbasis aus Mathematik, Elektrotechnik und Ingenieurwissenschaften im allgemeinen Disziplinen, in denen das kritische Denken hauptsächlich eingesetzt wurde, um einen optimalen Fortschrittsweg zu behandeln. Genderstudies ist dagegen entstanden aus der Kritik an stabilisierten Methoden von Erkenntniserwerb und hat das Kritisieren selbst zur Methode gemacht, die benutzt wird, ein differenziertes Bild von Methoden entstehen zu lassen: In der Methode selbst ist die Kritik eingeschlossen.

Das Nutzen der Genderladung

Das Nutzen der Genderladung verlangt die Konstruktion von kritischen offenen Räumen innerhalb der Informatik, wo die Informatik und ihr Handeln selbst Thema sein kann und wo der Partner Genderstudies nicht nur eingeladen wird aus Neugier, ob man Genderstudies zum "Owner" von bestimmten Problemen machen kann, wie "der Sozial(un)fähigkeit der Informatik" oder "der Anzahl von Frauen in der Informatik". Denn Neugier wird nur charakterisiert durch ein Fernhalten des Anderen, statt daß man den anderen Akteur herausfordert, die eigenen Interpretationen und Repräsentation zu bezweifeln.⁵ Das Zweifeln sollte nicht den Genderstudies überlassen werden, sondern die Genderladung sollte genutzt werden, um in der Informatik das Zweifeln wieder zur Gewohnheit zu machen.

Erfreulich ist es, daß Beispiele von solchen Interaktionen bereits vorhanden sind. Die Informatica Feminale [Oechtering et al. 98, Crutzen et al. 99] bietet viele Ansätze zur Änderung der Informatik. Aber warum wird dieser kritische Raum innerhalb der Informatik nur als nützlich zur Identitätsbildung von Informatikerinnen gesehen und kaum genutzt als "ein kritischer transformativer Raum" [Crutzen00,01], in dem viele Akteure in der Informatik die Phantasie als "Begehren" erleben können? Ein zweites Beispiel, das Seminar "Technik und Geschlecht", ein virtuelles Seminar der Universitäten Freiburg und Tübingen im WS 2001⁶ zeigt, daß durch die Verbindung von Genderstudies und Informatik kritische Fragen expliziert werden können:⁷

– *Ermöglicht nicht erst "Standardisierung und Routine" Verständigung in einer überkomplexen Welt? Wir denken dabei an Beispiele wie Netzwerkprotokolle oder Dokumentenformate. Inwiefern wird in diesen Fällen die angestrebte Offenheit zum Problem, und ist das lösbar?*

– *Worauf soll OO-Realisierung beruhen, wenn nicht auf OO-Analyse? Eine Abstraktion und Vereinfachung muss ja stattfinden.*

– *Die binäre Codierung Entwerfende/Benutzende scheint auch bedingt von der kapitalistischen Wirtschaftsordnung. Wird von daher nicht eine Aufhebung der Trennung Entwerfende/Benutzende durch die sozialen Rahmenbedingungen extrem erschwert?*

In den folgenden Überlegungen der Seminarteilnehmer werden schon Teile eines Gewebes sichtbar:

low level of the atoms and the lack of well-defined semantics for each atom, resulting sentences, paragraphs, and requirements become ambiguous.

One solution to this problem is to visualize a shell around the natural language with well-defined semantics. (...) We could construct another shell around English that would provide a richer set of semantically clear constructs. (...) this type of shell is often called a model. A model simply provides us with a richer, higher level, and more semantically precise set of constructs than the underlying natural language. Using such a model reduces ambiguity, makes it easier to check for incompleteness, and may at times improve understandability. [Davis93, pp. 213-214]

⁴ Hier sind sowohl Handlungen in der Interaktion zwischen Mensch und System wie auch in der Interaktion zwischen Informatiker und Benutzer gemeint.

⁵ *Die Neugier hat nichts zu tun mit dem bewundernden Betrachten des Seienden, (...) ihr liegt nicht daran, durch Verwunderung in das Nichtverstehen gebracht zu werden, sondern sie besorgt ein Wissen, aber lediglich um gewußt zu haben.* [Heidegger26, §36, S. 172]

⁶ <http://mod.iig.uni-freiburg.de/lehre/Ws2001/viror.html> http://www-pu.informatik.uni-tuebingen.de/iug/technik_und_geschlecht.html

⁷ Fragen und Argumente, welche die Seminarteilnehmer Heiko Titz, Vasile Padurean und Jürgen Steimle mit mir diskutiert haben, als Vorbereitung zu einem Seminarbeitrag über die Objektorientierte Methode.

In der Diskussion um Ihre Thesen haben wir festgestellt, dass es der SW-Industrie systembedingt nutzt, die Trennung Entwerfende/Benutzende aufrechtzuerhalten. Wir haben uns dazu folgendes überlegt:

1) Standardisierte IKT-Produkte sind vermutlich billiger und schneller herzustellen als offene Produkte. Die Herstellerfirmen selbst können die Anforderungen festlegen (eine entsprechende Marktstellung vorausgesetzt). Volkswirtschaftlich gesehen ist der EINSATZ dieser Produkte vermutlich nicht zwingend billiger oder besser, aber effizienter für den einzelnen Entwicklerbetrieb.

2) Da die Nutzer nicht gleichberechtigt in den Gestaltungsprozess integriert werden, geraten sie in ein Verhältnis der Abhängigkeit: Nicht die Produkte werden den Nutzern angepasst, sondern umgekehrt. Wenn die Produkte aber gleichzeitig billiger sind, werden Käufer bei gewerblichen Produkten diese Entmündigung hinnehmen, da andere Produkte unbezahlbar teuer sind. Wenn dies auch für Entwicklungstools gilt, dann geraten evtl. sogar jene Designer in Abhängigkeit, die offene Systeme herstellen wollen. Unter Kostendruck gibt es hier eine Sogwirkung zum Standardprodukt nach Standardmethoden für (künstlich geschaffene) Standardnutzer.

3) Wenn die Benutzenden nicht integriert werden, kann ihnen wertvolles Wissen vorenthalten werden.

4) In einem weiteren Schritt gelten die Erfolgskriterien für Software "gut" , "innovativ", "sicher" , "zuverlässig" schliesslich auch für die Arbeitnehmer/innen in der IT-Branche. Die von Ihnen angesprochene Meßbarkeit und Planbarkeit der Produkte nach diesen Kriterien wird auch auf die Entwickler/innen übertragen und auch auf die Nutzer/innen.

Wenn also die Privatwirtschaft kein Interesse an einer Aufhebung der Trennung Entwerfende/Nutzende hat, dann dürfte die Realisierung Ihrer Forderungen im kommerziellen Bereich noch schwerer fallen als in der Informatik (als wissenschaftliche Disziplin). Die Binarität wird nicht nur in der Informatik-Disziplin, sondern auch ausserhalb ständig verstärkt. Eine Dekonstruktion wendet sich damit auch gegen die gesellschaftlichen Rahmenbedingungen – und stößt auf (mehr oder weniger starken) Widerstand.

Schluß

Die Genderladung von Informatik sollte behutsam gepflegt werden, denn vielleicht ist sie das einzige Potential, das uns noch zum Phantasieren anregen kann. Einem Phantasieren, das nicht der Wirklichkeit gegengesetzt ist, aber Wirkung hat.

Literatur

- Butler, Judith 1990: Gender trouble: Feminism and the subversion of identity. New York: Routledge.
- Crutzen, C.K.M. (2000): Interactie, een wereld van verschillen. Een visie op informatica vanuit genderstudies [*Interaktion, eine Welt von Verschiedenheiten. Eine Sicht auf die Informatik aus der Perspektive der Frauenforschung*]. Dissertation, Open Universiteit Nederland, Heerlen.
- Crutzen C.K.M. (2001): Dekonstruktion, Konstruktion und Inspiration. In: FIF Kommunikation, 3/2001, S. 47-52.
- Crutzen, C.K.M./Vosseberg, Karin (1999): Die Interaktion zwischen objektorientiertem Denken und feministischer Kritik – eine dynamische Verbindung. In: Dreher, B. et al. (eds.), Software Engineering im Unterricht der Hochschulen (SEUH '99). Stuttgart, Teubner Verlag, S. 149-165.
- Davis, Alan M. (1993): Software Requirements. Objects, Functions and States. Englewood Cliffs, New Jersey, Prentice Hall.
- Hahn, Ulla (1997): Galileo und zwei Frauen. Gedichte. Stuttgart, Deutsche Verlagsanstalt.
- Lauretis, Theresa de (1997): Das Subjekt/Sujet der Phantasie. In: Kravagna, Christian (Hrsg.), Privileg Blick. Kritik der visuellen Kultur. Berlin, Edition ID-Archiv, S. 98-124.
- Markussen, Randi (1995): Constructing easiness – historical perspectives on work. In: Star, Susan Leigh (1995), The Cultures of Computing: Invisible Work und Silenced Dialogues in Knowledge Representation. S.158-180.
- Meijer, Maaïke (1991): Binaire opposities en academische problemen. In: Tijdschrift voor vrouwenstudies 45, jrg. 12, nr. 1, S. 108-115.
- Oechtering, Veronika/Rügge, Ingrid/Vosseberg, Karin (1998): Informatica Feminale – Das Informatikstudium anders gestalten. In: Claus, Volker (Hrsg.), Informatik und Ausbildung. Informatik aktuell. Heidelberg, Springer-Verlag, S.143-154.
- Yourdon, Edward/Argila, Carl (1996): Case Studies in Object-Oriented Analysis and Design. Yourdon Press, Upper Saddle River, S. 6.

Professionalisierung – Problemlösung oder vorzeitiger Abbruch der Diskussion ?

Eva Hornecker, Universität Bremen

Ist Informatik eine Profession? Wie kann sie eine werden? Das allein wirft schon viele Fragen auf. Im anglo-amerikanischen Raum ist diese Diskussion zur Zeit virulent (Denning 2001), und es gibt zahlreiche Bestrebungen, zumindest Teilbereiche wie das Software-Engineering zu professionalisieren (Ford & Gibbs 1996). Die Diskussion schwappt auch auf Deutschland über (Schinzel & Kleinn 2001), scheint sie doch einen Lösungsansatz für das allgemeine Unbehagen über die mangelnden Qualitätsstandards der Softwareindustrie zu bieten, der die IT-Professionals in ihre Verantwortung ruft. Ich möchte mich hier kritisch damit auseinandersetzen, ob es überhaupt wünschenswert wäre, die Informatik zu professionalisieren, bzw. in welchem Sinne von 'professionell' dies ratsam wäre.

"Professionell" im Alltags-Sprachgebrauch

"Professionell" heißt zunächst, das eigene Handwerk zu verstehen, und es gut und zum Nutzen des Auftraggebers – und letztlich auch zum eigenen Nutzen – auszuführen. Da ist man oft versucht, mit einem Stoßseufzer auszurufen: Ja wenn die Informatiker doch wenigstens das könnten! Daß Professionalität jedoch zunächst nichts mit ethischem Verhalten zu tun hat, wird deutlich, wenn man bemerkt, daß es auch professionelle Diebe und Killer gibt. Auch diese verstehen ihr Handwerk und üben es im Dienst ihres Auftraggebers aus – ohne Rücksicht auf die Gesellschaft und andere Menschen. Dabei können sie durchaus einem professions-internen Ehren-Kodex folgen. Zur Professionalität im normalen Sprachgebrauch gehören auch gewisse Verhaltensweisen. So manches, was nicht mit der eigentlichen Aufgabe zu tun hat und deren Erfüllung nicht notwendigerweise beeinträchtigt, gilt als unprofessionell. Gegenüber einem Kunden zuviel von sich selber preiszugeben und die Persönlichkeit nicht der Berufsrolle unterzuordnen, gilt häufig als unprofessionell. Jede Profession hat somit einen eigenen Habitus, der im Zuge der Sozialisierung angeeignet wird. Merkmale gemäß des Alltagsverständnisses wären somit Qualitätskriterien, spezifische Verhaltensweisen, Habitus, Ausübung einer Rolle.

Völlig falsch sind solche Forderungen jedoch nicht, denn es soll ja die Rollenerfüllung gewährleistet werden – ein Arzt, der dem Patienten von seinen Problemen erzählt, wird seinem Auftrag nur schwierig gerecht. Notwendig ist die Rücksicht auf Erwartungen und Bedürfnisse des Gegenübers. Umgekehrt bietet die Rolle auch einen Schutz für den, der sie ausübt. Wenn es als unprofessionell gilt, ein Gewissen zu haben, Verantwortung zu empfinden und daraus Konsequenzen zu ziehen, Gesetze einzuhalten (wie im Fall des Profi-Killers), dann sollten wir nicht mehr Professionalität fordern. Gute Kenntnis des Handwerks – ja, aber keine blinde Auftragserfüllung und reine Rollenausübung.

Das wissenschaftliche Professionsverständnis – in der US-Amerikanischen Version

Im aus dem anglo-amerikanischen Raum stammenden Verständnis von Profession sind deren Merkmale die Kodifizierung und Standardisierung von Wissen, damit seine Verwissenschaftlichung, die Kodifizierung von Methoden, die Selbstkontrolle über den Zugang neuer Professionals sowie die Möglichkeit des Ausschlusses (Akkreditierung, Zertifizierung, Lizenzierung) sowie ethische Standards (Denning 2001, Ford & Gibbs 1996). Diese Definition wird von vielen deutschen AutorInnen übernommen (Hartmann 1995, Schinzel & Kleinn 2001), obwohl Alternativen denkbar sind. Ansätze zu einem anders gearteten Professionsbegriff, die im deutschsprachigen Raum in den Sozialwissenschaften entwickelt wurden, sind bisher jedoch nicht von den technischen Disziplinen wahrgenommen worden.

Historisch stand die Etablierung einer Profession weniger in Zusammenhang mit der Gewährleistung von Qualitätsstandards (wie heute vielfach davon erhofft), sondern vielmehr war der "Hauptzweck jeglicher Professionalisierung [...] die Monopolisierung von Teilmärkten zur Erringung und Stabilisierung hoher Einkommen und eines hohen gesellschaftlichen Status" (Hartmann 1995, S. 17) bzw. die Reservierung eines Berufs für eine exklusive Elite und die Abgrenzung von Zuständigkeitsfeldern (Schinzel & Kleinn 2001). Ein Beispiel hierfür war der Kampf der medizinischen Profession gegen den Hebammen-Beruf und das Heilkräuterwissen. Monopolisierung impliziert Ausschluß, im Falle der

Informatik den Ausschluß der Quereinsteiger. Wollen wir dies wirklich? Denn rein zahlenmäßig geht es nicht ohne sie, und inhaltlich werden sie vielfach dringend ob ihres Anwendungswissens benötigt.

Allerdings sollte man nicht vergessen, daß die hohe Zahl von Nicht-Informatikern im IT-Bereich es erschwert, zu einem Selbstverständnis zu kommen und die eigene Kompetenz (als IT-Profession) zu erkennen. Dies gilt sowohl für die Informatiker selber wie für Arbeitgeber und Kollegen. Gerade kleine und mittlere Unternehmen wissen häufig nicht, welche Qualifikation sie von Informatikern erwarten können/dürfen. Sie wissen nichts von dem eher abstrakten, strukturierenden Denken sowie der Methodik, die im Studium geübt werden.

Ironischerweise laufen Berufsgruppen, die sich derart standardisieren, laut Hartmann (1995) Gefahr, Opfer eben dieser Standardisierung zu werden. Denn Standardisierung und Normierung machen einen gewissen Teil des Wissens zugänglich und leicht erlernbar. Durch Standardisierung kann immer stärker zwischen anspruchsvollen (interessanten) Tätigkeiten, welche nur wenigen Professionals zugestanden werden, und Routine-Tätigkeiten unterschieden werden. Je stärker die Software-Entwicklung methodisch standardisiert wird, desto kleiner wird der Vorsprung der diplomierten Informatiker und ihres Methodenwissens gegenüber anderen Entwicklern, die sich ja ebenfalls an die vorgeschriebenen Entwicklungsprozesse halten müssen (Hartmann 1995, S. 139-140). In allen Bereichen außer forschungsnahen sowie der Herstellung von Programmierertools und -verfahren wäre die Konkurrenz massiv, da ihre Kenntnisse über anwendungsbezogene, betriebliche Prozesse zu gering sind. Die Industrie wird also nur äußerst unwahrscheinlich auf den Vorschlag eingehen, die Software-Entwicklung nur in die Hände von IT-Professionals zu geben! Abgesehen davon, daß es gar nicht genug ausgebildete IT-Professionals gibt,¹ würde dies die anwendungsbezogene Qualität von Software gefährden (Software für sich allein betrachtet versus Software im Anwendungskontext). Twisselmann (2000) beschreibt anschaulich die Probleme, die sich durch die Trennung von Anforderungsanalyse und Implementierung sowie den fehlenden Bezug zu Endanwendern ergeben. Sie fordert daher, die Ausbildung stärker auf Beratungstätigkeiten auszurichten und vor allem eine Sensibilität dafür zu wecken, wie Informatiksysteme die Arbeitsabläufe anderer massiv beeinflussen.

Manche Autoren plädieren dafür, die Informatik explizit auf die Gebiete zu beschränken, in denen sie keinen Anwendungsbezug hat (also genau die Bereiche, in denen Informatiker nach Hartmann einen Vorteil haben) und alles Weitere den Anwendungsfächern zu überlassen.² Reduziert sich die Informatik aber auf den kleinen Bereich der Systemprogrammierung und Tool-Entwicklung und überläßt alles Anwendungsbezogene anderen Fächern und Bindestrich-Informatikern, verliert sie m.E. vieles von dem, was sie spannend macht. Sie würde alle von außen an sie herangetragenen Ansprüche von sich weisen und sich völlig von der gesellschaftlichen Entwicklung abtrennen.³

Betrachtet man einige Quellen der amerikanischen Professionalisierungsdiskussion genauer, fällt auf, daß diese oft davon ausgehen, daß die Informatik bzw. das Software-Engineering eine Ingenieurwissenschaft ist, und damit sozialwissenschaftliche Aspekte mehr oder minder indirekt ausschließen (z.B. Ford & Gibbs 1996, Parnas 2001). Das Bestreben, die Informatik näher an die 'respektablen' Ingenieursdisziplinen zu rücken (und weg von den 'weichen' Sozialdisziplinen), zeigt sich in der Umbenennung vieler potentiell 'sozialorientierter' Themen der Informatik in z.B. "Requirements Engineering" oder "Usability Engineering". Britta Schinzel wies in ihrem Ringvorlesungsvortrag auf der Informatica Feminale 2001 (Schinzel 2001) darauf hin, daß damit ein sehr eingeschränkter Technikbegriff propagiert wird, der sozialwissenschaftliche Methoden ausschließt (obwohl sich Ingenieursvorgehen eigentlich daran orientiert, adäquate Problemlösungen zu finden und entwickeln), der adäquate Problemlösungen durch den Ausschluß dieser Methoden verhindern kann und Gender-Relationen re-produziert (vergl. auch Crutzen 2000). Informatik ist aber eine Grenzdisziplin, da sie soziale Realität auf formale Modelle überträgt, sie dazu zuerst verstehen muß, und informatische Systeme in soziale Kontexte einbettet (Schinzel & Kleinn 2001, Crutzen 2000). Sie ist somit auf die Einbeziehung der Grenzgebiete und sozial/geisteswissenschaftlicher Fähigkeiten angewiesen. Auch Denning (2001) stellt zwar fest, daß nicht der Computer selber Fragegegenstand sei, sondern "the computers surround the concerns", und fordert, die Informatik müsse lernen, sich mit Anwendern, Klienten und Benutzern zu verbinden und auf sie zu hören. Punkte wie der zuletzt genannte kommen

¹ Nur 20% der in der IT beschäftigten sind einschlägig ausgebildet! (Schinzel & Kleinn 2001).

² Parnas (2001) unterscheidet z.B. zwischen Computer Science und Software Engineering, wobei unter Software Engineering nur die hardwarenahen Bereiche mit Verbindung zur Maschinensteuerung fallen (eine Art Systems Engineering oder Mechatronik), sich die restliche Angewandte Informatik jedoch weder dem einen noch dem anderen zuordnen läßt. Manche Autoren vertreten gar die Ansicht, die Informatik auf den Kernbereich der Forschung und Methoden- bzw. Toolentwicklung zu beschränken und für die angewandten Bereiche lieber gleich eigene Disziplinen zu etablieren oder diese den Anwendungsfächern zuzuschlagen.

³ Abgesehen davon wäre diese Reduktion (im etwas holprigen Vergleich) so, als würde sich der Maschinenbau auf Motoren oder Kurbelwellen und die E-Technik auf Transistoren und Halbleitertechnik beschränken. Statt einer Konzentration auf einen "Kern" der Disziplin wäre dies eine Spezialisierung auf allgemein verwendete Grundelemente.

aber immer an letzter Stelle einer Aufzählung der Konsequenzen bzw. der Themen, die "professionell societies" angehen sollten, während klassische, dem Ingenieursdenken verhaftete Aspekte zuerst genannt werden (dies gilt auch für Studienplanentwürfe des ACM).

Es scheint zudem, als ob viele der Autoren, die den desolaten Zustand der IT-Industrie und die mangelnde Qualität von Software beklagen, unter Qualität vor allem klassische Qualitätsmerkmale des Software-Engineering verstehen (# bugs, safety, reliability, speed performance, reusability) (Booch 2001, Constantine 2001, Denning 2001). Anwendungsbezogene Qualitätsmerkmale werden selten genannt, obwohl sich die Sicherheit eines Systems oft nur im Kontext seiner Nutzungspraxis erweist. Ähnliches gilt für die Qualität der Software-Entwicklungsprozesse, hier werden eher "specifications, reviews, test plans" (Booch 2001), "discipline" und "systematic techniques of construction" (Constantine 2001) genannt, als daß auf die Existenz von partizipativen oder benutzerorientierten Methoden auch nur verwiesen wird.

(Ford & Gibbs 1996) vollziehen zudem einen logischen Kurzschluß, wenn sie von der Reife (maturity) einer Profession (gemessen an ihrer Infrastruktur und der Erfüllung der Professions-Kriterien) auf den Wissenstransfer in die Praxis sowie auf Innovationsoffenheit schliessen – Kodifizierung und Standardisierung können ebensogut konservativ wirken! Gerade in der sich ständig wandelnden und von den Anforderungen der Anwendungsgebiete vorwärts getriebenen IT erscheint eine Kodifizierung schwierig und eher hinderlich. Die Nachteile der Professionalisierung und Kodifizierung zeigen sich z.B. in der schwierigen Anerkennung von Naturheilmethoden und Akupunktur in der Medizin sowie von neueren Therapiemethoden als Form von Psychotherapie. Während so einerseits eine gewisse Sicherheit für Klienten geschaffen wird, medizinische oder seelische Probleme einer vertrauenswürdigen, dafür ausgebildeten Person anzuvertrauen, werden andererseits oft Grabenkämpfe zwischen traditionellen und alternativen neueren (Therapie-)Methoden auf dem Rücken der Klienten ausgetragen.

Informatiker sind, solange ihre Ausbildung ist, wie sie ist, auf interdisziplinäre Zusammenarbeit und gemischte Entwicklungsteams angewiesen. Zum einen ergänzen sich in solchen Teams die Qualifikationen, und zum anderen haben Informatiker in ihnen Lernmöglichkeiten, die in fachlich homogenen Teams nicht bestehen.

Ein Fachgebiet wie "Informatik und Gesellschaft" (bzw. "Informatik im Kontext") proklamiert eine Öffnung der Informatik für ihren Kontext und möchte Informatik-schaffende für diesen Kontext sensibilisieren. Dies widerspricht dem amerikanischen Professionskonzept. Die Diskussion um die Professionalisierung kann wichtige Denkanstöße liefern, da sie zum Nachdenken über all das, was nicht gut läuft, und über Verbesserungsmöglichkeiten auffordert. Sie kann aber auch einen vorzeitigen Abbruch der Diskussion ("premature closure") bewirken, mit dem Risiko, ein unausgereiftes Selbstverständnis festzuschreiben.

Literatur

- Booch, Grady: Developing the Future. CACM Vol. 44, No. 3, March 2001, S. 119-121.
Constantine, Larry L.: Back to the Future. CACM Vol. 44, No. 3, March 2001, S. 126-129.
Crutzen, Cecile: Interactie, een wereld van verschillen – Een visie op informatica vanuit genderstudies, Dissertation, Open University Heerlen, NL 2000.
Denning, Peter J.: Who Are We? CACM Vol. 44, No. 2, February 2001, S. 15-19.
Ford, Gary, Gibbs, Norman E.: A Mature Profession of Software Engineering. Technical Report, CMU/SEI-96-TR-004 ESC-TR-96-004, Software Eng. Institute, Carnegie Mellon University, September 1996.
Hartmann, M.: Informatiker in der Wirtschaft – Perspektiven eines Berufs. Springer 1995.
Lorge Parnas, David: Wege und Irrwege der Informatik-Ausbildung. Vortrag auf der FlfF-Jahrestagung. Bremen 2001.
Schinzel, Britta: Bedeutung des Technikbegriffs in der Informatik. Vortrag im Rahmen der Ringvorlesung der Informatica Feminale 2001, 5.9. 2001.
Schinzel, Britta, Kleinn, Karin: Quo Vadis, Informatik? Informatik Spektrum 24(2) 2001, S. 91-97.
Twisselmann, Ute: Informatik und Arbeitsumgebungen. FlfF-Kommunikation 1/2000, S. 28-30.

Klare Strukturen sind effizient, Verwirrungen zeigen Möglichkeiten auf

Versuch eines feministischen Blicks auf Informatiksysteme

Barbara Kleinen, Universität Lübeck

Eigentlich wollte ich einen Versuch starten, eine Verbindung aufzuzeigen zwischen Klassifikationen in der Modellierung von Informatiksystemen, insbesondere objektorientierten Technologien, und der Klassifikation von Menschen in Genderkategorien. Weiter reizte es mich, die Trennung von Entwicklung und Benutzung von Informatiksystemen unter einem genderspezifischen Blickwinkel zu beleuchten. Beides hat Cecile Crutzen in einem Artikel in der FlfKO zum Schwerpunkt "Frauen in der Informationsgesellschaft" weitreichender getan, als es mir möglich gewesen wäre [Crutzen01]. Was mir daher bleibt, ist zunächst ein Kommentar zu der im Artikel enthaltenen Kritik an der Software-Ergonomie. Hierauf aufbauend möchte ich meinen Ansatz, Kategorien in einem Wissensarchiv für die Lehre (Koln) durchlässiger zu machen, darstellen.

"Der Informatiker legt sich selbst und den Benutzer bereit als jemand, der in einer Routine von bereitgelegtem geschlossenem Handeln erstarrt ist." [Crutzen01, S. 51]

Ausgehend von einem emanzipatorischem Anspruch, der Benutzerin die volle gestalterische Macht über die universelle Maschine Computer zu geben (oder nur zu lassen?), um eine Variation des Handelns zu ermöglichen, gerät eine schnell an die Frage, wie viel Freiheit der Benutzerin zuzumuten ist. Vielfältigkeit und Möglichkeiten sind immer komplizierter als ein einfacher, vorgegebener Weg, und es scheint so zu sein, dass "der Benutzer" lieber den einfacheren, vorgegebenen Weg sucht. Dies ließe sich an Beispielen wie der Entwicklung des WWW zunächst über eine Vereinfachung der Beschreibungssprache (von SGML zu HTML) und dann der Navigationsstrukturen (von freier Verlinkung hin zu geschlossenen Services mit hierarchischer Navigation) sowie der Verbreitung von Plattformen wie Linux vs. Windows-Variationen beleuchten. Frauen scheinen hier keine Ausnahme zu sein, wie die Erfahrungen meiner Freundin als Internetserviceproviderin für Frauen zeigen. Im Gegenteil, es scheint so, dass gerade Frauen ein System wünschen, das "auf Knopfdruck" einfach nur funktioniert. Der Aufwand, sich das System selber anzueignen, erscheint ihnen unnötig.

Hier liegt also ein Widerspruch vor, zwischen dem Anspruch, Benutzerinnen die Gestaltungsmacht zu geben, und dem offensichtlichen Bedürfnis der Benutzerinnen, in erster Linie ein unauffälliges System zu haben, ohne Wert auf eine Gestaltungsmacht zu legen.

Offene Systeme benutzbar machen

Ein weiteres Beispiel eines in jeder Hinsicht offenen Systems ist Squeak, eine moderne Implementierung von Smalltalk 80 [Guzdial01]. Diese ist nicht nur "Open Source", sondern gibt der Benutzerin durch ihre Architektur, u.a. der in Smalltalk üblichen fehlenden Trennung zwischen Entwicklungs- und Laufzeitumgebung, die Möglichkeit, jederzeit jeden Aspekt des Systems zu ändern. Ist hier die Trennung zwischen Entwicklern und Benutzern aufgehoben? Sicherlich insofern, als das es sich empfiehlt, zur vollen Ausnutzung der Fähigkeiten auch als Benutzerin über Smalltalk-Entwicklungskenntnisse zu verfügen. Ohne Squeak kritisieren zu wollen, und jederzeit bereit, den Standpunkt zu verfechten, dass zumindest grundlegende Smalltalk-Kenntnisse zur informationstechnischen Grundbildung gehören, ist festzustellen, dass hierin eine Benutzungshürde für Systeme wie Squeak besteht.

Wie lässt sich daher der Widerspruch zwischen erwünschter Freiheit und hierzu benötigter Kompetenz auflösen?

Im Forschungsprojekt WissPro (Wissensprojekt Informatiksysteme im Kontext) [WissPro] verfolgen wir die Idee, in einem gemeinsamen Prozess mit den Lernenden und Lehrenden – unseren BenutzerInnen – informationstechnische Unterstützung der Präsenzlehre in der Informatik zu gestalten.

Der hierzu in Lübeck entwickelte Prototyp Koln (Knowledge Infrastructure) verfolgt dabei den Ansatz, den Benutzerinnen die Informationsmodellierung ihres Wissensarchivs in die Hand zu geben. Benutzern soll so nicht nur die Möglichkeit gegeben werden, eine eigene Interpretation der gefundenen Objekte zu verwirklichen (vgl. [Crutzen01]), sondern eben diese Objekte aufbauend auf der eigenen Interpretation kooperativ zu entwerfen. Um die oben beschriebenen Benutzungshürden eines offenen Systems abzubauen, werden zwei Strategien verfolgt. Zum einem wird ein niedrighschwelliger Einstieg

geboden, indem an allgemein bekannten Metaphern (Nachrichtenübermittlung und Speichern von Dateien) angeknüpft wird, die zunächst die Offenheit des Systems verbergen. Zum anderem ist es möglich, Informationen festzuhalten, ohne sie in Ordnungsstrukturen einordnen zu müssen. Die kooperative Konstruktion der Informationsmodellierung, d.h. der komplexe Teil der Benutzung, der in die Gestaltung übergeht, findet in moderierten Prozessen gemeinsam mit den Entwicklerinnen statt.

Der Prototyp wird im kommenden Sommersemester erstmals in dieser Form in Lübeck eingesetzt werden. Bislang konzentrieren sich meine Erfahrungen in Bezug auf flexible Klassifizierungen auf technische Aspekte. Ich bin gespannt darauf, wie die Lehrenden und Studierenden auf die Gestaltungsmöglichkeiten reagieren, mit ihnen umgehen und wie – und ob überhaupt – sie genutzt werden.

Literatur

Crutzen, Cecile K. M.: Dekonstruktion, Konstruktion und Inspiration. In: *FifF Kommunikation 3 / 2001*. 47-52.

Guzdial, Mark: *Squeak. Open Personal Computing and Multimedia*. Upper Saddle River: Prentice-Hall, 2001.

WissPro: <http://www.wisspro.de>

Genderforschung am Beispiel der Informatisierung der Neurowissenschaften

Britta Schinzel, Universität Freiburg

1. Warum Genderforschung Informatik?

Die äußerst geringe Beteiligung der Frauen in informatischen Studien- und Ausbildungsgängen und in den diese Fächer gestaltenden Berufen hat Auswirkungen auf Forschungsprozesse und Fachkulturen, bedingt aber auch methodische und inhaltliche Verengungen, Ausgrenzungen und Verzerrungen in Wissenschaft und Entwicklung, sowie vermittelt in der Gesellschaft. Die von technikzentrierten Interessen dominierte Gestaltung vertieft die Geschlechterpolarisation nicht nur in der IT-Branche, sondern in allen Berufen, die Computer nutzen.¹ Zugangs- und Nutzungsmöglichkeiten kommen daher vorwiegend technizistischen Bedürfnissen entgegen. Hinzu kommen Probleme der Repräsentation: Abstraktion und Kategorisierung beruhen immer auf *Generalisierungen*. Die Eigenschaft oder Gefahr der Abstraktion, Normierungen, Übergeneralisierungen, Überspezialisierungen (wenn die der Abstraktion zugrunde gelegte Beispielmengende nicht genug gestreut ist) und Standardisierungen festzulegen, wird für Frauen dann problematisch, wenn sie mit androzentrischem Blick erfolgt.

Damit schließt sich der Kreis, in dem einseitige Interessen einseitige Gestaltung hervorbringen und die so gestalteten IT-Artefakte erneut Menschen mit bestimmtem Habitus ansprechen und andere ausschließen oder die Handhabung erschweren. Genderforschung will das beschriebene "Gendering" durch IT-Produkte daher sowohl im Konstruktionsprozess (als EntwicklerIn) wie auch bei der Anwendung (als NutzerIn) aufschlüsseln und die so gewonnenen Ergebnisse durch Entwicklung gendersensitiver IT-Produkte und IT-Systeme konstruktiv wenden. Nur auf diese Weise können Frauen langfristig nicht nur als partizipierende, sondern auch als gestaltende Subjekte in Informatik und Technik/Naturwissenschaft eingebunden werden.

Genderforschung in Technik und Naturwissenschaft stellt eine Reihe von epistemologischen, ethischen und sozialpolitischen Ansprüchen an ihre disziplinäre Forschung: z.B. die auf Offenlegung der individuellen oder forschungsschulischen ethischen und sozialen oder anderen Ziele, auf Reflexion der epistemischen Einbettung, der unausgesprochenen Hintergrundannahmen, auf Beobachtung der Forschungspraxis und schließlich auf Darstellung und Wirkung der Ergebnisse, auch in der Öffentlichkeit.

2. Beispielkontext Informatisierung der Neurowissenschaften und der Biomedizin

Die Informatik bestimmt mit ihren Modellbildungen und Mathematisierungsparadigmen ebenso wie mit ihren Visualisierungstechniken zunehmend Form und Darstellung wissenschaftlichen Wissens. Sie stellt nicht nur eine Sprache oder Sprachen bereit, sie formt wissenschaftliches Wissen auch durch ihre Integrations- und Simulationsfähigkeiten. Das gilt u.a., wenn komplexe Zusammenhänge in ungeheuren und ungeheuer komplexen Datensammlungen elektronisch fassbar gemacht werden. Dadurch werden neue Formen wissenschaftlicher Produktion möglich, die interpretiert werden müssen und neue Hypothesen- und Vorstellungswelten erschließen.

In den Neurowissenschaften und der Biomedizin sind die durch den Import theoretischer Modelle und instrumenteller Mittel der Informatik bedingten epistemischen Verschiebungen besonders deutlich sichtbar. Hier können die mit der Ablösung strukturalistischer durch postmoderne Episteme einhergehenden Verschiebungen, vom Text zum Bild, von geschlossenen zu offenen Systemen, von expliziter Repräsentation mit rationaler Rekonstruktion zur Simulation, sehr deutlich beobachtet werden.

In beiden Funktionen, der Problemlösung und der Nutzung als Repräsentationsmedium, wird der Computer hier gebraucht und die Informatik gefordert. In der Biomedizin kommt insbesondere der Repräsentationsaspekt mit beiden Polen, einerseits der Form von Abstraktion und Formalisierung und andererseits der gerade gegenläufigen Visualisierung abstrakter komplexer Datenmengen, zur Anwendung. Analog finden sich geschlossene und offene Problemlösungen bis hin zur Simulation. Diese Paradigmen gehen hier fließend ineinander über, von der Produktion, Strukturierung und Prozessierung höchst komplexer Datenmengen, deren Sammlung in Datenbanken und Hypertexten,

¹ Aus der Tatsache, daß derzeit über 80% aller Arbeitsplätze computerunterstützt sind und diese auch zunehmend vernetzt werden, läßt sich die Dringlichkeit weiblicher Kompetenzbildung und alternativer Einflussnahmen auf die Gestaltung ablesen.

der Produktion von Wissen durch Ableitung und Induktion (Data Mining), aus statistischen und komplex strukturierten Daten beim warping, bis hin zu Simulationen als Teillösungen der Bildgenerierung, wenn mathematisch geschlossene Lösungen oder deren numerische Behandlungsmethoden noch nicht existieren (Bsp.: Inverses Streuproblem zur Rekonstruktion des Streukörpers aus Streudaten aus CT etc.). Der Rückschluss von den Visualisierungen auf physiologische Realität läuft – epistemologisch problematisch – nur mehr über lange Interpretationswege der komplizierten Berechnungen. Es handelt sich um komplizierte Konstrukte aus kontingent kombinierten Methoden und Techniken, die auch immer die Möglichkeit der Herstellung von (medizinischen) Artefakten in sich tragen, die den physiologischen Tatbeständen im Inneren menschlicher Körper nicht entsprechen.

Die Wirkungen auf naturwissenschaftlich-medizinisches Wissen, auf Repräsentation und Wissensordnung und damit von Wissen selbst in diesen Wissenschaften, sowie die Wechselwirkungen biologischer und informatischer Modelle in Bioinformatik und Lebenswissenschaften beeinflussen insbesondere auch die Repräsentationen von Geschlecht in IT-Darstellungen und Visualisierungen. In der Bioinformatik und der Artificial Life-Forschung werden zudem Begriffe von Natur und Leben technisch verhandelt. Die Umdeutung zentraler Begriffe, wie Leben, Maschine, Organismus, Intelligenz, ermöglicht ja die Überführung der Biologie in synthetische Medien, wie von feministischen Naturwissenschaftlerinnen (Kollek, Weber) angemerkt.

3. Problembereich Kontingenz

Durch die mit informatischen Mitteln ermöglichten Verbindungsleistungen hat sich die Wissensproduktion in den Neurowissenschaften in ungeahnter Weise vervielfacht, aber gleichzeitig sind auch vielfältige Problemfelder entstanden. Eines davon möchte ich unter dem Begriff *Kontingenz* zusammenfassen.

– Kontingente Softwareentwicklung

Keine andere Technik ist so offen gestaltbar wie Software. Das gilt für den Modellierungsprozess ebenso wie für die Datenformate, die Algorithmenwahl und den erzeugten Code. Das macht aber auch klar, wie groß die Definitionsmacht nicht nur der Profession, sondern auch der einzelnen InformatikerInnen ist.

– Kontingente Wissensproduktion

Kontingent sind etwa die komplexen Kombinationen von physikalischen, mathematischen und informatisch-technischen Methoden in der medizinischen Bildgebung, die jeweils durch die sinngebenden Gewichtungen und Orientierungen einzelner ForscherInnen oder Forschungsgruppen entstanden sind. Die Verfahren können insgesamt in ihren komplexen Einzelheiten und deren komplexem Zusammenwirken nicht durchschaut werden. Werden die Methoden auch in ihren Charakteristiken und Einschränkungen von die Technik als black box verwendenden MedizinerInnen nicht durchschaut, so erzeugt dies Probleme und Fehler bei der Interpretation der Ergebnisse.

– Determinierende Repräsentation kontingenter Verhältnisse

Ein weiteres Problemfeld liegt in der adäquaten Darstellung von Kontingenzen, von Widersprüchen, Unterschieden und Abweichungen. Es besteht die Gefahr, die objektivierenden und normierenden Möglichkeiten, die die Technik über komplexe Transformationsleistungen bereit hält, zur Standardisierung auch tatsächlich zu nutzen, auch dort, wo dies inadäquat ist. Dabei können den Standardisierungs-Ergebnissen neue Sinngebungen unterlegt werden, die u.U. "biased" Konstruktionen sind, die für die Differenzierung etwa von Gesundheit oder Normalität Vorurteile gegen Rasse, Klasse, Geschlecht oder Alter enthalten können.

Daher sollte die Rückwirkung informatischer Modellbildung auf die Anwendungsdisziplin in den Blick genommen werden. Die formale ebenso wie die computergraphische Repräsentation strukturieren und modellieren Wissen in spezifischer Art. Formalisiertes Wissen muss auf spezifische Weisen (aus Gründen der Komplexitätsreduktion meist hierarchisch) geordnet und, weil man nicht alles explizieren und formalisieren kann, dekontextualisiert werden. Kontingentes Wissen kann dadurch leicht unzulässig generalisiert und in Eindeutigkeit sowie inadäquate Normierung gezwungen werden. Die gleiche Gefahr besteht bei graphischer Repräsentation: das menschlicher Kognition unmittelbar eingängige Bild kann standardisierende Wirkung haben, wird dieser Eindruck nicht weiter relativiert. Schmitz [01] und Masannek [01] haben Solches anhand der Wissensrepräsentation komplexer Inhalte und Bedeutungen für neurobiologisches Wissen in Hirnatlanten und Datenbanken demonstriert, bezogen auf die Darstellung von Geschlechterdifferenzen im Gehirn.

4. Kritik an Visualisierungen

Im Kontext der Biomedizin werden Normierungen häufig durch Bilder und Visualisierungen gegeben oder nahegelegt, verstärkt wenn sie über Mittelungen, statistische Ableitungen und folgende Standardisierungen gesetzt werden. Die unmittelbare kognitive Zugänglichkeit von Bildern wird mit der geringeren rationalen analytischen Zugreifbarkeit bezahlt: Die Bilder haften im Bewußtsein und leiten künftige Vorstellungen, wie etwas aussehen solle oder könne, sie werden Stereotype – Visiotype, wie Uwe Pörksen die Objektivität beanspruchenden Mathemate, Tabellen, Kurven oder Diagramme nennt.

Für die Neurowissenschaften bieten die modernen computertomographischen Verfahren² den "Blick ins Gehirn". Eine solche "Visualisierung des Lebendigen" ist das Ergebnis einer Kombination physikalischer Effekte, statistischer Verfahren, informatischer Modellierung und Verarbeitung und Visualisierungstechniken. Doch das höchst künstliche Bild verschleiert diesen Konstruktionsprozess und damit den Kunstcharakter.

Im Human Brain Project etwa werden überdies Bild-Datenbanken zur Standardisierung und Vergleichbarkeit der enorm variablen und methodenabhängigen Datensätze, insbesondere des Brain-Imaging hergestellt. Diese Kartographierungen und die mit ihnen verbundenen Standardisierungen tragen der Differenzierung des menschlichen Gehirns nicht genügend Rechnung und können so letztlich zu fehlerhaften wissenschaftlichen Ergebnissen führen. Dies ist umso problematischer, als sich an solche Definitionen von "Normalität" auch Definitionen von Normalität und Abnormalität, Geschlecht, Gesundheit und Krankheit, sowie Therapiebedürftigkeit knüpfen.

Gegen die zunehmenden Kartographierungen und Verbildlichungen des Körpers im Kontext der Biomedizin richtet sich auch feministische Kritik [Haraway 96]. Die Zunahme bildlicher Repräsentationen gegenüber textuellen ist ja ein wichtiges Element epistemischer Veränderungen in allen Wissenschaften, besonders aber in der Biomedizin.³ Flusser, dem es letztlich um Hypothesen zur Umgestaltung unseres Denkens durch den Computer und die Informatik geht, gibt Beschreibungen, die für die Interpretation von Bildern raumzeitliche und konnotative Aspekte heranziehen.⁴ In Ablösung von Rorty's "linguistic turn" sieht Mitchell im "pictorial turn" ebenfalls ein Zurückholen holistischer Mittel in Abwendung vom Semiotischen.⁵

Nach Fiala [97] synthetisieren die Informationstechnologien den abstrakten Begriff mit dem Bild, wie in unserem Kontext die abstrakten Streudaten mit deren Visualisierung: "Die computertechnische Simulation generiert eine artifizielle Naturalität, eine künstliche Natürlichkeit." Die Abstraktionen mutieren zu konkreten, sinnlich wahrnehmbaren Formen der "Wahrnehmungsgegenstände". "Zahlen lassen sich zu Bildern komputieren", wie Flusser es ausdrückt. Die "Substitution des Natürlichen" geschieht hier überdies auf der bio-technischen Ebene, nicht nur auf der symbolischen. Ein Grundzug dieses wissenschaftlichen Erkenntnismodells ist die Tendenz zur Konstruktion, zum Klon der Natur, deren Weg gerade über die Abstraktion verläuft.

Walter Benjamin [63] beschäftigt sich zwar mit Film und Fotografie, doch können seine Unterscheidungen auch auf die bildgebenden Verfahren übertragen werden: das Eindringen des Kameramanns in die Gegebenheit, die Zerstückelung der Bildkomponenten, die Maskierung und Denaturierung einer tieferliegenden Realität oder gar die Maskierung in Abwesenheit einer tieferliegenden Realität, bis hin zum Verweis auf keine Realität, sondern nur mehr auf sein eigenes Simulakrum. Baudrillard greift die Unterscheidung von Simulation und Simulakrum auf, wonach eine Simulation sich noch immer auf eine Art Urbild, ein Original, bezieht, während ein Simulakrum in einem ununterbrochenen Kreislauf ohne Referenz nur mehr auf sich selbst verweist, nicht auf etwas Reales außerhalb ihrer selbst. Wesentlich Visualisierungen aus den bildgebenden Verfahren der Biomedizin haben einen zu komplizierten epistemologischen Charakter, um klar in diese Unterscheidung zu

² Magnetresonanztomographie MRT, funktionelle MRT, Positronenemissionstomographie PET, Single Photon Emission Computed Tomography (SPECT), MEG, etc.; siehe [Hennig 01].

³ Medizinische Bilderzeugungsverfahren verursachen auf physikalischer Ebene eine Kausalkette bis zur Erzeugung von Streudaten und Daten aus durchdringenden Strahlen. Doch die weitere Prozessierung der Daten bis zur Visualisierung folgt weniger kausalen als pragmatischen Erfordernissen. Auf Physiologisches wird auf kompliziertem Weg rückgeschlossen. Was aber auf physiologischer Ebene gemessen wird, ist kausal nicht eruierbar, bestenfalls empirisch.

⁴ Er sieht in Bildern bedeutende Flächen, deren Bedeutung auf der Oberfläche liegt und sich mit nur einem Blick erfassen läßt. „Schweift der Blick über das Bild („Scanning“), um seine Bedeutung zu vertiefen, setzt sich die gewonnene Bildbedeutung aus der Bildstruktur und der Betrachterintention zusammen. Bilder sind nicht - wie z.B. Zahlen - denotative Symbolkomplexe, sondern konnotative: d.h. sie bieten Raum für Interpretationen. Während der Blick über die Bildfläche schweift, erfährt er ein Bildelement nach dem anderen und stellt bedeutungsvolle Beziehungen zwischen diesen her. In dem so entstehendem Bedeutungskomplex verleihen sich die Elemente wechselseitig Bedeutung. Die „vertiefte“, „gelesene“ Bedeutung des Bildes ist raum-zeitlich strukturiert, jedoch anders als die Bedeutung von linearen Texten.“ [Flusser 83, Flusser 92]

⁵ Wiederentdeckung des Bildes als komplexes Wechselspiel von Visualität, Apparat, Institutionen, Diskurs, Körpern und Figurativität. [Mitchell 97, S. 15-40]

passen, sie fungieren sowohl als Simulakra, mit denen interaktiv gearbeitet wird, als auch als bildliche Simulationen, die mit der Realität der Körper in bestimmte Übereinstimmungen gebracht werden müssen, um erfolgreich diagnostizieren oder (Roboter gestützt) operieren zu können.

5. Körperwahrnehmung

Die *Sicht auf den Körper* hat sich durch die Möglichkeiten der "optisch" in den Körper eindringen medizininformatischen bildgebenden Verfahren stark verändert. Er erscheint transparent, direkt, auch online einsichtig und präzise zugänglich. Die Universalität der informatischen Repräsentation macht Verbildlichung und Eingriff nahezu austauschbar: Wo in Voxel repräsentiert wird, kann nach dem elektronischen Skalpell, das nur das Bild bearbeitet, auch das Roboterskalpell im Körper geführt werden und schneiden, und die Wirkungen, also Eindringen, Verschiebung der Gewebe und Schnitte, sind direkt elektronisch visualisierbar. Die technische Verbindung der Visualisierung des Körperinneren mit der Roboterchirurgie ist nahtlos, da der Code der Visualisierung elektronische und physiologische Medien gewissermassen austauschbar und kombinierbar macht. Das Computerbild koppelt verschiedene Medien, den Tomographen, den Rechner, den Roboterarm und den menschlichen Körper, denn in den informationstragenden Bildeinheiten haben nicht nur bildliche (örtliche, farbliche und Intensitäts-Informationen), sondern noch viele andere Informationen Platz. Es ist ein Charakteristikum der universalistischen digitalen Medien, daß sie alle Medien zu verbinden erlauben und austauschbar machen. Die Interaktivität, die Grundvoraussetzung der robotergestützten Chirurgie ist, ist eine konsequente Folge dieser Verbindung von Realitätsebenen.⁶

Kritisch gewendet sind Austauschbarkeit der Materialien, Denaturalisierung und Naturalisierung Elemente des postmodernen Programms der Technowissenschaften, mit ihrer konstruktivistischen Konzeption alles Seienden als bestehend aus kontingenten und ineinander überführbaren Komponenten [Weber 99, S. 471]. Der Unterschied zwischen Natur und Kultur ist eliminiert. Das Simulakrum wird in Natur materialisiert: "it's `nature of no nature`". Feministische Forscherinnen sehen darin die Fortsetzung der maskulinistischen Träume von Unsterblichkeit und Omnipotenz, der Möglichkeit von Transzendenz und Manipulierbarkeit des Körperlichen, des "masculine birth of life" bis hin zu Phantasmen "lebendiger" autonomer Technofakte [Weber 99]. Diese Prozesse sind faktisch längst eingeleitet: Eine neue Verdinglichung des Körpers und seiner Komponenten zu entindividualisierter Biomasse, zu nachwachsendem Rohstoff, eine freie Kombinatorik der Körperstoffe macht sie unmittelbar kapitalisierbar [Gehring 01]. Die durch bildgebende Verfahren eingeleitete neue Durchlässigkeit der Körpergrenzen lässt die Elemente von Leben letztlich zu Objekten von Markt und Handel werden.

6. Konstruktive Nutzung der Genderforschung

Geschlechterforschung Informatik verhilft uns auf konstruktiver Ebene zu einem *allgemeinen Gestaltungsprinzip für die Benutzung von Softwaresystemen*, das auf Gleichberechtigung zielt:

Die Forderung nach *Schaffung eines offenen Raums* in Software-Systemen *für den Entwurf durch BenutzerInnen* kommt aus der feministischen Theoriebildung [Crutzen 00] (in Verallgemeinerung der von Peter Wegner [97] gegen geschlossene Systeme geforderten Interaktivität). Cecile Crutzen kritisiert, dass durch im Entwurf festgelegte Entscheidungen, wie ein Handeln interpretiert und welches Handeln formal repräsentiert wird, die Interpretationsvarianz der BenutzerInnen und damit sowohl deren Subjektivität als auch die der DesignerInnen vernachlässigt wird. Erst eine Öffnung von Software-Produkten zur Mitgestaltung und zum Mit-Entwerfen auf NutzerInnenseite kann umgekehrt neue Diskursräume schaffen. Das kann sich etwa ausdrücken in der Formbarkeit der Interaktion zwischen Mensch und Maschine, in der Schaffung eines offenen Diskursraums für die Wissensorganisation im Internet, in offener Navigationsfähigkeit in Informationssystemen, in inkrementeller Entwicklung, die Veränderbarkeit und Anpassbarkeit von Software gestattet, oder in offenem, unterschiedliche kognitive Stile zulassendem didaktischen Design für die Lehre mit Neuen Medien. Positive Effekte eines solchen Ansatzes kommen der Informatik selbst und der Nutzbarkeit ihrer Artefakte zugute: Starre, unflexible Benutzungsschnittstellen und Funktionalitäten, die sich unterschiedlichen Arbeitsstrukturen und -kontexten, Vorkenntnissen und Nutzungsgewohnheiten nicht anpassen können, sind ein, wenn nicht

⁶ Bernhard Vieh [89] schreibt, „die computergenerierten Bilder sind nicht mehr rein visuell, sondern vervollständigt bis zu einem Punkt, an dem der Unterschied zwischen Realität und Bild verschwindet. Das binäre Punktbild unterscheidet sich auch darin von seinen analogen Vorläufern, daß es ein interaktives Bild ist.“(S. 280) Das Bild bezieht die Betrachterin, den Betrachter mit ein. So kann es z.B. auf sie, ihn reagieren. ZuschauerInnen werden zu BenutzerInnen. „Es schwimmen zwei Realitätsebenen: die des Bildes und die des Benutzers.“ (S. 282) Und weiter: „Mit der Kompletierung (*gemeint ist: Universalität der Repräsentation; eigener Kommentar*) der Bilder wird ihr realitätsstiftender Charakter verstärkt.“ (S. 279)

das Haupthindernis für den wirtschaftlichen Erfolg von Software. Daher muss ein allgemeines Prinzip für informatische Modellbildung die Flexibilität der Benutzung sein.

Die dekonstruktivistischen Ansätze der Gender-Forschung können so einen kritischen Beitrag zur Sichtbarmachung und Auflösung starrer Dichotomien sowie konstruktiver Umsetzung leisten.

7. Behandlung von Kontingenz als künftige Aufgabe der Informatik

Die problematischen Wirkungen informationstechnischer Visualisierungen in der Biomedizin sind auch epistemologische Fragen der Informatik und der Informatisierung anderer Wissenschaften. Wenn die Informatik durch ihre Datenintegrationsleistungen, Mittelwertmodelle, Visualisierungen und Kartographierungen inadäquate Normierungen ermöglicht, so sollte sie auch ihre Aufgabe darin suchen, Unterschiede und Kontingenzen darzustellen. Solche Aufgaben sind jeweils nur problemabhängig zu bewältigen. Dennoch kann möglicherweise eine allgemeine Leitvorstellung helfen. Cecile Crutzens Modellierungskonzept zur Behandlung von Benutzungsunterschieden, das Offenheit der Gestaltung und Benutzung sichern soll, betrifft die Benutzungsschnittstellen. Es ist jedoch viel allgemeiner notwendig, Konzepte zur Darstellung von Kontingenzen und Unterschieden zu entwickeln, um Gegenmodelle gegen unerwünschte Normierungen und Übergeneralisierungen zu erhalten. Es geht zunächst darum, neue Dimensionen einzuführen, um Unterschiede auffächern zu können. Der Zeitverlauf und seine Dynamik ist eine davon. Aus der Genderforschung kommen weitere epistemologische Dimensionen hinzu, wie implizite Hintergrundannahmen, Erkenntnisziele, soziokulturelle Unterschiede, aber auch die Forschungsprozesse kanalisierende Kontexte, wie verfügbare Ressourcen, insbesondere Technik, die die kontingente oder widersprüchliche Datenproduktion erklären. Für manche dieser Variablen wird es möglich sein, sie im Datensatz mitzuführen, womit sich manche widersprüchlichen Ergebnisse auflösen lassen. Mit der Auffächerung werden allerdings Darstellungsprobleme eingehandelt, wie gerade von der ungeheuer vielparametrischen Datenproduktion aus bildgebenden Verfahren in der Medizin bekannt ist: Sie lassen sich nicht alle gleichzeitig – kognitiv aufnehmbar – visualisieren. Visualisierungstechniken könnten hier klug erweitert werden und sich überdies von der Kunst Inspirationen holen. Beispielsweise können Zeichnungen, Sprechblasen und Animationen, die aufbrechen, kommentieren, persiflieren, auf Defizite hinweisen, auch neben die technischen Bilder gesetzt werden, um diesen ihren allgemeinen Geltungsanspruch und Objektivitätsmythos zu nehmen.

Literatur

Baudrillard, Jean: "Agonie des Realen", Berlin 1978.

Benjamin, Walter: Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit. In: Ders.: Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit. Drei Studien zur Kunstsoziologie. Frankfurt a.M. 1963, S. 7-44.

Crutzen, Cecile: Interactie, een wereld van verschillen. Een visie op informatica vanuit genderstudies. Dissertation, Open Universiteit Nederlande, Heerlen 2000.

Crutzen, Cecile, Gerrisen, J.F.: Doubting the Object World. In: Balka, E.; Smith, R. (Eds.): Women, Work and Computerization. Charting a Course to the Future. Dordrecht, Boston: Kluwer Academic Pul. 2000.

Fiala, Erwin: Symbolische Welten und Abstraktionen. In: Ders.; List, E. (Hrsg.): Leib Maschine Bild. Wien 1997, S. 139-155.

Flusser, Vilém: Für eine Philosophie der Fotografie: Göttingen 1983.

–: Die Schrift. Hat Schreiben Zukunft? Frankfurt a.M. 1992.

Gehring, Petra: Neue Ökonomien: die Zirkulation von Körperstoffen, die Zirkulation von Körperdaten. <http://www.uni-freiburg.de/zag/zag/sites/gehring.htm>

Haraway, Donna: Anspruchloser Zeuge1@ Zweites Jahrtausend. FrauMann trifft OncoMouseTM. Leviathan und die vier Jots: *Die Tatsachen verdrehen*. In: Scheich, Elvira (Hrsg.): Vermittelte Weiblichkeit. Feministische Wissenschafts- und Gesellschaftstheorie. Hamburg 1996, S. 347-389.

Hennig, J.: Chancen und Probleme bildgebender Verfahren für die Neurologie. In Schinzel 01.

Masanneck, C: Das Human Brain Project – Hirnforschung im 21. Jahrhundert. In Schinzel 01.

Mitchell, W.J.T.: Der Pictorial Turn. In: Kravagna, Christian (Hrsg.): Privileg Blick. Kritik der visuellen Kultur. Berlin 1997, S. 15-40.

Schmitz, Sigrid: Informationssysteme zu neurobiologischem Wissen – Chancen und Grenzen. In Schinzel 01.

Schinzel, Britta (Hrsg.): Interdisziplinäre Informatik: Neue Möglichkeiten und Probleme für die Darstellung und Integration komplexer Strukturen in verschiedenen Feldern der Neurologie. Freiburger Universitätsblätter, 2001, Rombach, Freiburg.

- Vief, Bernhard: Vom Bild zum Bit. Das technische Auge und sein Körper. In: Kamper, Dietmar; Wulf, Christoph (Hrsg.): Transfigurationen des Körpers. Spuren der Gewalt in der Geschichte. Berlin 1989, S. 265-292]
- Weber, Jutta: Contested Meanings. Nature in the Age of Technoscience. In: Mittelstraß, Jürgen (Hrsg.): Die Zukunft des Wissens. Konstanz 1999, S. 466-473.
- Wegner, Peter: Why Interaction is More Powerful than Algorithms. In CACM, Vol. 40, No.5, 1997, pp. 81-91.

Klassifizierungen des Lebendigen

Über Ein- und Ausschließungen am Beispiel der Robotik und Artificial-Life-Forschung¹

Jutta Weber, Technische Universität Braunschweig

Kleine Vorbemerkung am Rande

Angesichts der aktuellen Entwicklungen bzgl. des Hochschulrahmengesetzes fiel es mir beim Formulieren dieses Positionspapieres schwer, wie geplant auf die Aus- und Einschließungen in der AL-Forschung zu fokussieren. Angesichts der rigiden Verfestigung althergebrachter Familien-, Lebens- und Karrierekonzepte durch das neue Hochschulrahmengesetz der rot-grünen Regierung war mein erster Impuls, die alte Debatte über die Ausgrenzung von Frauen (und anderer Menschen mit nicht-normierten Lebensläufen) und der Notwendigkeit von Anti-Diskriminierungspolitik wieder aufzunehmen.

Angesichts der Tatsache, daß wir uns in Bad Hersfeld mit der *Theorie der Informatik* auseinandersetzen, werde ich mich aber im Folgenden auf die Analyse weniger auffälliger, aber nicht weniger wirksamer Ausschließungsmechanismen beschränken, wie sie in der Bedeutungs- und Technologieproduktion der Informatik wirken. Wenn es mir auch Sorge bereitet, wer überhaupt noch in naher Zukunft diese Kritik professionell wird betreiben können.

Zum Thema

Im folgenden geht es mir um Ein- und Ausschließungen in symbolischen Ordnungen, theoretischen Konzeptionen, technischen Umgebungen und Artefakten, die schwer zu sehen und vor allem sichtbar zu machen sind, die aber auch zentral unseren Alltag und unser Leben bestimmen. Es geht damit um die angesprochene Frage, "welche 'scheinbaren' Gewissheiten - Kategorien, Denkweisen, Anschauungen – in informationstechnische Systeme (und Konzepte; JW) eingeschrieben werden und durch welche Prozesse diese materiell-symbolischen Verwicklungen des Technischen entstehen." (Bath, Aufruf zur AG)

Feministische Naturwissenschaftsforschung hat – zusammen mit anderen kritischen Strömungen wie Postkolonialismus, queer theory, cultural studies of science, etc. – Mechanismen der Schließung in Informatik bzw. Cyberscience immer wieder herausgearbeitet.² So schreibt Susan Leigh Star in ihrem - mit Geoff Bowker zusammen verfassten - Buch *Sorting Things Out*: "Information scientists work every day on the design, delegation and choice of classification systems and standards, yet few see them as *artefacts embodying moral and aesthetic choices that craft people's identities, aspirations, and dignity.*" (Bowker/Star 2000, 4; kursiv von J.W.). Vom mainstream der Wissenschaft werden diese kritischen Arbeiten bis heute nicht wahrgenommen – vermutlich würde das die eigene Arbeit, das business-as-usual, nachhaltig stören.

Im folgenden werde ich mich auf Ein- und Ausschließungen bzw. Klassifizierungen und Standardisierungen in den diskursiven und praktischen Verhandlungen über 'Leben' im Bereich der Robotik und Artificial Life-Forschung konzentrieren. Dabei verlaufen diese Klassifizierungen sicherlich *nicht ausschließlic*h entlang der traditionellen Männlich-Weiblich-Stereotypen, leider stellen diese aber weiterhin wesentliches Moment in den Markierungs- und Auswahlprozessen der AL- und Robotikforschung dar.

Artificial Life und (evolutionäre, situierte) Robotik

'Artificial-Life'-Forschung arbeitet an der Schnittstelle von Bio- und Informationswissenschaften und zielt auf die Konstruktion bzw. Simulation des Lebendigen.³ Sie – und die mit ihren Ansätzen arbeitende Robotik – ist darauf gerichtet, "Lebendiges jenseits des organischen Substrats zu konstruieren" (Becker 2000, 41) und auf mechanische Systeme, z.B. auf Siliziumbasis, zu übertragen. Während Gentechnologie und Biowissenschaften eher daran interessiert sind, neues Leben in Form von kohlenstoffbasierten Organismen und organischen Substanzen jenseits der 'natürlichen' Evolution zu produzieren, gehen die Forschungsbereiche Robotik und Artificial Life (kurz Alife oder AL) den umgekehrten Weg: Hier geht es um "[d]ie Konstruktion von intelligenten Artefakten ... durch Simulation

¹ Dieser Beitrag beruht in einigen Teilen auf meiner Dissertation Weber 2001/03.

² Vgl. u.a. Bath 2000, Hayles 1999, Schelhowe 1997, Schinzel 2001, Schmitz 2002, Weber 2001, 2002, 2003.

³ Vgl. Hayles 1999, 225.

lebendiger Prozesse auf der Basis *nicht-organischer* Substrate. Dabei stehen vor allem Versuche im Vordergrund, entsprechende Systeme zu autonomem Verhalten in komplexen Umwelten zu befähigen." (Becker 2000, 44; Hervorhebung von J.W.) Leben soll in der Hardware (Roboter) und Software erweckt werden.

Theoretische Grundlagen

Allgemeines Ziel der Artificial-Life-Forschung sind ihrem 'Gründungsvater' Christopher Langton zufolge, die Erkundung der "essential properties of life – properties that must be shared by any living system *in principle*" (Langton 1996, 39). Die Idee, allgemeine Prinzipien zu destillieren und in künstliche Medien zu implantieren, erscheint weniger wunderbar, wird man gewahr, daß heute Leben sowohl in der Cyberscience als auch in den neueren Biowissenschaften häufig als ein allgemeines Prinzip verstanden wird, welches sich primär in Selbstorganisation und genetischem Programm manifestiert und in seiner Logik weitgehend unabhängig von der Materie ist.⁴ Auf dieser Grundlage hält es die AL-Forschung für möglich, nicht nur maschinelle Intelligenz, sondern auch Leben in silicio zu schaffen oder wenigstens zu simulieren. Der bekannte Robotikforscher und Unternehmer Rodney Brooks am Massachusetts Institute of Technology in Boston drückt das folgendermaßen aus: "Mein Wunsch ist es, vollkommen autonome bewegliche Agenten herzustellen, die mit den Menschen in der Welt koexistieren und die von diesen Menschen als gleichberechtigte, intelligente Wesen angesehen werden. Ich würde solche Agenten *Kreaturen* nennen." (Brooks nach Levy 1996, 335; Hervorhebung von mir)

Der transdisziplinäre Forschungsbereich Artificial Life will – anders als die KI-Forschung – die basaleren Prozesse auf der Ebene des Lebens nachschöpfen, die nun als Voraussetzung für die Ausbildung von smarten, umweltorientierten und flexiblen Artefakten gelten. Dabei scheidet sich die AL-Forschung in zwei Strömungen: Die starke Variante der AL-Forschung zielt direkt auf die Nachschöpfung des Biologischen und gibt sich nicht – wie der schwache Ansatz der AL-Forschung – mit einer Simulation bzw. Analyse der 'Lebensprinzipien' zufrieden, sondern er motiviert sich gerade aus der Schöpfung von *neuem, postbiologischem Leben*. Christopher Langton spricht dies klar und deutlich aus: "Note that we expect the synthetic approach to lead us not only to, but quite often *beyond*, known biological phenomena: beyond *life-as-we-know-it* into the realm of *life-as-it-could-be*." (Langton 1996, 40)

Materie und Form bzw. Selbstorganisation

Die entscheidenden Kriterien für Leben sind der AL-Forschung zufolge Information sowie Emergenz. Nach Christopher Langton, einem der bekanntesten AL-Forscher, manifestiert sich Leben primär in der (In)Formierung von Materie, in ihrer Selbstorganisation und deren emergenten Effekten: "Life is a property of form, not of matter, a result of the organization of matter rather than something that inheres in the matter itself. Neither nucleotides nor amino acids nor any other carbon-chain molecule is alive – yet put them together in the right way, and the dynamic behaviour that emerges out of their interactions is what we call life" (Langton 1996, 53). Innerhalb dieser Logik ist die Überführung von Organismen bzw. lebenden Systemen in Maschinen (und vice versa) sowie die technische Nachkonstruktion von Leben möglich. Die Produktion von neuen Organismen scheint nur noch eine Frage der Umwandlung der (richtigen) Information durch kompetente Wissenschaftler zu sein.

Um aber die universale Anwendbarkeit der Muster der Selbstorganisation auf die ganze lebendige Natur garantieren zu können, muß die materiale Verfaßtheit der jeweiligen Organismen und Entitäten von marginaler Bedeutung sein. Die Muster der Selbstorganisation, die – in unterschiedlich komplexer Form – die komplette Natur durchziehen, sind das bestimmende Element. Diese informationstheoretisch formulierten Muster liefern die Gesetze, mit deren Hilfe man die lebendige wie tote Natur in ihrer Logik begreifen, mathematisch-physikalisch und biochemisch rekonstruieren, produzieren, weiterentwickeln und zu autoevolutionären Sprüngen veranlassen kann. Dabei wird nicht bestritten, daß diese Operationalisierung einer materialen Grundlage bedarf, aber es wird davon ausgegangen, daß die jeweilige Beschaffenheit dieser materialen Grundlage für die Prozesse der Selbstorganisation, der Emergenz etc. irrelevant sind: "Die mathematischen Symmetrien von Elementarteilchen oder Molekülen sind zwar unabhängig vom jeweiligen stofflichen Träger definiert. Um aber physikalisch oder chemisch wirksam werden zu können, bedarf es eines jeweiligen materiellen Trägers. Die mathematische Struktur einer Information oder eines Computerprogramms ist zwar unabhängig von einem stofflichen Träger definiert. Um aber als Information oder Programm wirken zu können, bedarf es eines materiellen Trägers – sei es in den Genen einer Zelle, auf biochemischer Grundlage von Gehirnen oder mit den elektronischen Materialien herkömmlicher Computer." (Mainzer 1996, 87) Nur vage deutet der Wissenschaftsphilosoph Klaus Mainzer im Anschluß an, daß sich – wie

⁴ Vgl. Kay 1996; Keller 1995, 2002. Ich danke Evelyn Fox Keller für die Überlassung von Teilen ihres damals noch unveröffentlichten Buchmanuskripts.

Aristoteles schon wußte – "begriffliche Unterscheidungen wie `Stoff´ und `Form´ auf verschiedene Funktionen und Prinzipien der Natur [beziehen; J.W.], die aber voneinander abhängen und keine separierte Existenz besitzen." (Mainzer 1996, 87) Nichtsdestotrotz wird von der Materie nur als Träger gesprochen.

Diese Trennung von Stoff und Form, von Materie und Information reproduziert eine uralte Polarisierung und Hierarchisierung innerhalb westlichen Denkens. Ihr zufolge ist Stoff nichts als Material, das passiv ist und dem eine Form aufgeprägt wird. Die Form ist das Kreative, das entscheidende und aktive Moment. Zumeist war und ist dann ersteres als `weiblich´ und letzteres als `männlich´ gekennzeichnet. Diese Argumentation findet sich von Aristoteles´ Zeugungstheorie bis zur Entwicklungsbiologie und setzt sich auch in vielen kosmologischen Schöpfungs- und Weltordnungsvorstellungen innerhalb der Philosophie bis zum heutigen Tag fort.⁵

Das altvertraute Muster die Überlegenheit des Geistigen über das Körperliche, der Information über die Materie,⁶ wobei häufig zweiteres dem sogenannten Weiblichen zugeordnet wird⁷ – setzt sich in dieser neuen Hightech-Forschung fort. Auch sie ist – zumindest partiell – eine Wissenschaft, die sich als allmächtig imaginiert und voller Schöpfungs- und Omnipotenzphantasien steckt. Donna Haraway beschreibt diese Haltung mit Blick auf die Molekularbiologie und Genetik sehr treffend: "Nature is the program; we replicated it, we own it; we are it. Nature and culture implode into each other and disappear into the resulting black hole, Man™ makes himself in a cosmic act of onanism. The nineteenth-century transfer of God's creative role to natural processes, within a multiply stratified, ... industrial culture committed to relentless constructivism and productionism, bears fruit in a comprehensive biotechnological harvest in which control of the genome is control of the game of life itself – legally, mythically, and technically." (Haraway 1997, 149f)

Starker versus schwacher Ansatz. Oder: Nur ein paar Spinner?

Von der Analyse theoretischer Grundlagen lässt sich natürlich nicht eins zu eins auf die Beschaffenheit der realen Artefakte schließen. Die Wirkmächtigkeit und Umsetzung dieser theoretischen, erkenntnistheoretischen wie ontologischen Grundannahmen des starken Ansatzes der AL-Forschung muss hier weiterverfolgt werden.

In meiner Analyse des Lebensbegriffs in der AL-Forschung habe ich mich zudem primär auf den starken Ansatz der AL-Forschung bezogen, insofern in ihm die theoretischen Prämissen dieser Wissenschaft deutlicher zum Ausdruck kommen. Die ForscherInnen des starken Ansatzes werden oftmals von dem Rest der Informatik-Community belächelt oder gar als Spinner betrachtet. Doch kann ich mich nicht des Verdachts erwehren, daß sie die Funktion von Sündenböcken bekommen. Um den problematischen Entwicklungen und Reduktionismen in Robotik und AL-Forschung nicht ins Gesicht sehen zu müssen, redet man sich gerne darauf heraus, daß es sich hierbei nur um die wilden Ideen und schrillen Phantasmen einiger weniger handele. Fragt sich allerdings, warum sich entscheidende Momente wie die Reduktion von Lebensprinzipien auf Informationsstrukturen, eine naturalistische Konzeption von Emergenz und anderes mehr auch im schwachen Ansatz der AL-Forschung finden.

Was man allerdings dem schwachen Ansatz der AL-Forschung zugute halten muss, ist ein relativ ausgeprägtes Bewußtsein um die eigene Konstruktionsleistung. Lars Risan hat in seiner Laborstudie über die AL-Forschung an der School of Cognitive and Computing Sciences an der Universität Sussex in England darauf hingewiesen, daß die schwache Version der AL-Forschung sich nicht als objektivistische "real science" (Risan 1996, 41) präsentiert, sondern eher als "postmodern science" (Risan 1996, 41), als *eine Wissenschaft, die sich der Konstruktion neuer Maschinen bzw. Welten, aber nicht der Erforschung und objektiven Repräsentation der Natur verschrieben hat*. Nicht nur bezeichnen sich einige AL-ForscherInnen selbst als KünstlerInnen und ihre Simulationen als Kunst,⁸ die sie allerdings als technische orientierte Entwicklungsarbeit verkaufen müssen, um weiterhin Forschungsgelder zu erhalten: "We would get troubles with the funding agencies if we admitted openly that it was kind of art or self-expression we were doing, and nothing more. So we will have to claim that it is engineering we are doing. We are, and we will have to be, closet artists clothed as engineers." (Interview in: Risan 1996, 42) Einige sehen offensichtlich ihre ganze Forschung als konstruktivistisches und / oder künstlerisches Unterfangen.

⁵ Vgl. u.a. Klinger 1995, 43ff.

⁶ Vgl. Hayles 1999, 19.

⁷ Vgl. u.a. Keller 1992, Saupe 2002.

⁸ Vgl. Risan 1996, 42 sowie Risan 1996, Kapitel 6.

Forschungspraxis

Andererseits gibt es auch das Phänomen, daß RobotikforscherInnen, die einen starken Ansatz vertreten und davon ausgehen, biologische Konstruktionsleistungen auf natürliche Systeme zu übertragen, im Rahmen der rechnergestützten Ethologie "keinen Wert auf biologische Genauigkeit für ihre Kreaturen" (Levy 1996, 359) legen. Die Robotikforscherin Pattie Maes schätzt Tiere als 'suboptimal' ein,⁹ insofern diese durchaus Verhaltensweisen an den Tag legen, die "nicht als effizientes Roboterverhalten gelten konnten." (Levy 1996, 359) Ärgerlicherweise assimilierten sich wohl diese nicht-menschlichen und nicht-maschinellen AgentInnen nicht so gut an die durch die Maschinen geprägten Vorgaben bezüglich des eigenen Handelns wie ihre menschlichen Verwandten, deren Anpassungsleistungen die Übertragungsleistung von Mensch und Maschine häufig überhaupt erst überzeugend werden läßt.¹⁰ Dies gründet letztendlich auch darin, daß die Sponsoren der Robotik-Forschung eher an funktionstüchtigen Robotern als an der genauen Simulation von Tierverhalten interessiert sind und damit der Robotikforschung recht eindeutig den Weg weisen.

Spannende Frage für die Diskussion wäre, was die Spannung zwischen durchaus geschlechtsstereotyp codierten theoretischen Grundlagen, naturalistischer Rhetorik und realer, doch recht pragmatischer Technologieproduktion für eine Funktion und für Folgen hat.

Auf jeden Fall stimmt es mich nachdenklich, wenn im Rahmen des EU-Förderprogramms zu 'New and Emerging Technologies' im Feld der Neuroinformatik die Entwicklung von 'living artefacts' anvisiert wird "Beyond the Made and the Born". (Cordis)

Literatur

- Bath, Corinna (2000): 'The virus might infect you'. Bewegt sich das Geschlechter-Technik-Gefüge? In: metis. Zeitschrift für kritische Frauenforschung und feministische Praxis, 9. Jg., H. 17, S. 48-66
- Bammé, Arno/Feuerstein, Günter/Genth, Renate/Holling, Eggert/Kahle, Renate/Kempin, Peter (1983): Maschinen-Menschen, Mensch-Maschinen. Grundrisse einer sozialen Beziehung. Reinbek.
- Bowker, Geoffrey, Star, Susan Leigh (2000): Sorting Things Out. Classification and Its Consequences. Cambridge/London: MIT Press.
- Becker, Barbara (2000): Cyborgs, Robots und Transhumanisten. Anmerkungen über die Widerständigkeit eigener und fremder Materialität. In: dies./Irmela Schneider (Hrsg.): Was vom Körper übrig bleibt. Körperlichkeit – Identität – Medien. Frankfurt a.M./New. S. 41-70.
- Christaller, Thomas: Von Artificial Intelligence zu Artificial Life und wieder zurück. Künstliche Intelligenz 4, 1996, S. 41-44.
- Cordis (2000): Information Society Technologies: Fet – Proactive Initiative 2000. Neuroinformatics for living artefacts. In: www.cordis.lu/ist/fetni-4.htm, 1 (last access: 29.11.2000)
- Haraway, Donna J. (1997): Modest_Witness@Second_Millennium. FemaleMan©_Meets_Onco-Mouse™. Feminism and Technoscience. New York/London.
- Hayles, N. Katherine (1999): How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics. Chicago.
- Heintz, Bettina (1993): Die Herrschaft der Regel. Zur Grundlagengeschichte des Computers. Frankfurt a.M.
- Kay, Lily E. (1996): Life as Technology: Representing, Intervening, and Molecularizing. In: Sahotra Sarkar (ed.): The Philosophy and History of Molecular Biology: New Perspectives. Dordrecht/Boston/London. S. 87-100.
- Keller, Evelyn Fox (1992): Secrets of Life - Secrets of Death. Essays on Language, Gender and Science. New York/London.
- Keller, Evelyn Fox (1995): Refiguring Life. Metaphors of Twentieth-Century Biology. New York/Chichester.
- Keller, Evelyn Fox (2002): Making Sense of Life. Explaining Biological Development with Models, Metaphors and Machines. Cambridge, MA/London: Harvard University Press.
- Klinger, Cornelia: Beredtes Schweigen und verschwiegenes Sprechen (1995): Genus im Diskurs der Philosophie. In: Hadumod Bußmann/Renate Hof (ed.): Genus. Zur Geschlechterdifferenz in den Kulturwissenschaften. Stuttgart. S. 409-445.
- Langton, Christopher G. (1996): Artificial Life. In: Margaret A. Boden (ed.): The Philosophy of Artificial Life. New York. S. 39-94.
- Levy, Steven (1996): Künstliches Leben aus dem Computer. München. (im Orig. 1992)
- Maes, Pattie (1997): Modeling Adaptive Autonomous Agents. In: Christopher G. Langton (ed.): Artificial Life. An Overview. Cambridge/London. S.135-162.

⁹ Vgl. Maes in Levy 1996, 359.

¹⁰ Vgl. Heintz 1993; Bammé et al. 1983.

- Mainzer, Klaus (1993): Künstliche Intelligenz, Neuroinformatik und die Aufgabe der Philosophie. In: Gert Kaiser/Dirk Matejovski/Jutta Fedrowitz (Hrsg.): Kultur und Technik im 21. Jahrhundert. Frankfurt a.M./New York, S. 118-134.
- Mainzer, Klaus (1996): Materie: von der Urmaterie zum Leben. München.
- Risan, Lars (1996): Artificial Life: A Technoscience Leaving Modernity? www.uio.no/~Irisan/Thesis/heliopg.html (last access 2.5.2000)
- Saupe, Angelika (2002): Verlebendigung der Technik. Perspektiven im feministischen Technikdiskurs. Bielefeld: Kleine-Verlag.
- Schelhowe, Heidi (1997): Hat der Computer ein Geschlecht? Frauenforschung in der Informatik. In: Dorothea Mey (HG.): Frauenforschung als Herausforderung der traditionellen Wissenschaften? Göttingen. S.81-92.
- Schinzal, Britta (Hrsg.) (2001): Interdisziplinäre Informatik: Neue Möglichkeiten und Probleme für die Darstellung und Integration komplexer Strukturen in verschiedenen Feldern der Neurologie. Freiburger Universitätsblätter 153/3.
- Schmitz, Sigrid (2002): Informationstechnische Darstellung, kritische Reflexion und Dekonstruktion von Gender in der Hirnforschung - Das Projekt GERDA. Niedersächsischer Forschungsverbund für Frauen und Geschlechterforschung in Naturwissenschaft, Technik und Medizin. Dokumentation II. Tagungsbeiträge 1999-2001 (im Druck).
- Weber, Jutta (2001): Eine lebendige, komplexe Natur und ihre Maschinen: Über den Transfer von 'Lebensprinzipien' in der Artificial Life-Forschung. In (dies. 2001/3).
- Weber, Jutta (2002): Berechenbare Organismen, wildgewordene Maschinen? Heilsvisionen und Apocalyptica der Artificial-Life-Forschung. In: ZiF-Bulletin. Zentrum für Interdisziplinäre Frauenforschung Berlin Nr. 23, April 2002, 73-86.
- Weber, Jutta (2001/2003): Umkämpfte Bedeutungen: Natur im Zeitalter der Technoscience. Dissertationsschrift, Bremen, 2001.
In: http://elib.suub.uni-bremen.de/publications/dissertations/E-Diss228_webersec.pdf (erscheint leicht überarbeitet im Februar 2003; Frankfurt a.M./New York, Campus Verlag).

Kulturelle und Geschlechterperspektiven auf die Informatik

Bericht zur Arbeitsgruppe

Corinna Bath, Jutta Weber

Die Arbeitsgruppe ging dem Thema "Genderforschung und Informatik" nach. Es wurden Einschreibungen von Geschlecht im Zirkel der Technikentwicklung und -wirkung sowie die kulturell-politisch-soziale Wirksamkeit von Kategorien und Klassifizierungen in den informatischen Theorien, Konzepten und Produkten genauer in den Blick genommen. Dieser Aufgabe waren neun Teilnehmende mit diversen, durchaus spannenden Beiträgen nachgekommen. Analysiert wurden hier u.a. geschlechterdifferenzierende Annahmen und Konnotationen in Software/IT, in der Softwareentwicklung, in der Dichotomie von NutzerInnen und EntwicklerInnen und in Professionalisierungsprozessen. Während die Beiträge durchaus von Geschlecht sprechen, stellten erstaunlicherweise nur wenige explizit Bezüge zu einem feministischen Theorierahmen her. Analysen bezüglich der Annahmen über kulturelle, ethnische u.a. Differenzen in der Informatik wurden nicht unternommen. Rückgriffe auf Postcolonial Studies oder andere, nicht vornehmlich auf Geschlecht rekurrierende Ungleichheitskonzepte sind aber im allgemeinen bis heute eine Leerstelle in der Theorie der Informatik.

In der Arbeitsgruppe standen wissenschafts- und erkenntniskritische Ansätze sowie die Reflektion der gesellschaftlichen Wirkungen von Informatik, Software und Informationstechnologien im Mittelpunkt. Als Moderatorinnen legten wir von Anfang an Wert darauf, dass es nicht um Untersuchungen und Maßnahmen zur Frauenförderung in der Informatik geht. Vielfach wird jedoch eine Geschlechterperspektive nachwievor als Suche nach Unterschieden oder als "Frauenthema" interpretiert. Auch in unserer Arbeitsgruppe bestand dementsprechend zunächst ein Klärungsbedarf zum Verständnis von Geschlecht und von Gender Studien.

Dabei wurde deutlich, dass Geschlecht in Bezug auf eine Theorie der Informatik nur auf einer strukturell-symbolischen Ebene verhandelt werden kann. Es geht dabei also nicht um Frauen und Männer, um "weibliche" oder "männliche" Zugangs- und Umgangsweisen mit Technik oder um die Frage, warum es so wenige Frauen in der Informatik gibt. Vielmehr geraten hier Zeichen von Geschlechtlichkeit, kulturell-gesellschaftliche Zuschreibungen und Verknüpfungen mit Geschlecht, aber auch soziale und gesellschaftliche Geschlechterverhältnisse - wie sie sich bspw. in der gesellschaftlichen Arbeitsteilung niederschlagen oder im hierarchischen Gendering von "technischen" und "sozialen" Kompetenzen - in den Blick.

Gleichzeitig kam zum Ausdruck, dass sich feministische Wissenschafts- und Technikforschung bzw. Gender Studien zumeist als Wissenschaftskritik verstehen, indem sie sich kritisch mit dem Androzentrismus und Sexismus in Wissenschaft und Technik auseinandersetzen. Dabei geht es vielfach um die Frage, welche Themen verhandelt werden und welche nicht, um Kategorien, Klassifizierungen und Denkstrukturen, oder auch um Dichotomien, die in unserer abendländischen Kultur nur allzu oft geschlechtlich konnotiert sind, ohne dass sie notwendigerweise direkt auf Frauen und Männer in einem biologi(sti)schen Sinne verweisen.

Gender Studien leisten nicht nur auf der Ebene von Analysen einen produktiven Beitrag, z.B. wenn sie aufdecken, welche Klassifizierungen, 'Normalisierungen' und Standardisierungen durch die Artefakte der Informatik vorgenommen werden, die diese selbst nicht reflektiert. Dadurch, dass Gender Studien z. B. eine Situierung einfordern, können sie als Inspiration für eine kritische Wissenschaft und Praxis der Informatik dienen. Damit wird schon angedeutet, warum die (Theorie der) Informatik Genderforschung braucht.

Insgesamt möchten wir aus der anderthalbtägigen Diskussion hier vier Punkte herausgreifen, die uns als Quintessenzen bzw. auch als Dissenzen besonders wesentlich oder besonders umstritten erscheinen. Eine erste Kontroverse, welche die gesamte Arbeitsgruppe durchzog, betrifft

Einschreibungen von Gender in die Informatik und ihre Produkte

Während wir als Moderatorinnen und viele der Teilnehmenden einerseits vielfältige Beispiele identifizieren und zusammentragen konnten, die aufzeigen, auf welche Weise sich Geschlecht in informatische Konzepte, informationstechnische Systeme oder die Informatik-Fachkultur einschreiben, wurden andererseits diese konkreten Fälle immer wieder in Frage gestellt. Es wurde nachgefragt, inwieweit das jeweilige Beispiel wirklich für die Einschreibung von Geschlecht in informatische Artefakte stünde oder ob es nicht eher die gesellschaftlichen Umstände und Einbettung der Informatik beträfe.

Letztendlich schienen für den Dissens über diesen Punkt vor allem unterschiedliche Vorstellungen von "Einschreibung" und von "Geschlecht" verantwortlich. Möglicherweise zeigten sich darin auch ein unterschiedliches Verständnis von Informatik.

Die Beispiele, an denen wir geschlechtliche Einschreibungen diskutierten, bewegten sich auf verschiedenen Ebenen. Zunächst angeführt wurden Metaphern und Sprechweisen. Auffällig geschlechtsspezifisch formuliert sind etwa - trotz langjähriger Kritik - Aufgaben und Beispiele, mit Hilfe derer in der Informatik theoretische Konzepte erläutert und Übungen beschrieben werden. Dies betrifft sowohl Lehrbücher als auch wissenschaftliche Beiträge und Fachzeitschriften. Unter der Hand werden damit gesellschaftliche Zuschreibungen und Rollen reproduziert. Stereotypisierungen dieser Art ließen sich zumeist leicht beheben, wenn eine gewisse Sensibilität und ein Interesse bestünde, überkommene Muster aufzubrechen.

Ein großer Bereich geschlechtlicher Einschreibung umfasst die Verwendung von Metaphern, die – wie das Vater-Sohn-Prinzip in Bäumen verdeutlicht – in der Informatik weit verbreitet sind. Die Herausforderung besteht hier darin, wohletablierte Fachtermini der Informatik auf ihre direkten oder auch mitschwingenden Bedeutungen zu überprüfen und ggf. geschlechtsneutralere Begriffe zu verwenden. Denn sprachliche Bestimmungen hängen eng mit gesellschaftlich-kulturellen Zuschreibungen zusammen, die eine starke soziale Wirksamkeit entfalten.

Der vergleichsweise hohe Frauenanteil in dem relativ jungen Studienfach Medieninformatik etwa läßt sich auch darauf zurückführen, dass der Begriff "Medien" – unabhängig von den tatsächlichen Inhalten des Curriculums – weniger technisch und somit auch weniger männlich konnotiert erscheint als Informatik oder gar technische Informatik (vgl. Schinzel 2002a). Die sozialen Konstruktionen einzelner Zweige der Informatik, ihre Selbst- und Fremdbilder, ihre Rhetoriken und Praktiken sind unterschiedlich stark geschlechtlich geprägt. So konnte eine empirische Studie (Erb 1996) zeigen, dass den Informatikerinnen während der 80er Jahre die Theoretische Informatik als Zugangsschneise zum Fach diene. Stereotype Vorstellungen vom Hacker und Computerfreak waren dagegen kaum mit dem Selbstbild von Frauen vereinbar. Heutzutage scheinen Mädchen und junge Frauen ihre naturwissenschaftlich-technische Studienfachwahl weniger legitimieren zu müssen. Auch ihre Selbstwahrnehmung kann sich damit verändern. Dennoch wird mancherorts weiterhin versucht, mit traditionellen Klischees zu arbeiten und die Informatik dadurch für Frauen attraktiv zu machen, dass kommunikative Anteile des Faches gegenüber den technischen herausgestellt werden. Ein solcher Ansatz ist zwar in der Hinsicht berechtigt, dass vorurteilsbeladene Sichtweisen der Informatik korrigiert werden. Er läuft jedoch Gefahr, die Gleichsetzung von Technik mit Männlichkeit und von Kommunikation mit Weiblichkeit aufrechtzuerhalten und den geringen Frauenanteil der Informatik allein als Problem der Frauen zu sehen. Ein Aufbrechen zweigeschlechtlicher Schemata wird damit nicht erreicht.

Offensichtlicher noch wird die Zweigeschlechtlichkeit in der Informatik durch Dichotomien hergestellt. Eines der deutlichsten Beispiele dafür ist die Trennung von Nutzung und Entwicklung, von Designern und Usern, deren Pole strukturell-symbolisch stark geschlechtlich konnotiert sind. Traditionellen Auffassungen der Informatik zufolge gehören die Bereiche, die eher dem Weiblichen zugeordnet werden, also die Felder der Nutzung und der NutzerInnen, nicht zur Disziplin bzw. in abgeschwächter Position nicht zur sogenannten Kerninformatik. Auch die Unterscheidung in Kern und Peripherie der Informatik stellt eine Dichotomie dar, die aus einer Geschlechterperspektive hinterfragt werden kann und sollte. Denn gerade die nicht als mathematisch oder als ingenieurwissenschaftlich-technisch geltenden Bereiche werden auf diese Weise abgewertet und ausgegrenzt.

Weiteres Thema der Arbeitsgruppe waren soziotechnische Einschreibungen und das Gendering im Sinne der sozialwissenschaftlichen Technikforschung. Dabei geht es um die Frage, wie sich soziale und gesellschaftliche Aspekte, u.a. Vorannahmen über die Geschlechter, in IT-Systemen und Software widerspiegeln und vergegenständlichen. Insbesondere unreflektierte Vorstellungen und Denkmuster werden dabei relevant. Als Beispiele für diese Art Einschreibung von Geschlecht wurden die historischen Studien zur Entstehung von Textverarbeitungssystemen genannt (Hofmann 1999, Webster 1993), die aufzeigen, dass bestimmte geschlechtliche, nicht überprüfte Annahmen über die NutzerInnengruppe in die Entwicklungsprozesse eingingen.

Ein weiteres Beispiel sind Krankenhausinformationssysteme, über die Ina Wagner bereits Ende der 80er Jahre Studien durchgeführt und veröffentlicht hatte. Dabei geht es vor allem um die Sichtbarkeit, Anerkennung und organisatorischer Umstrukturierung von sog. Frauenarbeit, die im pflegerischen Dienst geleistet wird. Es wurde die These vertreten, dass Fehlentwicklungen in diesem Bereich, die heutzutage hervortreten, vermeidbar gewesen wären, wenn diese frühen feministischen Untersuchungen Gehör gefunden hätten und ihre Relevanz rechtzeitig erkannt worden wäre.

Auch aktuelle Beispiele geschlechtlicher Einschreibungen wurden in unserer Arbeitsgruppe diskutiert. Immer wieder zur Veranschaulichung herangezogen werden dazu Computerspiele, deren Polarisierung in sog. "Ballerspiele" und "Barbiespiele" Geschlechtszuschreibungen reproduziert. Es läßt sich empirisch weiterhin nachweisen, dass Mädchen und Frauen statistisch häufiger an Ziel, Nutzen und Sinn des Spiels interessiert sind, während Jungen und Männer eher die Technik spannend finden.

Auf anderen Ebenen bewegen sich Einschreibungen geschlechtlich markierter Körperbilder im Kontext grafischer und animierter Computerdarstellungen. Beispiele überzogener Geschlechtsstereotypisierung in menschlichen oder Menschlichkeit nachahmenden Repräsentationen finden sich etwa bei Avataren im Netz sowie bei kommerziellen und in der Entwicklung befindlichen Interface-Agenten (vgl. Bath 2002a, b).

Im Bereich der Artificial-Life-Forschung und Robotikforschung läßt sich etwa bei der Analyse ihrer Heilsversprechen wiederum schön die Perpetuierung traditioneller und geschlechtscodierter Vorstellungen von Körper und Geist, Vernunft und Gefühl, Allgemeinem und Besonderen nachvollziehen (vgl. Weber 2002b). Auch bei der Konstruktion von Robotern selbst läßt sich unschwer die Reproduktion von Gender-Klischees und banalsten Stereotypen finden. So werden Staubsauger-Robots gern in weibliche Formen verpackt. Aber auch für andere Funktionen zieht man gern das sogenannte 'Weibliche' heran: So werden neue, sog. androide Roboter, sofern sie nicht als 'Maschine pur' verpackt werden, oft als 'weiblich' inszeniert. Im kommerziellen Robotik-Projekt 'Valerie, a domestic Android' soll der Roboter in Form einer großen, schlanken, dunkelhaarigen Schönen (?) mit grau-grünen Augen und tiefrotem Dress gestaltet werden. Dies sei angeblich geeignet, Ängste vor Robotern abzubauen, da ja bekanntermassen Frauen weniger furchterregend seien: "We are pleased to announce the commencement of our own domestic android project. This android will look like a woman. Why? Because a woman is less threatening than a full-sized adult male." (<http://www.androidworld.com/prod19.htm>, S. 1; last access 25.2.2002) (vgl. Weber 2002a)

Aber auch im Bereich bildgebender Verfahren werden körperliche Realitäten konstruiert. Am Beispiel der Informatisierung der Neurowissenschaften und der Biomedizin wird die Definitionsmacht deutlich, die InformatikerInnen bei der Herstellung von Wissen über Natur, Leben und Körper haben. Technisch generierte Bilder verändern unsere Körperwahrnehmung und die Grenzen zwischen normal und krank. Feministische Wissenschaftskritik gepaart mit partizipativen Entwicklungsmethoden könnten hier zur Sichtbarmachung und Auflösung stereotyper Dichotomien beitragen (Schinzel 2002b, Schmitz 2001, Schmitz & Schinzel 2002).

Einschreibungen, sprachliche Konstruktionen und informationstechnologische Artefakte

Eine zweite Kontroverse drehte sich um die verschiedenen Formen von Einschreibungen. Es wurde zwischen sprachlichen und informationstechnischen Konstruktionen unterschieden. In der Arbeitsgruppe stellte sich die Frage, ob Einschreibungen und Klassifizierungen, die durch Sprache vorgenommen werden, genauso starr sind wie informationstechnisch materialisierte Kategorien und Unterscheidungen. Welche Systeme sind offener oder geschlossener?

Die Mehrzahl der Teilnehmenden tendierte dazu, Klassifizierungen in den Artefakten der IT-Technologien als rigider einzuschätzen als diejenigen in natürlichen Sprachen. Die Informatik als Wissenschaft der Wiederholung perpetuiere permanent gewisse Formalisierungen, Quantifizierungen und Klassifizierungen. Die Methoden der Klassifizierung sind dabei disjunkt und hierarchisch aufgebaut, z.B. in Menüstrukturen oder Bäumen. Daraus ergab sich die Frage, inwieweit die Klassifizierung nur ordnenden Charakter hat oder damit auch soziale Dominanzverhältnisse einhergehen.

Gegen die Auffassung, dass Sprache im Vergleich zu informationstechnischen Systemen flexibler sei, wurde eingewandt, daß auch die natürliche Sprache hierarchisch aufgebaut sei und Begriffsbildungen per se zu identitätslogischen Effekten (vgl. Adorno 1982) führten. Dies ließe sich z.B. gut am sog. principium identitatis exemplifizieren: Zum einen bedarf Denken der Begrifflichkeit, diese erweckt aber wiederum den Schein, daß das zu Denkende mit den Begriffen in Eins fällt. Nicht nur werden zu differenzierende Sachverhalte mit den gleichen Begriffen belegt, auch das (formal) identische Objekt ist aufgrund seiner Veränderbarkeit im historischen Verlauf nicht immer absolut identisch mit sich selbst. In diesem Prozeß finden Vereinfachungen, Formalisierungen und identitätslogische Verkürzungen statt, die zumeist in den sprachlichen Verständigungsprozessen nicht noch einmal reflektiert werden. Auch Niklas Luhmann geht davon aus, daß Erkenntnis bzw. begriffliches Denken der Leitunterscheidungen (bzw. Formen) bedarf, damit Kognition nicht der Mannigfaltigkeit von Welt, dem ubiquitären Rauschen, erliegt (vgl. Luhmann 1995). Diese Unterscheidungen wie etwa Sein und Nichtsein, System und Umwelt, Subjekt und Objekt gründen aber nicht mehr – wie bei Kant – in verbindlichen, transzendentalen Kategorien, sondern sind mehr oder weniger kontingent und von der BeobachterIn zu wählen. Unterscheidungen basieren demzufolge auf einer Arbeit des Diskriminierens, die selbst aufgrund der dichotomen und

hierarchischen Strukturierung dieses Prozesses wiederum unweigerlich zur Produktion von blinden Flecken führt, die sich in Abhängigkeit von der gewählten Leitunterscheidung ergeben. Mit Verweis auf Adorno und Luhmann, aber auch Klinger (1995) läßt sich zeigen, daß auch natürliche Sprache hierarchisch und disjunkt operiert.

Mit einer Sichtweise, welche die Aufgabe der Informatik vor allem in der Informationsmodellierung sieht, würde die Dichotomisierung der Klassifizierungen durch Sprache und Informationstechnik tendenziell abgeschwächt. Dies würde allerdings das Problem, dass InformatikerInnen z.B. in der Objektorientierten Modellierung Klassifizierungen setzen müssen, die im Anschluß von anderen Personen genutzt werden, nicht aufheben.

Es wurde jedoch die Hoffnung geäußert, daß elektronische Netzwerke womöglich flexibler sind und gewisse Freiheitsgrade erlauben, die herkömmlichen Informatiksystemen nicht gegeben sind. Dies führte zu der Frage, inwieweit komplexere Artefakte eher die Möglichkeit bieten, rigide Formalisierungen, Klassifizierungen und Reduktionismen in ihrer Weltbeschreibung abzufedern.

Andererseits sind Klassifizierungen der Sprache und der Informatik inhärent. Man kann ihnen letztendlich nicht entgehen. Es wurde jedoch das Bedürfnis formuliert, die Konstruktion der Artefakte nicht jedesmal einer zweiten Reflexion zu unterziehen. So wie man nicht jedes Mal aufs Neue beim Essen über die Funktion von Gabel und Messer sinnieren möchte, will man auch Technik ohne Bedenken und ohne Neukonfigurationen nutzen können.

Von anderen wurde darauf beharrt, daß Klassifikationen mächtige Technologien sind, die – wenn sie in lebendige und vielgenutzte Infrastrukturen implementiert sind – eben relativ unsichtbar sind, ohne dabei jedoch ihre Wirksamkeit und Macht zu verlieren. Sie transportieren politische und ethische Entscheidungen, die zumeist nicht explizit verhandelt werden (vgl. Bowker/Star 1999; Suchman 1994). Und genau die Reflexion darauf gelte es zu leisten.

Transdisziplinarität

Ein weiterer wichtiger Diskussionspunkt war die Frage des Konnexes von Genderforschung und Informatik und die Möglichkeit von transdisziplinärem Arbeiten. Frauen-, Geschlechter- und Genderforschung hat sich seit jeher als inter- und transdisziplinär verstanden. Diese Transdisziplinarität ist weniger aus der Marginalisierung und Not jeder neuen Forschungsrichtung und Wissenschaft (wie auch der Informatik) geboren, als aus der Überzeugung, dass situiertes kritisches Wissen nicht anders möglich ist. Vor diesem Hintergrund könnte sie für die Informatik Anschlußmöglichkeiten und Ideen liefern.

Genderforschung könnte man in diesem Zusammenhang als Möglichkeit begreifen, die diversen Ebenen des Material-Technischen, Gesellschaftlichen und Symbolischen, die in der Informatik meist unreflektiert nebeneinander verhandelt werden, zusammenzufügen. Dies würde eine weitere Option darstellen, auf die Informatik und ihre Artefakte kritisch zu reflektieren.

Von der Genderforschung bzw. generell kritischer Wissenschaftsforschung ließe sich der historisch-kritische und gendersensitive Zugang lernen, aber auch die Benutzung von neuen Werkzeugen wie Metaphern-, Diskurs- und Epistemeanalyse und die kritischen Optionen einer kritischen Ideengeschichte (vgl. Crutzen 2002).

Mit diesem Wissen und der dazugehörigen Methodik wird die Aufmerksamkeit für Standardisierung, Formalisierung und damit verbundene Ausgrenzungen gestärkt. Auch in der Lehre wäre zu betonen, dass es immer nur einen Werkzeugkasten von verschiedenen Methoden gibt, nicht aber DIE Methode. Um dies zu leisten, wäre ein Propädeutikum der Transdisziplinarität zu entwickeln, das das kritische Instrumentarium der Gender- und Wissenschaftsforschung auf die Informatik anwendet und wiederum der Informatik die Implementierung dieser Einsichten in ihre eigenen Methoden, Werkzeuge und Lehrbücher ermöglicht.

Gleichzeitig könnte die Auseinandersetzung der Genderstudien mit der Informatik eine Öffnung der Genderstudien ermöglichen, insofern für letztere die Auseinandersetzung mit der Informatik eine Herausforderung ist, mit der sie sich selbst wandeln müssen, um ihrem neuen Forschungsfeld gerecht werden zu können.

Ein- und Ausgrenzungen

Der vierte Punkt, der hier besonders erwähnt werden soll, betrifft das wechselseitige Verhältnis der Definitionen von Informatik und Geschlecht. Kennzeichnend für die Diskussion der Arbeitsgruppe war es, sich nicht auf eine positive Definition und Bestimmung der Informatik einzulassen. Aus theoretische Gründen plädierten die TeilnehmerInnen gegen eine explizite Bestimmung der Inhalte oder des

Umfangs der Disziplin. Erstaunlicherweise wurde jedoch sowohl bei der konkreten Nennung von Beispielen zur Einschreibung von Geschlecht als auch deren Kritik und Zurückweisung immer wieder implizit auf ein Wissen darüber zurückgegriffen, was Informatik ist bzw. nicht ist. Im Falle der Kritik führte das dann immer dazu, dass das, was sich als geschlechtliche Konnotation und Einschreibung unmissverständlich zeigt, nicht als Teil der Informatik definiert und ausgegrenzt wurde.

Dieses Muster der Konstruktion eines gegenseitigen Ausschlusses von Informatik und Gender findet sich interessanterweise auch in der umgekehrten Richtung. So wird in der Informatik traditionellerweise zwischen Softwaregestaltung und -nutzung relativ rigide getrennt, wobei oft nur erstere als 'wahre' Informatik gilt. Wenn aber von feministischer Seite argumentiert wird, dass eine Integration der Softwarenutzung in den Kernbereich der Informatik notwendig ist, um sinnvolle Artefakte konstruieren zu können und unnötige Reduktionismen und Ausgrenzungen zu verhindern, so gilt diese Forderung nach dem Einbezug der UserInnen plötzlich als selbstverständlicher Bestandteil der Informatik. Der spezielle "Nutzen" der Gender Studies als "Eye Opener" für bestimmte blinde Flecken in der Informatik und Softwareentwicklung wird damit verdeckt.

Insgesamt wird an diesen beiden Argumentationsstrategien deutlich, dass die Definitionen von Geschlecht und von Informatik permanent verschoben werden und fließend sind bzw. gehalten werden. Die Grenzen der Informatik werden dabei – je nach Kontext – stets so gesetzt, dass die Inhalte und Anwendungen der Disziplin nichts mit Geschlecht oder gar mit Geschlechterforschung zu tun haben. Und falls sich einmal die Bedeutung einer feministischen Kritik nicht verleugnen lässt, so wird diese plötzlich als allgemeine kritische Sicht signifiziert, die sich längst – und zwar ohne feministische Intervention – in der Informatik etabliert habe.

Deutlich wird an dieser wechselseitigen Verschiebung, dass die Zugehörigkeit zur Informatik und der Ausschluss aus der Disziplin auf das Engste mit Anerkennung und Machtaspekten gekoppelt ist. Insbesondere die Teilbereiche der Informatik, die – wie die Angewandte bzw. Sozialorientierte Informatik oder das Gebiet "Informatik und Gesellschaft" – selbst stark um die Anerkennung und Integration innerhalb der Disziplin ringen, scheinen sich durch die Geschlechterforschung besonders bedroht zu fühlen. Dabei könnten gerade kritische Perspektiven auf die Informatik, Theorie(n) der Informatik und feministische Wissenschafts- und Technikforschung gegenseitig voneinander lernen und ein unbefangenes Zusammendenken und Zusammenarbeiten dieser Sichtweisen produktiv sein!

Literatur

- Adorno, Theodor W. 1982: Negative Dialektik. Frankfurt a.M.: Suhrkamp (im Orig. 1962).
- Bath, Corinna 2002a: Wie "Menschlichkeit" gemacht wird. Geschlechterrepräsentation von Avataren und Agenten. In: Gleichstellungsbeauftragte der Universität Köln (Hrsg.): Frauen in den neuen Technologien. Köln, S. 5-21.
- Bath, Corinna 2002b: Was können uns Turing-Tests von Avataren sagen? Performative Aspekte virtueller Verkörperungen im Zeitalter der Technoscience. In: Epp, Astrid/Taubert, Niels C./Westermann, Andrea (Hrsg.) Technik und Identität. Tagung Juni 2001 Universität Bielefeld: IWT-Paper 26, Bielefeld, 79-99. Auch unter: <http://archiv.ub.uni-bielefeld.de/kongresse/technikidentitaet/Bath.pdf>
- Bowker, Geoffrey C./Star, Susan Leigh 1999: Sorting Things Out. Classification and Its Consequences. Cambridge, Mass./London: MIT Press.
- Crutzen, Cecile 2002: Gender als Phantasie oder Potential einer Disziplin? (In diesem Band)
- Erb, Ulrike 1996: Frauenperspektiven auf die Informatik. Informatikerinnen im Spannungsfeld zwischen Distanz und Nähe. Münster: Westfälisches Dampfboot.
- Green, Eileen/Owen, Jenny/Pain, Den 1993 (Hrsg.): Gendered by design? Information Technology and Office Systems. London.
- Hofmann, Jeanette 1999: Writers, Texts and Writing Acts: Gendered User Images in Word Processing Software. In: Donald MacKenzie, Judy Wajcman (Hrsg.): The Social Shaping of Technology. Second Edition. Open University: Buckingham, Philadelphia, pp. 222-243.
- Klinger, Cornelia 1995: Beredtes Schweigen und verschwiegenes Sprechen: Genus im Diskurs der Philosophie. In: Hadumod Bußmann/Renate Hof (Hrsg.): Genus. Zur Geschlechterdifferenz in den Kulturwissenschaften. Stuttgart, S. 409-445.
- Luhmann, Niklas 1995: Dekonstruktion als Beobachtung zweiter Ordnung. In: Henk de Berg / Matthias Prangel (Hg.): Differenzen. Systemtheorie zwischen Dekonstruktion und Konstruktivismus. Tübingen/Basel, S. 9-35.
- Schinzel, Britta 2002a: Die Bedeutung des Status der Informatik als Ingenieurwissenschaft für die Exklusion von Frauen. In: Verein FluMiNuT (Hrsg.): Wissen_schaf(f)t Widerstand. 27. Kongress von Frauen in Naturwissenschaft und Technik. Wien, S. 321-326.
- Schinzel, Britta 2002b: Genderforschung am Beispiel der Informatisierung der Neurowissenschaften. (in diesem Band)

- Schmitz, Sigrid 2001: Gender-Forschung zur Mediatisierung der Neurowissenschaften. FiFF-Kommunikation 3/01, S. 36-41.
- Schmitz, Sigrid & Britta Schinzel 2002: GERDA: A brain research information system for reviewing and deconstructing gender differences. In: Ursula Pasero / Anja Gottburgen (Hg.): Wie natürlich ist Geschlecht? Westdt. Verlag, S. 126-139.
- Suchman, Lucy 1994: Do Categories Have Politics? The Language/Action Perspective Reconsidered. In: Computer Supported Cooperative Work (CSCW) 2: 177-190.
- Wagner, Ina 1989: Regulierung der Krankenhausarbeit. Ein Vergleich des Computereinsatzes in Österreich, Frankreich und den USA aus der Perspektive der Organisation von Pflegearbeit und Labortätigkeiten. In: Journal für Sozialforschung, 29. Jg., Heft 2, 165-180.
- Wagner, Ina 1991: Transparenz oder Ambiguität? Kulturspezifische Formen der Aneignung von Informationstechniken im Krankenhaus. In: Zeitschrift für Soziologie, Jg. 20, Heft 4, S. 275-289.
- Weber, Jutta 2002a: Berechenbare Organismen, wildgewordene Maschinen? Heilsvisionen und Apocalyptica der Artificial-Life-Forschung. In: Zentrum für Interdisziplinäre Frauenforschung (Hg.): Cyberfeminismus. Feministische Visionen mit Netz und ohne Boden. Bulletin Nr. 24. Berlin: Universitätsdruckerei der HU, S. 73-86.
- Weber, Jutta 2002b: Klassifizierung des Lebendigen. Über Ein- und Ausschliessungen am Beispiel der Robotik und Artificial Life- Forschung. (In diesem Band)
- Webster, Juliet 1993: From the word processor to the micro: gender issues in the development of information technology in the office. In: Green/Owen/Pain 1993, pp. 111-123.

Arbeitsgruppe

Theorie der Anwendungen der Informatik

Havarie und Sanierung

Aufruf zur AG Theorie der Anwendungen der Informatik

Gerhard Wohland, Diebold Deutschland

Vorab

Mit den Begriffen: EDV, Programm, Software oder "Anwendung" ist hier immer ein Werkzeug zur Wertschöpfung gemeint. Theorie wird als "Praxis im Kopf" genommen.

Die Idee

Durch die Globalisierung der Wirtschaft haben es überraschungs-robuste Unternehmen leichter. Sie erzeugen den Marktdruck, unter dem die trägen Unternehmen leiden.

In dynamischer Umgebung werden auch konventionelle Unternehmen "zerlegt" in Zentrum und Peripherie. In der marktnahen Peripherie entsteht eine neue Form von Wissen. Dieses ist zu konkret, um mit üblichen Mitteln kommuniziert zu werden. Dem konventionellen Manager oder EDV-Projektleiter macht sich die neue Form als sogenanntes "Gegenwissen" bemerkbar. In dieser Situation handelt die Organisation eines Unternehmen wesentlich klüger, als seine Mitglieder denken können. So kommt es, dass die Macher das Problem mit seiner Ursache bekämpfen.

Dabei spielt die EDV, und damit die Informatik, eine tragende Rolle.

Im Prozess einer Wertschöpfung laufen Triviales (Kompliziertes) und Komplexes nebeneinander her. Wenn, wegen wachsender Dynamik, mehr Überraschendes geschieht, wächst der komplexe Anteil der Prozesse. EDV-Systeme sind wie andere Maschinen empfindlich gegen Überraschungen. Deswegen kann nur der jeweils triviale, formale Anteil eines Prozesses mit EDV-Programmen verwoben werden.

Wird in bester, aber naiver, Absicht trotzdem mehr versucht, gibt es nur zwei Möglichkeiten: Eine starke Organisation "wehrt" sich mittels ihrer Lebenserhaltungs-Reflexe; das zugehörige EDV-Projekt tritt auf der Stelle. Eine bereits geschwächte Organisation geht daran zu Grunde.

Diese Situation nenne ich *Havarie*.

Analyse dieser Havarien auf Basis der Unterscheidung von "lebendig" und "tot" ist für mich der Zugang zu einer "Theorie der Anwendungen der Informatik".

Das Havarie-Problem ist neu. Es ist erst mit der Globalisierung entstanden. Im Zeitalter tayloristischer Wertschöpfung war es weitgehend unbekannt.

Mögliche Themen für die Arbeitsgruppe

- *Unterscheidungen*
Lebendig - tot, kompliziert - komplex, Daten - Information, Schnittstelle - Schnittmenge, Verhalten - Werte, Methode - Theorie, Selbstorganisation - Selbstbestimmung, Führung - Steuerung, Zentrum - Peripherie.
- *Globalisierung und dynamische Wertschöpfung*
Globale Märkte sind eng, weil sie groß sind.
- *Neue Wissensformen und ihre Kommunikation*
Wissensströme, Gegenwissen.
- *Typen von EDV-Projekten*
Trivial kompliziert, komplex.
- *Sanierung von EDV-Havarien*
Ausflaggen von Tabuzonen, Veredelung des Gegenwissens.
- *Kriterien post-tayloristischer EDV*
Neutral, medial, nicht integriert, Werkzeug-Charakter.

Unser aller Profession gib uns heute ... oder die Frage nach einer mäeutischen Informatik

Peter Bittner, Humboldt-Universität zu Berlin

*"Der Kopf ist rund, damit die Gedanken
ihre Richtung ändern können."
(Francis Picabia)*

Vorgeplänkel

Diese kleine Arbeit¹ steht in einer Serie von Arbeiten (Bittner 2002, 2003a, 2003b), die zusammen drei übergeordnete Konstruktionsprinzipien von Theorien (angewandter) Informatik markieren sollen und dem Theorie-Diskurs "querdenkende" Impulse geben möchten.

Ich möchte diese theoretischen Überlegungen weitgehend "pragmatisch" orientieren. Im Kern steht die Frage der Vermittlung der Informatik in ihrer lebensweltlichen Praxis. Wie kann dies konzeptionell (Allgemeine Informatik), praktisch (Handlungsorientierung) und bezogen auf das professionelle Handeln (Mäeutische Informatik) geschehen?

An dieser Stelle möchte ich den Blick auf die Professionssoziologie und die Professionalisierungsdebatte in der Pädagogik lenken. Ich versuche einige Brückenschläge zur Informatik und hoffe auf eine angeregte Diskussion. Der geneigte Leser wird Querbezüge zum Beitrag von Eva Hornecker (in diesem Bericht) entdecken.

Theorien der Informatik: Unser aller Profession gib uns heute ... oder "mäeutische" Informatik

Blicken wir auf das informatische Handeln als professionelles Handeln. Professionalisierung wird für die Informatik erst seit kurzem im angelsächsischen Bereich (z.B. Denning, 2001) oder im deutschsprachigen Raum (z.B. Schinzel, 2001) (wieder?) eingehender diskutiert. Diese Diskussionen aufeinander beziehen zu können, setzt aber voraus, dass man die jeweiligen strukturellen Differenzen wahrnimmt. Während sich in den USA Berufe eher bottom-up professionalisieren, geschieht dies im deutschsprachigen Raum eher top-down (vgl. Koring, 1999: Teil 6.4). Eine einfache Übernahme angelsächsischer (zumeist funktionalistischer) Professionsbegriffe verengt dabei unsere Sicht und führt bezogen auf die "Professionalisierung" der Informatik zu unnötigen und schwerwiegenden Schließungen, die sich aus den neueren professionssoziologischen Diskursen nicht ergeben müßten.

Ein professionssoziologischer Exkurs

Bestimmend in der Professionssoziologie sind *klassisch-kriterielle (funktionalistische)* Ansätze, wie man sie z.B. bei Parsons (1939) oder Goode (1957, 1972) findet. Professionen sind dabei im wesentlichen akademische Berufe (mit langer spezialisierter Ausbildung), bei denen eine Steigerung von Rationalität bei der Verwirklichung von Handlungszielen feststellbar ist. Sie sind markiert durch eine deutliche Begrenzung der Kompetenz, die durch die Aufgabenstellung und das Problem des Klienten (Ausrichtung auf wichtige individuelle oder kollektive Probleme) definiert ist. Professionelles Handeln sei nicht von partikularen Interessen (z.B. Sympathie oder Antipathie) geprägt. Die hierfür notwendige hohe Autonomie der Professionellen schlägt sich nach Goode nieder:

- im Recht, den eigenen Nachwuchs zu bilden und zu erziehen
- im Recht der professionellen Selbstkontrolle
- bei der (autonomen) Strukturierung des professionellen Berufsalltags

Hinzu kommt eine spezielle Ethik, die den Schutz der Klienten in ihrer jeweiligen Situation vor Ausbeutung sichert und eine Selbstverpflichtung der Professionsmitglieder beinhaltet. Bezogen auf den Wunsch der "Verallgemeinerung"² der Informatik haben wir aber das Problem, dass durch die erhöhte

¹ Sie basiert auf einem Vortrag *Unser aller Profession gib uns heute ...*, den ich im Rahmen der Sitzung des Arbeitskreises „Verantwortung und Informatik“ des FB 8 „Informatik und Gesellschaft“ der GI am 15.03.2002 an der HU Berlin gehalten habe.

² In Anlehnung an Wille (1996b, 1988) – dessen Ansatz stark durch v. Hentig (1972) geprägt ist – soll der Anteil Allgemeiner Wissenschaft, der für die Informatik relevant ist, Allgemeine Informatik genannt werden. Diese sei charakterisiert durch (1) die Einstellung, Informatik für die Allgemeinheit zu öffnen, sie prinzipiell lernbar und kritisierbar zu machen, (2) die Darstellung informatischer Entwicklungen in ihren Sinngebungen, Bedeutungen und Bedingungen, (3) die Vermittlung der Informatik in ihrem

Selbstkontrolle und kollegiale Kontrolle Professionen unempfindlicher für Laienkritik und eine "Kontrolle" durch die Gesellschaft werden.

Jahrzehntlang stand die Professionssoziologie fest und ziemlich unerschüttert in der Tradition von Carr-Saunders und Wilson (1933), die Waddington (1996) "checklist approach" oder auch "trait approach" nannte. Problematisch ist, dass sich anhand von derartigen Listen komplexe Identitäten von Gruppen, die mit einer Vielzahl von Adressaten sowie ihren Trägern und der ganzen Gesellschaft interagieren, kaum erfassen lassen. Desweiteren wird über die Tätigkeit der Professionellen und die zugehörigen Denk- und Handlungsmuster nur wenig ausgesagt.

Die Arbeiten Oevermanns (1978, 1983) markieren eine professionssoziologische Wende³, die es auch für die Informatik ermöglichen sollte, einen anderen, weniger schließenden Professionsbegriff zu entwickeln. Dabei kann die Informatik m.E. sehr von der "Professionalisierung-Debatte" der Pädagogik lernen. Den Professionen werden innerhalb dieses Ansatzes drei zentrale Aufgaben zugewiesen:

- Sie seien zum einen mit der kritischen Prüfung von Wahrheitsbehauptungen zu beschäftigen, wie dies in der Wissenschaft der Fall ist.
- Sie seien für die Beschaffung von Konsens und Konformität zuständig, wie dies bei Richtern und Rechtsanwälten, teilweise auch bei Politikern der Fall ist.
- Sie hätten für die Bereitstellung therapeutischer Leistungen (um Menschen gesund, handlungsfähig und orientierungsfähig zu halten) zu sorgen, wozu Ärzte, aber auch Priester, Lehrer und Sozialpädagogen zählen.

Wahrheit, Konsens und Therapie werden dabei bei Oevermann als die wesentlichen Funktionsvoraussetzungen jeder Gesellschaft verstanden. In jeder Profession spielen alle drei Aspekte eine gewisse Rolle. Es gibt aber Spezialisierungen. So wird z.B. die Pädagogik von Oevermann im Bereich der Therapiebeschaffung angesiedelt. Es stellt sich die Frage, ob man z.B. auch die Informatik als Therapie im Sinne des *Ausräumens von Beschränkungen persönlicher Handlungsautonomie* denken könnte. Zunächst aber zur Frage, wie denn Professionalität im Handeln realisiert werden kann.

Realisierte Professionalität im informatischen Handeln

Für Oevermann verbinden sich in der *realisierten Professionalität* die:

- *wissenschaftliche Kompetenz*, die den Umgang mit Theorie und den engen Kontakt zum Fachwissen der Disziplin (Jura, Medizin, Theologie oder Erziehungswissenschaft) betrifft und die
- *hermeneutische Kompetenz*, aufgrund derer ein bestimmtes Problem verstanden werden kann. Dazu ist wissenschaftliches Wissen allein nicht ausreichend; praktische Erfahrung ist notwendig.

Das wissenschaftliche Wissen ist aber nicht umstandslos auf praktische Probleme anwendbar, weil es abstrakt ist und die Wirklichkeit demgegenüber sehr konkret und vielfältig. Die intuitive Auseinandersetzung ist nicht ausreichend, weil Intuition und Erfahrung sich im Kreise drehen können. Die Aufgabe des Professionellen besteht darin, zum Zweck der Bearbeitung eines Problems, das wissenschaftliche und das hermeneutisch-fallbezogene Wissen, so zu verbinden, dass praktische Deutungen und Handlungsstrategien zustande kommen.

Eine zentrale Stellung nimmt im professionellen Handeln die stellvertretende Deutung ein. Professionelle deuten für Klienten (Dienstleistungssicht!) ein Problem, das der Klient selbst nicht verstehen und lösen kann, weil er von dem Problem betroffen ist. An dieser Stelle ergeben sich wesentliche Fragen an die Informatik: Wie handeln hier Informatiker?⁴ Wie verhält es sich in diesem Sinne mit partizipativen Verfahren? Können/Dürfen Informatiker handeln, wenn ein Klient sein Problem nicht (hinreichend) versteht?

Im Anschluss an die stellvertretende Deutung werden bewährte Handlungsstrategien der Profession auf den Fall angewandt und im Zuge dieser Anwendung so modifiziert, dass sie auf den konkreten Fall passen. Professionen sieht Oevermann als einen wichtigen gesellschaftlichen Ort der Vermittlung von Theorie und Praxis in der modernen Welt an. Professionelle haben diese Vermittlung konkret bei jedem bearbeiteten Fall (neu) zu leisten.

lebensweltlichen Zusammenhang über die Fachgrenzen hinaus, (4) die Auseinandersetzung über Ziele, Verfahren, Wertvorstellungen und Geltungsansprüche der Informatik.

³ Oevermanns Ansatz wird zumeist als *strukturtheoretisch*, zuweilen auch als deduktiv charakterisiert.

⁴ Trotz der durchgängigen Nutzung der männlichen Berufsbezeichnung möchte der Autor die Informatikerinnen mit eingeschlossen wissen.

Ganz ähnlich die Position Nohls (2002/1933), der versucht, Kriterien für die Angemessenheit pädagogischen Handelns zu entwerfen. Versucht man dessen Argumentation auf die Informatik zu übertragen, dann wird der Informatiker zu einer Vermittlungsinstanz zwischen Subjektivität (Perspektive des Klienten/Adressaten) und Objektivität (gesellschaftliche Anforderungen). Sein Kennzeichen ist, dass er aufgrund wissenschaftlicher und praktischer Kenntnisse zum einen auswählend und vermittelnd, zum anderen interpretierend tätig ist.

Mäeutische Informatik?!

Diese Überlegungen lassen sich – Koring (1999: Teil 6.8) folgend – konstruktiv weiterführen. Sein Bild professioneller Pädagogik ist geprägt von zwei regulativen Ideen, die ich hier sinngemäß auf die Informatik übertrage:

- Der Informatiker hat sich an der Ermöglichung von Selbsttätigkeit/Selbständigkeit der Klienten zu orientieren.
- Er hat sich an der Struktur einer mäeutischen Informatik zu orientieren, also einer Informatik, die an schon vorhandene Kompetenzen produktiv anknüpft.

Der Informatiker muss also mit situativen Arrangements dafür sorgen, dass Selbsttätigkeit möglich ist und gefördert wird. Der Klient muss sich produktiv mit dem, was entstehen soll (Informatik-System) und den kulturellen Veränderungen befassen – ansonsten ist die "anwaltliche" Aufgabe des Informatikers nicht wahrnehmbar.

Dies führt uns gradewegs zur Mäeutik als (ehemals pädagogischer) Hebammenkunst. Für Informatiker in der Dienstleistungssituation heißt dies, darin geschult zu sein, im (dialektischen) Gespräch ein Wissen/Können zutage zu fördern, das dem Gegenüber zunächst verborgen war. Im professionellen Handeln strukturiert und begleitet der Informatiker den Prozess, in welchem die Klienten versuchen, die Probleme und Bedingungen ihres eigenen "Arbeitens" zu artikulieren. Der Informatiker deutet diese artikulierten neue Bedeutung in ihrem Verhältnis zum Thema, zum Problem, zur Person und zum Gestaltungsprozess selbst. An diesen "informatischen" Deutungen können die Adressaten erkennen, an welcher Stelle sie im Gestaltungsprozess stehen.

Es zeigt sich, dass wir aus dem professionssoziologischem Diskurs lernen und wichtige Anregungen für unser Verständnis professionellen informatischen Handelns gewinnen können. Diese Diskussion steht für die Informatik allerdings erst am Anfang.

Literatur

- Bittner, Peter: Theorien der Informatik und Kritische Theorie. Über die Vermittlung zweier Denkwelten. In: Gehrlein, Ulrich; Krebs, Heike; Pfeiffer, Judith; Schmidt, Jan C. (Hrsg.): Perspektiven interdisziplinärer Technikforschung. Konzepte, Analysen, Erfahrungen. Münster: agenda-Verlag, 2002, S. 209-219.
- Bittner, Peter: Theorien der Informatik – allgemein, handlungsorientiert, mäeutisch. Ein "kritisches" Manifest. Beitrag zu einem Sammelband anlässlich der Emeritierung von Prof. Dr. Rudolf Wille. 2003a. (in Vorbereitung, erscheint im Frühjahr 2003).
- Bittner, Peter: Informatik (anders) denken ... Über "gute" Disziplinarität, Kritische Theorie und Informatik. In: Böhme, Gernot; Manzei, Alexandra (Hrsg.): Kritische Theorie der Natur und der Technik. München: Wilhelm Fink Verlag, 2003b (in Vorbereitung).
- Carr-Saunders, Alexander M.; Wilson, P.A.: The Professions. Oxford University Press, 1933 (reprint by Frank Cass, London 1964).
- Denning, Peter J.: Who are we? Communications of the ACM (CACM), 44 (2), S. 15-19.
- Goode, William J.: Community within a Community: The Professions. American Sociological Review, 22, 1957, S. 194-200.
- Goode, William J.: Professionen und die Gesellschaft. Die Struktur ihrer Beziehungen. In: Luckmann, Thomas; Sprondel, Walter M. (Hrsg.): Berufssoziologie. Köln: Kiepenheuer & Witsch, 1972, S. 157-167.
- Hentig von, Hartmut: Magier oder Magister? Über die Einheit der Wissenschaft im Verständigungsprozeß. Stuttgart: Klett, 1972.
- Koring, Bernhard: Grundprobleme pädagogischer Berufstätigkeit. Thema 6: Die Frage nach der Professionalität pädagogischer Tätigkeit. (Text und Folien zur Veranstaltung) <http://www-user.tu-chemnitz.de/~koring/sem-v1-paed-beruf/tma6.htm>, Stand: 05.10.1999 (zuletzt gesehen am 06.01.2003).
- Nohl, Herman: Die pädagogische Bewegung in Deutschland und ihre Theorie. Frankfurt/Main: Vittorio Klostermann, ¹¹2002 (zuerst 1933).

- Oevermann, Ulrich: Probleme der Professionalisierung in der berufsmäßigen Anwendung sozialwissenschaftlichen Kompetenz. Frankfurt/Main: unveröffentlichtes Manuskript, 1978.
- Oevermann, Ulrich: Hermeneutische Sinnrekonstruktion: Als Therapie und Pädagogik mißverstanden, oder: das notorische strukturtheoretische Defizit pädagogischer Wissenschaft. In: Garz, Detlev; Kraimer, Klaus: Brauchen wir andere Forschungsmethoden? Frankfurt/Main: Scriptor, 1983, S. 113-155.
- Parsons, Talcott: The Professions and Social Structure. *Social Forces*, 17 (1939), S. 457-467.
- Schinzel, Britta; Kleinn, Karin: Quo vadis, Informatik? *Informatik-Spektrum*, 24 (2) 2001, S. 91-97.
- Waddington, I.: Professions. In: Kuper, Adam; Kuper, Jessica (eds.): *The Social Science Encyclopedia*. London: Routledge, ²1996, S. 677-678.
- Wille, Rudolf: Allgemeine Wissenschaft als Wissenschaft für die Allgemeinheit. In: Böhme, H.; Gamm, H. J. (Hrsg.): *Verantwortung in der Wissenschaft*. TH Darmstadt, 1988, S. 159-176.
- Wille, Rudolf: *Allgemeine Mathematik – Mathematik für die Allgemeinheit*. TH Darmstadt: FB4-Preprint Nr. 1822, 1996b.

Über den notorisch unproduktiven Umgang mit Computersystemen

Peter Brödner, Institut für Arbeit und Technik, Gelsenkirchen

Das Produktivitätsparadox der IT

Das sogenannte "Produktivitätsparadox der IT" verweist auf den erstaunlichen Sachverhalt, dass trotz jahrzehntelanger massiver und im Zeitverlauf anschwellender Investitionen in Informationstechnik keine merklichen Produktivitätsschübe über übliche Produktivitätsfortschritte hinaus festzustellen sind. So haben sich etwa in den 30 Jahren von 1960 bis 1990 die realen IT-Investitionen in den USA nach anfänglich fast konstantem Verlauf bis 1975 auf niedrigem Niveau (ca. 20 Mrd. USD p.a.) bis zum Jahre 1990 auf 220 Mrd. USD mehr als verzehnfacht. Im selben Zeitraum stieg die Produktivität in den produzierenden Sektoren mit relativ niedrigen gleichbleibenden jährlichen Zuwachsen, während in den Dienstleistungssektoren die Produktivität seit 1972 stagnierte (Brynjolfsson 1993). Das hat den Nobelpreisträger für Ökonomie Robert Solow zu dem berühmt gewordenen Ausspruch veranlasst: "You can see the computer age everywhere except in the productivity statistics" (Solow 1987).

Daran hat sich bis heute nicht viel geändert. Noch immer lässt sich auf makroökonomischer Ebene kein zuverlässiger Zusammenhang zwischen IT-Investitionen und Produktivitätswachstum nachweisen, obgleich (wiederum in den USA) seit 1991 die IT-Investitionen die Investitionen in Produktionstechnik übersteigen (Stewart 1997). Nun gab es in der zweiten Hälfte der 90er Jahre in den USA einen deutlichen Produktivitätssprung von jahresdurchschnittlich 1,4% in den Jahren 1972-95 auf jahresdurchschnittlich 2,5% in den Jahren 1995-2000. Schon glaubte man feststellen zu können, dass sich nun endlich, wenn auch mit jahrzehntelanger Verzögerung, die gigantischen IT-Investitionen auch auf die gesamtwirtschaftliche Produktivität positiv auszuwirken beginnen. Doch weit gefehlt. Wie die jüngste vom McKinsey Global Institute (MGI) durchgeführte Produktivitätsanalyse in den USA aufzeigt, ist dieses außergewöhnliche Produktivitätswachstum ausschließlich auf besondere Entwicklungen in den 6 Branchen Groß- und Einzelhandel, Wertpapierhandel, Elektronik, Maschinen (einschl. Computer) und Telekommunikationsdienste zurückzuführen, die auf Restrukturierungen zurückgehen und nur zu geringen Teilen mit erhöhten IT-Investitionen zusammenhängen. Alle übrigen 53 Branchen zusammen tragen nichts mehr zum Produktivitätswachstum bei, da einzelnen Bereichen mit geringfügiger positiver Entwicklung solche mit negativer Produktivitätsentwicklung (trotz hoher IT-Investitionen) gegenüberstehen (McKinsey Global Institute 2001).

Während das IT-Produktivitätsparadox in der wissenschaftlichen Debatte der USA eine beträchtliche Rolle spielt und seit den achtziger Jahren zu zahlreichen Produktivitätsanalysen Anlass gab, wird es in Deutschland schlicht ignoriert. Daher ist letztlich auch nicht festzustellen, ob es hierzulande ebenfalls besteht. Nichts deutet freilich darauf hin, dass es in Deutschland einen positiven Zusammenhang zwischen IT-Investitionen und Produktivitätsentwicklung gibt. Es wird gemeinhin – von Anbietern der IT-Systeme wie von deren Anwendern – einfach unterstellt, dass es ihn gibt, tatsächlich nachgewiesen ist er nicht. Dieser Mythos von der Informationstechnik als Rationalisierungsinstrument ist so tief in den Denkmustern der Akteure verwurzelt, dass sie Produktivitätsprobleme einfach negieren.

Diese tiefe Kluft zwischen Wunsch und Wirklichkeit gilt es zu erklären. Dies geschieht im folgenden auf zwei Ebenen: der praktischen Ebene der Organisation von Arbeit und Wertschöpfung und insbesondere der organisatorischen Einbettung von (informations-)technischen Artefakten sowie der theoretischen Ebene eines angemessenen Verständnisses der besonderen Natur von IT-Systemen.

Begründungsebene 1:

Der Primat der Organisation von Arbeit und Wertschöpfung

Nach der handlungsorientierten Theorie der Technik, wie ich sie im "überlisteten Odysseus" entwickelt habe (Brödner 1997), ist Technik ständige Begleiterscheinung von Arbeit, aus der sie hervorgeht und die sie verändert. Technische Artefakte entstehen – im gesellschaftlichen Zusammenhang – als Produkt der Reflexion menschlicher Arbeit, als Vergegenständlichung von Begriffen und explizitem Wissen über eine eingespielte Praxis. Umgekehrt werden sie wieder gebraucht als Mittel für menschliche Handlungszwecke. Als "geronnene Erfahrung" verkörpern sie ein Stück sozialer Praxis und als Arbeitsmittel stellen sie Handlungsanforderungen (Dialektik von Form und Prozeß). Stets enthalten sie infolge der Dekontextualisierung Leerstellen, die erst im Gebrauch durch Interpretation und Rekontextualisierung ausgefüllt werden. Technische Systeme konstituieren sich mithin in der Anwendung; sie sind offen für abweichende Konventionen des Gebrauchs.

Als eine besondere Form der Vergegenständlichung von Begriffen und explizitem Wissen erweist sich Technik, also Gestaltung und Gebrauch von Werkzeugen und Maschinen, als unmittelbare Verwandte der Sprache (die nur ein anderes Ausdrucksmittel ist). Die Form- und Funktionselemente mit ihren Kombinationsmöglichkeiten bilden die funktionale "Sprache der Artefakte", in der Bedeutung sowohl durch deren Form (Semantik der Artefakte) als auch durch die Handlungsmuster im zweckmäßigen Umgang mit ihnen (Semantik der Situationen) zum Ausdruck gebracht wird. Gebrauchstüchtige Werkzeuge und Maschinen zeichnen sich dadurch aus, daß sie – ausgedrückt in der funktionalen "Sprache der Artefakte" – dem Benutzer Sinn und Verständnis für ihren Gebrauch vermitteln und zugleich im Gebrauch Erfahrungen für sinnvolle, weiterführende Handlungsweisen ermöglichen. Dazu müssen sie erlauben, Form, Funktion und Handlungskontext derart aufeinander zu beziehen, daß sich die Benutzer angeregt fühlen, bei der Lösung ihrer Arbeitsaufgaben erkundend mit ihnen umzugehen und neue Handlungsweisen zu erproben. Entwicklung und Einsatz technischer Artefakte haben mithin einen doppelten Sinn: alte Aufgaben effizienter erledigen und, wichtiger noch, auch neue Aufgaben erschließen und bewältigen zu können. Chomsky paraphrasierend läßt sich sagen: Wir lernen nicht eigentlich eine Technik, sondern wir lernen, mittels einer Technik schöpferisch tätig zu sein.

Technik ist diesem Verständnis zufolge die Gesamtheit von Herstellung und Gebrauch künstlicher Mittel für menschliche Zwecke. Ihr werden damit nicht nur die Artefakte und Sachsysteme selbst zugerechnet (wie das oft in wesentlich zu kurz greifender Perspektive geschieht), sondern gerade auch deren Herstellung und Gebrauch, damit auch die "Systeme zweckrationalen Handelns", durch die Artefakte erst ihren Sinn erhalten und in ihrer Qualität zu beurteilen sind (technische Artefakte und ihre Umgangsregeln sind Ergebnis und Medium von Arbeit zugleich).

Aus diesen Zusammenhängen wird ersichtlich, wie notwendig es ist, Einsicht darüber zu gewinnen, wie der Umgang mit technischen Artefakten in die jeweils konkrete Organisation von Arbeit und Wertschöpfung eingebettet und mit den Arbeitsabläufen verwoben ist. Erst auf dieser Basis lassen sich deren Formen und Funktionen sinnvoll auf den Handlungskontext beziehen und deren Funktionsweise tätig aneignen, um sie wirksam und zweckmäßig zu gebrauchen. Ohne diesen im Rahmen der Organisation sozial konstituierten Handlungskontext ist ein produktiver Umgang mit technischen Artefakten, ist wirksames technisches Handeln nicht zu haben. Diese Einsicht ist umso wichtiger, je komplizierter, vielseitiger und funktionsreicher technische Artefakte gestaltet sind.

Auf dieser Ebene kommt dann auch vieles von dem zum Tragen, was Gerhard Wohland in seinem Einführungstext "Havarie und Sanierung" geltend macht. Als "geronnene Erfahrung" sind technische Artefakte "tote" Gebilde, für sich ungeeignet, Überraschendes zu bewältigen. Erst das durch tätige Aneignung gewonnene Können, ihre individuelle und kollektive Handlungskompetenz, befähigt lebendige Menschen zu beurteilen, ob und wie sie mit Überraschungen fertig werden. Daher ist es so wichtig, dass die durch die technischen Artefakte gestellten Handlungsanforderungen genügend Spielraum für autonomes und kontingentes (eben komplexes) Handeln lassen. Dementsprechend bleibt das Zusammenspiel von Gestaltung und Aneignung technischer Artefakte, mithin die Dialektik der Formalisierung und Vergegenständlichung von Praxis als Form und der Aneignung von Form als neue, erweiterte Praxis eine Daueraufgabe der Verbesserung von Arbeits- und Wertschöpfungsprozessen.

Insbesondere gilt es, dabei die Balance zu finden zwischen der Flexibilität des Handelns und der Rigidität der Handlungsanforderungen. Das erfordert die Bildung angemessener Handlungsstandards und -routinen unter Einschluß der Gestaltung der eingesetzten Arbeitsmittel. Die resultierenden Handlungsanforderungen dürfen weder zu unspezifisch und offen sein, um nicht die Effektivität und Effizienz zu schmälern, noch dürfen sie zu rigide sein, um nicht die Flexibilität und Entfaltung von Handlungskompetenz zu gefährden. Die Handlungskompetenz der lebendigen Arbeit nimmt, zureichende Bedingungen vorausgesetzt, im Handeln nicht ab, sondern zu, im Unterschied zu ihren toten Produkten, die sich durch Gebrauch verzehren. Sie wächst mit jeder Aufgabe, die neue Herausforderungen stellt; sie verkümmert aber auch, wenn ihre Träger immer wieder gleichen Handlungszwängen unterworfen werden.

Mit Blick auf das Produktivitätsparadox der IT läßt sich auf dieser Ebene festhalten, dass die Gestaltung technischer Artefakte meist aus einer sehr einseitigen technikzentrierten Perspektive betrieben wird und Fragen der organisatorischen Einbettung und der Aneignung durch die Benutzer stark unterbelichtet bleiben. Dadurch werden Potentiale produktiver Arbeitsgestaltung und Nutzung der Artefakte nicht ausgeschöpft – insbesondere bei komplizierten, funktionsreichen IT-Systemen, wo diese Defizite stark ins Gewicht fallen (während bei anderen Maschinen Produktivität überwiegend aus ihren material- oder energiewandelnden Eigenschaften erwächst).

Begründungsebene 2: Der Computer als "semiotische Maschine"

Die dargestellten Zusammenhänge der Gestaltung von Form und der Aneignung von Form als erweiterte Praxis gelten für alle technischen Artefakte, vom Faustkeil bis zum Computer. Gleichwohl macht es das Aufkommen informationstechnischer Artefakte erforderlich, aufgrund fundamentaler Differenzen zwei Maschinenklassen zu unterscheiden. Auf der einen Seite haben wir mechanische, elektrische, chemische und biologische Maschinen (bzw. artifiziell geschaffene Prozesse). Sie alle sind geschaffen worden, um den Zustand von Naturstoffen zu verändern oder Energie zu wandeln, und ihre Funktionsweise beruht auf explizitem Wissen über die Natur. Auf der anderen Seite stehen die IT-Systeme zur Manipulation von Daten, und ihre Funktionsweise beruht auf der Formalisierung von Zeichenprozessen. Während erstere sich in ihrem Wirkbereich auf Zustandsänderungen der Natur erstrecken, verbleibt der Wirkbereich der letzteren ganz und gar im sozialen Raum der Zeichen, die sie nach festgelegten Vorschriften verändern. Sehr zurecht werden sie daher auch semiotische Maschinen genannt.

Die besondere Natur der semiotischen Maschine Computer offenbart sich vor allem im Doppelcharakter von Software: Einerseits ist sie durch Menschen (wenn auch schwierig) lesbarer Text, andererseits maschinell ausführbarer Operationscode, mithin eine Maschine. Darin unterscheidet sie sich von vollständigen Beschreibungen herkömmlicher Maschinen (etwa Zeichnungen und Stücklisten), die sich nicht selbst unmittelbar als Maschine realisieren können.

Bei genauerem Hinsehen zeigt sich, dass der Umgang mit Computersystemen auf zwei gekoppelten Zeichenprozessen beruht (von denen der eine freilich auf die syntaktische Ebene der Signalverarbeitung reduziert ist). In der Interaktion mit dem Computersystem werden von Menschen Zeichen eingegeben, die in ihrem jeweiligen Handlungskontext Bedeutung tragen. Im System selbst werden diese von außen lesbaren und sinnvoll interpretierbaren Zeichen auf bloße Signale als deren materielle Träger reduziert, die mittels der Software nach vollständig festgelegten Vorschriften verarbeitet werden. Das mithin vollständig determinierte Resultat dieses Signalverarbeitungsprozesses kann dann, sobald es an der Systemoberfläche erscheint, erneut als Zeichen interpretiert werden. So ist Interaktion gekennzeichnet durch Determination von Signalverarbeitung im Innern und durch sinngebende Interpretation resultierender Signale als Zeichen außerhalb der Maschine. Der soziale Raum der Zeichenprozesse wird dabei nicht verlassen.

Software ist folglich durch und durch semiotischer Natur; ihre Zeichenhaftigkeit erweist sich darin, daß ihr Programmtext oder "Code" die Menge der dadurch determinierten Prozesse der Signalverarbeitung und die Einbettung dieser Prozesse in Kontexte praktischen Handelns folgendermaßen in Relation zueinander setzt:

- Software existiert in Form ihrer endlichen Beschreibung durch den Programmtext;
- dieser Programmtext bestimmt eine i.a. unendliche Menge von Prozessen, deren jeder einen "Lauf" als Abfolge von Signalzuständen der Hardware realisiert;
- diese Prozesse erhalten Sinn dadurch, daß sie interaktiv in Handlungskontexte eingebettet sind, dass Menschen den Programmtext entsprechend ihren Zwecken gestalten und die Ergebnisse vor diesem Hintergrund interpretieren.

Damit verkörpert Software eine semiotische Relation Peircescher Provenienz: "ein Erstes steht für ein Zweites vermittelt und zum Zwecke eines Dritten". Diese drei Aspekte der semiotischen Relation begründen im übrigen auch die Hauptaufgaben der Softwaretechnik (Nake 1997):

- der Programmtext wirft die Frage nach seiner *korrekten* Konstruktion auf,
- bei den Prozessen stellt sich die Frage nach der *Effizienz* der berechenbaren Funktionen,
- die Einbettung in die Praxis erfordert *Nützlichkeit* und *Nutzbarkeit* der Software.

Damit stellen sich ganz neue, heute noch meist unbewältigte Herausforderungen, durch Interaktion mit Computersystemen die Produktivität menschlicher Arbeit zu steigern.

Zunächst einmal: Der Einsatz von IT-Systemen beruht auf der Formalisierung von Kopfarbeit, und Kopfarbeit ist in sich ein besonderer Typ von Arbeit. Kopfarbeit ist Metaarbeit; sie erzeugt und benutzt explizit kodifiziertes Wissen über Arbeit und umfaßt die Analyse, Modellierung, Planung, Anleitung, Koordination, Integration und Kontrolle anderer Arbeit. Sie ist daher grundsätzlich gesellschaftlicher Natur und tritt erst im Zusammenhang mit der Arbeitsteilung und der notwendigen externen Vermittlung von Kooperation (als Koordination der Einzeltätigkeiten) in Erscheinung.

Kopfarbeit beruht auf Zeichenbildung. Sie produziert vergegenständlichtes begriffliches Wissen über Arbeit, dessen sie sich zugleich bedient (Semiotisierung). Dieses Wissen ist prinzipiell abstrakt

(dekontextualisiert), unvollständig (wegen der Grenzen der Explikation) und zersplittert in unterschiedliche Gegenstandsbereiche, Disziplinen und Institutionen (Wissensteilung).

Sodann stellt sich die Frage nach angemessener Formalisierung von Handlungsroutinen in einem äußerst komplexen Handlungsfeld, in dem die Balance zwischen Flexibilität und Rigidität (s.o.) schwer herzustellen und aufrecht zu erhalten ist. Es steckt voller Überraschungen, die hohe Anpassungsleistungen verlangen. Jeder Interpretationsvorgang ist kontingent, Zeichen können jederzeit auch anders interpretiert werden. Zudem stellen Funktionsreichtum und Kompliziertheit der meisten IT-Systeme extreme Anforderungen an deren Aneignung und die aufwendige Entwicklung individueller und kollektiver Handlungskompetenz im Umgang mit diesen Systemen (wie wir das etwa beim Einsatz von PPS-Systemen aufgrund langjähriger Empirie im einzelnen aufzeigen konnten).

Das alles macht sehr aufwendige Prozesse sozialer Interaktion und Kommunikation zur Vermittlung und Synthese von Wissensbereichen, zur Formalisierung und Modellierung verwickelter Handlungsmuster und zur Koordination von Handlungen erforderlich. Kein Wunder also, dass derlei Probleme i.a. noch keineswegs befriedigend bewältigt sind und Produktivitätsfortschritte, von Ausnahmen abgesehen, (noch?) kaum zu verzeichnen sind. So beruht der Mythos der IT als Rationalisierungsinstrument letztlich auf einem Missverständnis der Natur informationstechnischer Artefakte und ihrer leichtfertigen Gleichsetzung mit herkömmlichen Maschinen.

Nachbemerkung

Bei diesen Überlegungen sehe ich mich inspiriert von den beiden Einführungstexten "Havarie und Sanierung" (Gerhard Wohland) und "Informatik als Hybridwissenschaft" (Dirk Siefkes). Und ich denke auch, dass sie für beide Themenfelder Relevantes beitragen.

Literatur

- Brödner, P., 1997: Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen. Berlin: edition sigma
- Brynjolfsson, E., 1993: The Productivity Paradox of Information Technology, CACM 36, No. 12, 67 – 77
- McKinsey Global Institute, 2001: Productivity in the United States.
<http://www.mckinsey.com/knowledge/mgi/reports/productivity.asp>
- Nake, F., 1997: Der semiotische Charakter der informatischen Gegenstände. In: Bayer, U. (Hrsg.): Festschrift zu Ehren von Elisabeth Walter, Baden-Baden: agis
- Solow, R.M., 1987: "We'd Better Watch Out". New York Times Book Review, July 12: 36
- Stewart, T. A., 1997: Intellectual Capital. The New Wealth of Organizations. New York: Doubleday

Historische, wissenschaftstheoretische und kulturelle Wurzeln der angewandten Informatik

Versuch einer Bestimmung in Form von Thesen

Christian Dahme, Humboldt-Universität Berlin

1. Wissenschaftshistorisch lassen sich mindestens drei Wurzeln für die Nutzung des Computers und als Grundlagen für Software erkennen:

- a) Rechner als Mittel zur Unterstützung/Automatisierung von Berechnungen¹
- b) Automatisierung von Tätigkeiten
- c) Software zur Problemlösung

a) ist wissenschaftshistorisch m.E. die älteste. Sie ist u.a. mit dem Wunsch verbunden, aufwendige Berechnungen durch "Rechenwerkzeuge", später Rechenautomaten, zu unterstützen und Fehler zu vermeiden.

Hier geht es um die *Automatisierung des Rechnens* oder allgemeiner um die *Automatisierung von Mathematik*.

Dabei sind zwei Richtungen zu unterscheiden:

a1) - zur Unterstützung wissenschaftlicher Berechnungen

- als Mittel zum Berechnen von Modellen (Durchführung von Modellexperimenten)
- von Schickardts Rechner zur Unterstützung der Berechnungen von Kepler bis hin zur Nutzung von Computern für Modelle der modernen Physik und Naturwissenschaften und damit auch als Grundlage für die Entwicklung von Atombomben, Raketen, Raumfahrt u. dgl. (im Sinne von angewandten Naturwissenschaften)

a2) - zur Unterstützung kommerzieller Berechnungen

- als Mittel zum Berechnen von ökonomischen Sachverhalten und Modellen
- von Pascal, der die Verwaltungsarbeit seines Vaters unterstützen (rationalisieren) wollte, damit dieser mehr Zeit für die Familie hat, über Charles Thomas² bis hin zu ökonomischen Modellen (einschließlich Operations Research) wie z.B. Weltmodellen des Club of Rome bzw. IIASA³

b) bezieht sich auf die *Automatisierung* (von Teilen) *von Tätigkeiten* und geht insbesondere auf die Automatisierung des Webens (Lochplatte von Falcon, Jacquardscher Webstuhl) zurück.

Ausgehend von diesem Prinzip der "Programmsteuerung" entwickelte Herman Hollerith⁴ ein elektro-mechanisches Lochkartensystem für die Auswertung der Volkszählung in den USA von 1890 – im Sinne von Datenerfassung, -Bereitstellung und -Auswertung – insbesondere großer Datenmengen.⁵

Ausgehend von den Erfahrungen bei der Volkszählung wurde dieses Verfahren für Aufgaben von staatlicher bis hin zu betrieblicher Verwaltung wie unterschiedlichste Statistiken, Lagerhaltung, Kostenerfassung oder Lohnabrechnung angewandt.

c) ist wissenschaftshistorisch am jüngsten (entsteht etwa in den 50er Jahre des 20. Jahrhunderts). Sie ist mit der kognitiven Psychologie (insbesondere der kybernetischen Variante) verbunden und führte zum kognitivistischen Paradigma (Kognitivismus).⁶ Psychische Phänomene werden als "Informationsverarbeitung" rekonstruiert.

¹ S. u.a. auch W.&U. Brauer: Änderungen von Forschungszielen und Denkgewohnheiten. In: W. Langenheder, G. Müller, B. Schinzel: Informatik cui bono? GI-FB 8 Fachtagung, Freiburg, September 1992. Springer-Verlag, Berlin.

² Charles Thomas beschäftigte als Chef zweier Versicherungsgesellschaften in Paris eine große Anzahl von "Rechnern" (d.h. Menschen, die für ihn rechneten). So kam er auf die Idee, eine Rechenmaschinenproduktion großen Stils aufzuziehen, um diese "Rechner" durch Rechenmaschinen zu ersetzen. Seine erstes "Arithmometer" war 1820 fertig. Insgesamt wurden ungefähr 1500 solcher Rechenmaschinen in seiner Werkstatt gebaut.

³ International Institute for Applied Systems Analysis.

⁴ Seine Firma war Vorläufer von IBM.

⁵ Das führte u.a. zur Entstehung von Datenbanken (Informationssystemen).

⁶ Einer der damals führenden kybernetisch-mathematischen Vertreter der kognitiven Psychologie war Friedhard Klix (Information und Verhalten; Berlin 1971). Einen guten Überblick über den Kognitivismus findet man bei Günter Sämmer: Paradigmen der Psychologie; 8. Kapitel "Das Paradigma des Kognitivismus" (<http://home.arcor.de/saemmer/dis/diskap8.htm>).

2. Den in 1. dargestellten historischen Wurzeln können unterschiedliche Interpretationen von Software zugeordnet werden.

a) Der (gegenstandsorientierte) Modell-Ansatz⁷

Software basiert auf einem Modell⁸ des Gegenstandes, das in Software übertragen werden soll. Software kann dann als Automatisierung der "Berechnung" von (mathematisierten) Modellen verstanden werden (im Sinne von 1a). Grundlage für die Software ist hier ein Modell des Gegenstandes, und damit die Disziplin⁹ des Gegenstandsbereiches, zu dem der Gegenstand¹⁰ gehört.

Aus dieser Sicht ist Softwareentwicklung ein interdisziplinärer Prozeß. Dabei kann die Software selbst Rückwirkungen auf den ursprünglich abgebildeten Gegenstand haben, so daß man verkürzt auch spricht von: Software als (Mittel zur) System- bzw. Organisationsgestaltung.

b) Der tätigkeitstheorieorientierte Ansatz¹¹

Software wird als Automatisierung von operationalisierbaren Anteilen (innerer) menschlicher Tätigkeit¹² aufgefasst, oder besser als Resultat der Transformation von Anteilen realer bzw. möglicher menschlicher Tätigkeit in eine maschinelle Form, d.h. als Vergegenständlichung von menschlichen Fähigkeiten.

Grundlage für die Softwareentwicklung ist hier die Tätigkeitstheorie. In diesem Sinne könnte man die Tätigkeitstheorie als eine "Grundlagendisziplin" für die Softwareentwicklung ansehen. Es wird danach gefragt: Welche Anteile der Tätigkeit lassen sich in Software transformieren.

c) Der kognitivistische Ansatz

Aus der kognitiven Psychologie heraus entwickelte sich das Denkmuster: Software als Problemlösung, besser als Automatisierung einer Problemlösung oder als Automatisierung (Maschinisierung) eines Algorithmus, der wiederum (selbst) eine Problemlösung repräsentiert.

Das führte u.a. zu GPS (General Problem Solver) und zur KI. Diese Art der Charakterisierung von Software ist heute noch weit verbreitet und wird in der Theoretischen Informatik bevorzugt, da sie explizit auf dem Begriff des Algorithmus basiert.

3. Zwischen diesen Interpretationen kann ein Zusammenhang hergestellt werden.

Alle drei Ansätze haben (aus unterschiedlichen Perspektiven) ihre Berechtigung. Sie stehen aber in einem gewissen Zusammenhang:

- Da Problemlösen eine innere Tätigkeit ist, schließt 2b) 2c) ein.
- Aus tätigkeitstheoretischer Sicht kann eine Zusammenhang zwischen 2a) und 2b) dadurch herstellen werden, dass man mit 2b) die Tätigkeit und mit 2a) den Gegenstand der Tätigkeit beschreibt. Aus dieser Sicht sind beide Ansätze notwendig, indem sie sich ergänzen.
- Andererseits kann man die Tätigkeit selbst zum Gegenstand machen; dann wird 2b) ein Teil von 2a).¹³

Aus dieser Sicht lassen sich die Interpretationen 2b) und 2c) auf 2a) zurückführen.

4. Softwareentwicklung kann als mehrstufiger Transformationsprozeß interpretiert werden.

Hierbei wird vom Modell-Ansatz (siehe 2a)) ausgegangen.

Bei der ersten Transformation (von der realen in die symbolisch virtuelle Welt¹⁴ – siehe Abb. unten) spielt die erkenntnistheoretische Position des "Modellentwicklers" eine wesentliche Rolle, d.h. hier geht es um die Frage: Wie reflektiert er Wirklichkeit als Grundlage für die Modellbildung und damit für die Softwareentwicklung?

Dieser Transformationsprozeß ist nicht zu verwechseln mit einem Vorgehensmodell der Softwareentwicklung. Bei letzterem steht die Organisation der Herstellung von Software im Mittelpunkt. Der Transformationsprozeß charakterisiert primär das Ziel¹⁵, aber noch nicht "wie man das Ziel erreicht". Er legt aber Randbedingungen hierfür fest. Bei der Herstellung von Software können diese

⁷ Naturwissenschaftliches, insbesondere physikalisches Denkmuster, aber auch kybernetisch-systemtheoretisches Herangehen, s. u.a. Dahme, 1997, insbesondere „1 Methodologische Positionen“.

⁸ Unabhängig davon, ob das Modell bewußt oder unbewußt gebildet wurde.

⁹ Das können auch mehrere Disziplinen sein.

¹⁰ Und die Fragestellung, die mit Hilfe von Modellexperimenten beantwortet werden soll.

¹¹ Siehe Dahme/Raeithel 1997. Dieser Ansatz präferiert das OO-Paradigma.

¹² Ein ähnlicher Ansatz bei F. Nake: Software als Maschinisierung von Kopfarbeit. Siehe Coy 92, S. 181-201.

¹³ Siehe Dahme, Ch.: Systemanalyse menschlichen Handelns – Grundlagen und Ansätze zur Modellbildung, Westdeutscher Verlag, Opladen 1997.

¹⁴ Statt „symbolisch virtuelle Welt“ könnte man auch "semiotische Welt" sagen.

¹⁵ Bzw. Teilziele bezogen auf die einzelnen Transformationen.

Transformationen mehrfach in unterschiedlicher Reihenfolge durchlaufen werden. Beim Prototyping werden manchmal sogar Teile der letzte Transformation (im Sinne von Human-Interface-Prototyping) vorgezogen, um diesen Prototyp als Katalysator zur Modellbildung und Spezifikation zu nutzen.

5. Wenn Softwareentwicklung zu einer Dienstleistung¹⁶ wird, dann wird Softwareentwicklung zu einem interkulturellen Prozeß.

Wenn Softwareentwicklung zu einer Dienstleistung wird, stehen sich mindestens zwei Kulturen gegenüber – die der Anwender und die der Entwickler –, deren Vertreten mit einander kommunizieren und gegebenenfalls kooperieren müssen/sollten. Dabei können die unterschiedlichsten Kommunikations-, Kultur- und Koordinationsprobleme auftreten. Das Kriterium "brauchbar"¹⁷ – im Sinne von brauchbarer Software – ist ein typisches Kriterium für diesen interkulturellen Prozeß.

Solche interpersonellen, interkulturellen, kooperativen Prozesse finden in der Regel als Selbstorganisationsprozesse¹⁸ statt. Selbstorganisationsprozesse lassen sich nicht direkt, sondern nur über ihre Randbedingungen beeinflussen.

Fazit: In diesem Sinne kann Softwareentwicklung als ein Transformationsprozeß, der in einen Selbstorganisationsprozeß eingebettet ist, interpretiert werden.

Literatur

Coy, Wolfgang: Sichtweisen der Informatik. Vieweg, Wiesbaden 1992, S. 181-201.

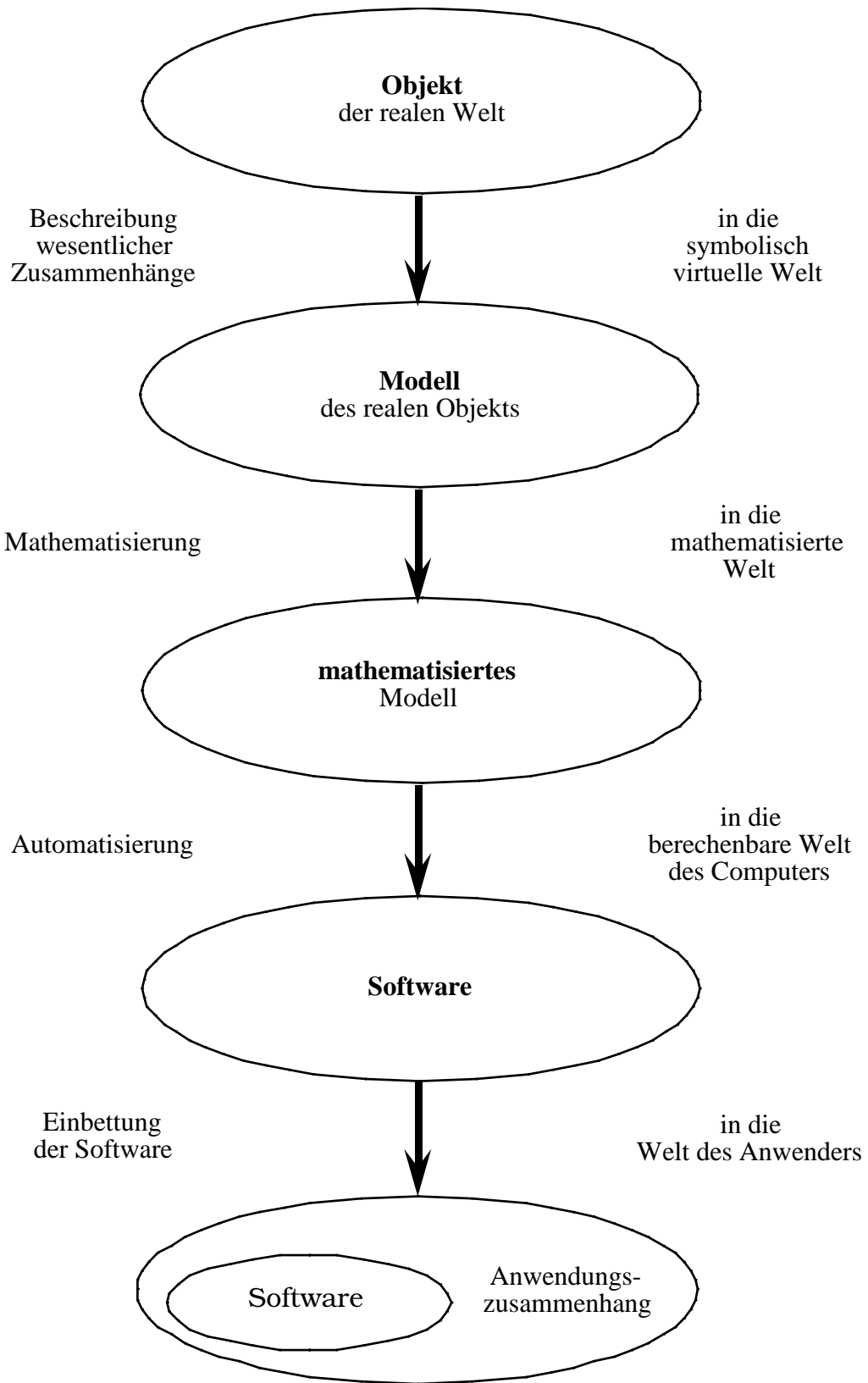
Dahme, Ch.: Systemanalyse menschlichen Handelns – Grundlagen und Ansätze zur Modellbildung, Westdeutscher Verlag, Opladen 1997.

Dahme, Christian, Arne Raeithel: Ein tätigkeitstheoretischer Ansatz zur Entwicklung von brauchbarer Software. In: Informatik-Spektrum 20, Heft 1, Februar 1997, S. 5-12 (<http://waste.informatik.hu-berlin.de/Dahme/Da%2FRa.html>).

¹⁶ Wenn Software ingenieurmäßig/industriell entwickelt und vertrieben werden soll, dann wird Softwareentwicklung zu einer Dienstleistung.

¹⁷ Spiegelt sich insbesondere in der letzten Transformation in 4. wider.

¹⁸ Siehe Dahme 1997, Kap. „5.4 Kooperation und Selbstorganisation“ – erläutert an Hand eines Beispiels.



Anwendungsspiele

Eine Antitheorie der Anwendungen

Martin Fischer, GIPA, Berlin

Im Folgenden wird "Anwendungen" immer im Sinne von informatischen Artefakten, mit denen menschliche Nutzerinnen und Nutzer umgehen, verwendet.

Ludwig Wittgenstein vollzieht vom Tractatus Logico-Philosophicus zu den Philosophischen Untersuchungen eine Wendung von der Betrachtung der Sprache als formales System zum Sprechen als Handeln. Dafür verwendet er den Begriff des Sprachspiels. Damit kommen eine Reihe von Fragen ins Spiel:

- In welchen Handlungskontext sind (sprachliche) Äußerungen eingebettet?
- Was will der/die Sprecher/in mit einer Äußerung in einer bestimmten Situation erreichen?
- Welche unterschiedlichen Handlungen werden mit denselben Äußerungen in unterschiedlichen Situationen vollzogen?
- Nach welchen Regeln, und zwar weniger formalen denn habituellen Regeln (sprich Gewohnheiten), erfolgt sprachliches Handeln?

Für informatische Anwendungen lassen sich ähnliche Fragen formulieren:

- In welchen (organisatorischen) Kontext sind Anwendungen eingebettet?
- Was will der/die Benutzer/in mit/in einer Anwendung in einer bestimmten Situation erreichen?
- Welche unterschiedlichen Handlungen werden mit Aktionen in einer Anwendung in unterschiedlichen Kontexten vollzogen?
- Nach welchen Regeln und Gewohnheiten erfolgt die Benutzung einer Anwendung?

Partizipative Softwareentwicklung und Use-Case-orientiertes Vorgehen sind Ansätze, die den Handlungsaspekt von Software und für die Entwicklung von Software in den Vordergrund rücken.

Daneben will ich noch den Begriff "Handlungswissen" verwenden. Handlungswissen ist im Gegensatz zu propositionalem Wissen implizit und auch nur begrenzt explizierbar und kontextgebunden, vielleicht auch nur ein anderes Wort für die oben verwendeten "Gewohnheiten".

Eine Sprachspielsicht ist insofern hilfreich beim Design von Anwendungen. Sie fragt nach gespielten Sprachspielen; nach der Organisation, nach dem Kontext, in den eine Anwendung eingebettet ist, nach Zielen, nach Verfahren und Abläufen, nach Gewohnheiten und nach dem Handlungswissen der zukünftigen Benutzer/innen.

Eine Sprachspielsicht kann auch erhellend sein, um Anwendungen selbst zu analysieren. Dadurch dass (Handlungs-)Modelle mit Computern (Software, Hardware, Ein-/Ausgabe-geräte) praktisch gemacht werden, entstehen "künstliche" Sprachspiele. Die Sprachspielsicht fragt nach den Sprachspielen, die mit/in der Anwendung gespielt werden, nach den Handlungsmöglichkeiten der Benutzer/innen, wie die Anwendung (als Handlungsraum) gestaltet ist, welche Handlungen einfach zu realisieren, welche kompliziert sind, ob die Benutzer/innen bei ihren Zielen unterstützt werden, ob die Regeln klar sind, ob es einfach ist, Gewohnheiten bei der Benutzung herauszubilden, ob es Mehrdeutigkeiten gibt, ob Benutzer/innen unterschiedliche Sprachspiele mit/in derselben Anwendung spielen.

Schließlich bietet eine Sprachspielsicht eine Reihe von Möglichkeiten, das Scheitern von Softwareprojekten zu analysieren. Sie fragt dann nach dem Verhältnis verschiedener gespielter Sprachspiele.

Softwareprojekte können scheitern, weil das Anwendungsspiel nicht mit der Organisation zusammenpasst, weil das Handlungswissen der Benutzer/innen mit dem Anwendungsspiel kollidiert, wegen zu großer Diskrepanz zu den gewohnten Abläufen oder der vorher benutzten Anwendung, weil das Anwendungsspiel zu schwierig, zu schwierig zu erlernen ist, weil die Ziele nicht zu dem Anwendungsspiel passen.

Softwareprojekte können scheitern, weil die Entwicklersprachspiele und die Benutzersprachspiele zu sehr kollidieren, weil die organisatorischen Anpassungen an das Anwendungsspiel unterschätzt werden, weil die Benutzer/innen das Anwendungsspiel anders spielen als von den Auftraggebern oder Entwicklern intendiert, weil die Anwendung für einen anderen Kontext entwickelt wurde, als dem, in dem es eingesetzt wird oder werden soll.

Solchen Problemen lässt sich mit einer handlungsorientierten Sicht, mit einem Sprachspielmodell für Anwendungen nachgehen.

Auch das Problem von Wissen und Gegenwissen scheint mir hauptsächlich darin zu bestehen, dass es unterschätzt wird, kontextgebundenes Handlungswissen in einen anderen Kontext zu transferieren, nicht nur weil es schwierig ist implizites Wissen zu explizieren, sondern auch weil das Wissen im Zielkontext, wo andere Sprachspiele gespielt werden, nicht verstanden, genauer, nicht genutzt werden kann.

Anwendungen als Sprachspiele und als eingebettet in Sprachspiele zu verstehen, liefert keine Theorie der Anwendungen. Das Sprachspielkonzept von Wittgenstein kann eher als Methode gesehen werden, Einzelfälle zu untersuchen und Unterschiede zwischen vermeintlich ähnlichen Fällen herauszuarbeiten. Wenn eine Theorie etwas aussagt über Allgemeines, über Gemeinsamkeiten, dann können Sprachspiele als ein antitheoretisches Konzept gelten.

Wer nicht nach der Softwarenutzung fragt, bleibt dumm

Bernd Pape, Universität Hamburg

Ausgangspunkt meiner Überlegungen in diesem Beitrag sind Diskrepanzen zwischen den Erwartungen an die Nutzung von Anwendungssoftware und den dabei erzielten Erfolgen. In der Literatur wird häufig über widersprüchliche Wirkungen von Computeranwendungen berichtet (vgl. bspw. Kling 1996). Diese Berichte finden eine Zuspitzung in dem sogenannten Produktivitätsparadoxon der Informationstechnologie (vgl. bspw. Potthof 1998). Und auch in zwei Projekten, in denen ich in den vergangenen vier Jahren als wissenschaftlicher Berater tätig war, war das Missverhältnis zwischen Erwartungen und Erfolgen beim Einsatz von Anwendungssoftware wahrzunehmen bzw. sogar Ausgangspunkt für mein Engagement. Im folgenden skizziere ich zunächst meine Erfahrungen in diesen beiden Projekten. Anschließend stelle ich die groben Züge des Interpretationsrahmens vor, mit dem ich meine Erfahrungen reflektiere und expliziere und den ich auch gestalterisch anwenden kann. Dieser Interpretationsrahmen fokussiert auf typische Formen im alltäglichen Handeln mit Software und mit ihnen verbundenen normierenden Einschreibungen.

Zwei Fallbeispiele – wer, wie, was?

In dem Projekt Prüfungsverwaltung – kurz: ProPrüVer – haben eine Kollegin, drei Kollegen und ich über drei Jahre lang die Einführung einer Prüfungsverwaltungssoftware in einer großen deutschen Universität beratend begleitet (vgl. Klischewski 2001). Die Softwareeinführung geschah im Rahmen einer Bemühung zur Reorganisation der Prüfungsverwaltung. Diese umfasste die Dezentralisierung von drei Prüfungsämtern. Sie sollten aufgelöst und stattdessen in den einzelnen Fachbereichsverwaltungen Stellen für Prüfungsangelegenheiten gebildet werden. Die Prüfungsverwaltungssoftware sollte dabei unterschiedliche Personengruppen, Studierende bzw. Prüflinge, Lehrende bzw. Prüfende sowie Verwaltungsmitarbeiterinnen und -mitarbeiter insbesondere in der Abwicklung von Verfahren unterstützen, die über institutionelle Grenzen hinweg verlaufen. Die Anmeldung zu einer Prüfung im Nebenfach stellt ein Beispiel dafür dar. Im Rahmen der Softwareeinführung haben wir, das Beratungsteam, zunächst die Auswahl eines geeigneten Softwaresystems begleitet. Der langwierige, etwa einjährige Auswahlprozess offenbarte, dass es nicht das eine richtige System gab, sondern jedes System mit seinen Stärken und Schwächen jeweils andere Konsequenzen in seiner Bereitstellung und der Arbeit mit ihm nach sich ziehen würde. Daher haben wir im Anschluss an die Auswahlentscheidung als Berater darauf gedrungen, die Voraussetzungen für eine erfolgreiche Nutzung zu schaffen. Folgende drei Handlungsfelder haben wir dafür benannt: Projektmanagement, damit die beteiligten Akteure sich über die Teilziele und die dafür erforderlichen Schritte abstimmen und entsprechend klar abgegrenzte Aufgaben bzw. Verantwortlichkeit übernehmen; Organisationsentwicklung zur Qualifizierung und Betreuung der betroffenen Mitarbeiterinnen und Mitarbeiter sowie zur Umstellung bestehender Verwaltungsabläufe; technischer Betrieb im Sinne der Bereitstellung von geeigneter Hard- und Software an zentraler Stelle und an allen Arbeitsplätzen inklusive ihrer Vernetzung und notwendiger Maßnahmen zur IT-Sicherheit und zum Datenschutz. Doch auch fast ein Jahr später kamen wir zur Feststellung, dass eine Reihe notwendiger Voraussetzungen für einen erfolgversprechenden Projektfortschritt nicht gegeben waren. Aus unserer Sicht war das Konzept der Systemeinführung nicht genügend mit den beteiligten Akteure abgestimmt. Im Versuch der Kooperation dominierten stets Teilaspekte wie Planstellen, Dienstvereinbarungen, Computerausstattung, Datenschutz, Kabelschächte, Firewalls etc., wobei die Akteure jeweils ihr Partikularinteresse gegenüber dem Fortschritt insgesamt in den Vordergrund stellten. Das Projektgeschehen drehte sich daher mehr um die Konfliktbewältigung der Entscheider als um die Menschen, die mit der neuen Prüfungsverwaltungssoftware unterstützt werden sollten. Als Fazit aus diesem nicht gerade glücklichen Projektverlauf ziehe ich, dass die Art und Weise, wie eine Anwendungssoftware genutzt wird, weniger von der Software selbst abhängt, als von dem Mit- und Gegeneinander der beteiligten Akteure.

Die zweite Fallstudie, die Fallstudie CommSy, handelt von der Entwicklung und Nutzung eines web-basierten Softwaresystems in und für universitäre Lerngemeinschaften. CommSy steht für Community System und wird seit Mai 1999 in den Arbeitsgruppen von Arno Rolf und Christiane Floyd im Fachbereich Informatik, Universität Hamburg unter Beteiligung einer Reihe von Studierenden und wissenschaftlichen Mitarbeiterinnen und Mitarbeitern entwickelt und seit März 2001 in dem Forschungsprojekt WissPro weiterentwickelt (vgl. Bleek et al. 2002; <http://www.wisspro.de>). Auch ich war und bin weiterhin daran beteiligt. CommSy unterstützt die Kommunikation in kleinen bis mittelgroßen Arbeits- bzw. Lerngemeinschaften (vgl. Pape et al. 2002). In diesem Fallbeispiel ging und geht es mir um die Wechselwirkungen zwischen der Entwicklung von CommSy und dem Einsatz in

Lehrveranstaltungen. Mit der Erfahrung aus der Fallstudie ProPrüVer, dass das Vorhandensein einer Anwendungssoftware alleine nicht unmittelbar zu seiner Nutzung und damit auch nicht zur Unterstützung von Arbeit in organisatorischen Zusammenhängen führt, konzentriere ich mich in diesem Anwendungskontext, hier in einer Lerngemeinschaft, darauf, mögliche Maßnahmen zu erproben, die die Benutzung einer Anwendungssoftware, hier von CommSy, unterstützen sollen. Das soll nicht in ein "Rezept mit Erfolgsgarantie" münden, sondern es geht mir darum, eine Vielfalt möglicher Maßnahmen sowie ihre situative Kombination zu erproben. Als Fazit aus diesem zweiten Fallbeispiel ziehe ich, dass die zur Unterstützung der Systembenutzung ergriffenen Maßnahmen für alle Beteiligten zusätzlichen Aufwand bedeuteten. Doch die Maßnahmen sind notwendig, damit das System überhaupt genutzt wird. Oder andersherum: Wer nicht bereit ist, den zusätzlichen Aufwand zu tragen, sollte das System gar nicht erst einsetzen.

Interpretationsrahmen – wieso, weshalb, warum?

Der Begriff "Nutzung" umfasst für mich sowohl das Benutzen als auch das Ausnutzen von Anwendungssoftware. Be- und Ausnutzen stehen in einem rekursiven Verhältnis zueinander: Das eine geht ohne das andere nicht und umgekehrt. Software lässt sich nur ausnutzen, wenn sie benutzt wird, und umgekehrt wird sie benutzt, um sie auszunutzen. Dieses scheinbar triviale, wechselseitige Verhältnis stellt sich durchaus vertrackt dar, wenn ich den Begriff ‚Nutzung‘ vom unmittelbaren Umgang mit Software auf mittelbar damit zusammenhängendes Handeln ausweite. Das Ausnutzen von Anwendungssoftware ist dann in einer Vielzahl von unterstützten Tätigkeiten zu finden und die Benutzung erst durch eine Vielzahl unterstützender Praktiken möglich. Mit dieser Ausweitung verlieren eindeutige Fragen und Antworten an Bedeutung. Es geht nicht mehr darum, wie ist ein Softwaresystem "richtig" zu benutzen, um ein Anwendungsproblem "korrekt" zu lösen, sondern es stellen sich unscharfe Fragen, und es gibt nur bedingte Antworten. Dann hängt das, wofür Anwendungssoftware genutzt werden kann, davon ab, was für die Nutzung getan wird, und das, was für die Nutzung von Anwendungssoftware getan wird, hängt davon ab, was mit ihr erreicht werden soll. Mit dieser Wendung verliert das Verhältnis zwischen Be- und Ausnutzen, verliert die Nutzung von Anwendungssoftware ihre Eindeutigkeit, und es zeigt sich ihre Kontingenz. Die Nutzung von Anwendungssoftware ist stets auch anders möglich. Damit drängen sich unmittelbar folgende Fragen auf: Wie kommt es überhaupt zur Nutzung von Anwendungssoftware und wie zur einer bestimmten Art der Nutzung? Lässt sich die Nutzung beeinflussen, und wenn ja, wie? Sowie: Welche Rolle spielt dabei die Anwendungssoftware und die Art ihrer Gestaltung?

Konkret: Wenn Anwendungssoftware für organisatorisches Handeln ausgenutzt werden soll, soll sie als Arbeits- und Kommunikationsmittel bestehende oder neue Handlungsweisen von Menschen in Organisationen unterstützen. Das Benutzen von Software, beispielsweise um Briefe oder Aufsätze zu schreiben, um Datensätze, Dokumente oder Archive zu ordnen und zu durchsuchen oder um per Email mit anderen zu kommunizieren, soll das Arbeitsleben von Menschen erleichtern oder mit zusätzlichen Handlungsoptionen anreichern. Darüber hinaus werden Anwendungssoftware organisatorische Potentiale zugeschrieben. Denn mit der Nutzung von Anwendungssoftware werden nicht nur Operationen von Menschen auf Computer übertragen, sondern sie führt fast immer auch zu einer Tätigkeitsverlagerung zwischen den beteiligten Menschen und damit zu einem Wandel der Abläufe und Strukturen in Organisationen. Diese organisatorischen Potentiale werden so hoch eingeschätzt, dass die Entwicklung neuer Organisationsformen derzeit ohne Einbeziehung von Anwendungssoftware kaum denkbar ist. Das bringen Organisationskonzepte wie "Business Reengineering", "virtuelle Organisation", "E-Business" oder "Wissensmanagement" pointiert auf die Tagesordnung von Forschung und Lehre in Informatik und Wirtschaftsinformatik (vgl. Rolf 1998).

Doch das Vorhandensein von Anwendungssoftware in Organisationen führt nicht ohne Weiteres zu ihrer Benutzung und auch nicht zu ihrer Ausnutzung, also nicht zur Unterstützung der Arbeit in Organisationen. Es können diverse Probleme dabei auftreten, die Benutzung sowie die damit verbundenen Handlungsweisen und Organisationsformen anfänglich um- und dauerhaft fortzusetzen, um von einem Ausnutzen sprechen zu können. Susanne Maaß (1993, 191) drückt das sehr anschaulich aus: "Der Umgang mit Anwendungsprogrammen ist im Detail schwer zu lernen, die Funktionalität unzureichend und undurchschaubar, einfache Vorgänge werden durch die Einführung von Computern kompliziert, und die erhofften Vorteile und Arbeitersparnisse sind zunächst nicht auszumachen. Zudem werden bewährte und eingespielte Arbeitsabläufe beeinträchtigt, oder sie müssen gänzlich umgestellt werden; neue Qualifikationen sind gefordert und alte nicht mehr gefragt."

Ich wähle zwei Ansatzpunkte zur Betrachtung der rekursiven Beziehung zwischen Ausnutzung und Benutzung von Anwendungssoftware – erstens: Damit Anwendungssoftware anfänglich und dauerhaft so benutzt wird, dass sich eine Unterstützung von Handlungsweisen in Organisationen daraus ergibt, bedarf die Benutzung der Anwendungssoftware selbst Unterstützung. Mein Ziel dabei ist es, mögliche

Maßnahmen zur Unterstützung der Benutzung von Anwendungssoftware sowie Kriterien zu ihrer situativen Kombination zu erarbeiten. Ziel dabei soll es sein, die Benutzerinnen und Benutzer in ihrer konkreten Arbeitssituation dazu zu befähigen, mit Anwendungssoftware sinnvoll umgehen zu können. Zum sinnvollen Umgang gehört für mich, Anwendungssoftware nicht nur bedienen, sondern diese auch gezielt für verschiedene Zwecke einsetzen und ausnutzen zu können. Dafür spielt auch das Erlangen einer fundierten Urteilsfähigkeit über die Bedingungen für die Nutzung von Anwendungssoftware sowie über ihre Auswirkungen eine zentrale Rolle.

Um den sinnvollen Umgang mit Anwendungssoftware zu charakterisieren, wähle ich einen zweiten Ansatzpunkt: Für die Nutzung von Anwendungssoftware sind die beteiligten Menschen stets auf andere, auf die Zusammenarbeit mit anderen, auf kollektives Handeln angewiesen. Sowohl das Bereitstellen eines Softwaresystems als auch seine Integration in Arbeitszusammenhänge erfordert, dass etliche Personen sich untereinander darüber abstimmen und ihr Handeln daraufhin koordinieren, wann, wo und wie die Anwendungssoftware zu benutzen ist. Gerade das (Nicht-)Gelingen der notwendigen Koordination kann dazu führen, dass Anwendungssoftware in Organisationen (nicht) sinnvoll genutzt wird. Beim Aufstellen von Kriterien zur situativen Kombination von Maßnahmen der Benutzungsbetreuung ist die dafür notwendige, fortlaufende Koordination zwischen den beteiligten Akteuren für mich richtungsweisend.

Zur Verknüpfung der beiden Ansatzpunkte meines Interpretationsrahmens konzentriere ich mich auf typische Formen im Handeln der Beteiligten (vgl. Yates/Orlikowski 1992; Orlikowski/Yates 1994; Yates et al. 1999). Handeln heißt für mich, in die Welt einzugreifen, mit dem Effekt, den Lauf der Dinge zu beeinflussen, oder aber dieses Eingreifen zu unterlassen (vgl. Giddens 1995, 51ff.): Was auch immer geschieht, es wäre nicht geschehen, wenn ein Individuum nicht in den Lauf der Dinge eingegriffen hätte. Damit setzt Handeln das Wissen und die Autonomie voraus, eingreifen und einen Unterschied im Lauf der Dinge machen zu können. Handeln ist jedoch nicht beliebig und die notwendige Autonomie begrenzt. Die Akteure können sich weder vollständig von dem Lauf der Dinge trennen, der vor ihrem Eingreifen stattgefunden hat, noch von der Tatsache, dass sie nicht als einzige in den Lauf der Dinge eingreifen. Damit hängt auch zusammen, dass das Wissen der Akteure über ihr Handeln begrenzt ist. Sie können nicht alles wissen, was zuvor geschehen ist oder gleichzeitig geschieht. Dennoch sind sie mit den komplexen Geschehnissen und Zusammenhängen um sie herum so vertraut und ihnen gegenüber – außer in zwanghaften, lebensbedrohenden Situationen – so autonom, dass sie sich in ihnen zurechtfinden und einen steuernden Einfluss auf sie ausüben. Für meine Arbeit beziehe ich mich auf zwei Prinzipien, mit denen Menschen diese Leistung fortlaufend vollbringen (vgl. Giddens 1995, 51ff.; Ortmann 1995, 49ff.; Treibel 1997, 237f.): Zum einen routinisieren sie ihr Handeln. Den meisten Handlungen haftet ein gewisser Wiederholungscharakter an. Damit gelangt eine gewisse Kontinuität in den menschlichen Alltag. Zum anderen orientieren sich Menschen am erwarteten Handeln anderer – und erwarten umgekehrt diese Orientierung auch von anderen. Dazu versuchen sie, Regelmäßigkeiten in sozialen Gefügen auszunutzen, sie zu erfassen und zu pflegen. An diesen Regelmäßigkeiten können Handelnde ihre Erwartungen gegenüber anderen ausbilden. Während im alltäglichem Handeln die Routinisierung und Bezugnahme auf andere häufig unbewusst und beiläufig geschehen, gilt ihnen in Organisationen gezielte Aufmerksamkeit. Auch wenn beides dennoch nicht immer gelingen mag, sind typische Formen im Handeln Ansatzpunkte zur wechselseitigen Bezugnahme der beteiligten Akteure.

Als Beispiele regelmäßig verwirklichter Handlungstypen für die Softwarenutzung können etwa Arbeitssitzungen, Präsentationen, Schulungsseminare, das Schließen von Verträgen, das Schreiben und Lesen von Projektberichten oder auch von der Dokumentation sowie das Bedienen der Software gelten. Sie lassen sich in verschiedenen Handlungsfeldern, die für die Nutzung relevant sind, ausmachen wie beispielsweise in der Entwicklung, Einführung, Anpassung oder Benutzung der Software ebenso wie in der Veränderung des organisatorischen Nutzungskontextes. Um verschiedene Handlungstypen bestimmen zu können, sind Unterscheidungsmerkmale für sie herauszuarbeiten. Zu ihrer Unterscheidung können bspw. der Anlass und das Motiv des Handelns, die Teilnehmer und ihre sozialen Rollen, die Form und das Medium einer Interaktion herangezogen werden. Darauf aufbauend lassen sich tatsächliche und mögliche Ausprägungen von Handlungsformen in Nutzungssituationen erörtern. Neben beispielhaften Merkmalsausprägungen lässt sich darüber hinaus darstellen, wie die typischen Handlungsformen im Verlauf der Softwareeinführung durch das Handeln einzelner Akteure geprägt werden und wie sie umgekehrt die Interaktion zwischen den Akteuren prägen können. Dadurch wird Handeln für die Nutzung von Anwendungssoftware und auch damit verbundene normierende Einschreibungen darstellbar und auch beeinflussbar.

Literatur

- Bleek, W.-G.; Krause, D.; Oberquelle, H.; Pape, B. (Hrsg.) (2002): Medienunterstütztes Lernen – Beiträge von der WissPro-Wintertagung 2002. Berichte des Fachbereichs Informatik der Universität Hamburg, FBI-HH-B-239/02
- Giddens, A. (1995): Die Konstitution der Gesellschaft, 3. Auflage, Frankfurt a.M./New York.
- Kling, R. (1996): Computerization and controversy: value conflicts and social choices, 2. Ausgabe, San Diego u.a.
- Klischewski, R. (2001): Commitments Enabling Co-operation in Distributed Information Systems Development. In: Proceedings 9th European Conference on Information Systems (ECIS 2001, Bled, Slovenia, June 27-29), pp. 509-519.
- Maaß, S. (1993): Software-Ergonomie. In: Informatik-Spektrum, Jg. 16, 191 – 205.
- Orlikowski, W. J., Yates, J. (1994). Genre Repertoire: The Structuring of Communicative Practices in Organizations. Administrative Science Quarterly, 39 (December), 541-574.
- Ortmann, G. (1995): Formen der Produktion, Opladen.
- Pape, B., Bleek, W.-G., Jackewitz, I., Janneck, M. (2002): Software Requirements for Project-Based Learning – CommSy as an Exemplary Approach. In: Proceedings of the 35th Hawaii International Conference on System Sciences – 2002.
- Potthof, I. (1998): Empirische Studien zum wirtschaftlichen Erfolg der Informationsverarbeitung. In: Wirtschaftsinformatik, Jg. 40, Nr. 1, 54 – 65.
- Rolf, A. (1998): Grundlagen der Organisations- und Wirtschaftsinformatik, Berlin/Heidelberg.
- Treibel, A. (1997): Einführung in soziologische Theorien der Gegenwart, 4. Auflage, Opladen.
- Yates, J., Orlikowski, W. J. (1992). Genres of Organizational Communication: A Structural Approach to Studying Communication and Media. Academy of Management Review, 17 (2), 299-326.
- Yates, J., Orlikowski, W. J., Okamura, K. (1999): Explicit and Implicit Structuring of Genres in Electronic Communication: Reinforcement and Change of Social Interaction. In: Organization Science, Vol. 10, No. 1, pp. 83 – 103.

Informatiksysteme in Organisationen

Vorarbeiten für eine Theorie der Anwendungen der Informatik

Arno Rolf, Universität Hamburg

Softwaretechnik, Software-Ergonomie, Wirtschaftsinformatik und einige andere Bindestrich-Informatiken, z.B. die Verwaltungsinformatik, haben Organisationen und Unternehmen, mittel- oder unmittelbar, im Blick. Ihre Produkte, Modelle und Methoden sind letztendlich für diese Gegenstandsbereiche gedacht, dafür sind ökonomisch verwertbare Lösungen zu entwickeln. Hierin liegt ihre Gemeinsamkeit, es geht dabei um Informatiksysteme für bzw. in Organisationen.

Die Teilgebiete der praktischen Informatik und Bindestrich-Informatiken haben sich im Laufe ihres Bestehens ausdifferenziert. Nach Luhmann zeichnen sich fortschrittliche Gesellschaften durch permanente Komplexitätserhöhung bei gleichzeitiger funktionaler Differenzierung aus. Für die Teilgebiete wie für alle Ausdifferenzierungen gilt: Gemeinsames, Koordinationen und Kooperation bleiben dabei leicht auf der Strecke. Die Informatiker in diesen Teilbereichen werkeln eher nebeneinander her. Sie sind zumeist nur begrenzt über die jeweils angrenzenden Forschungsthemen informiert.

Ein Bedarf nach Koordination, möglicherweise sogar nach Schärfung einer gemeinsamen Perspektive, könnte sich aus der Arbeit am gemeinsamen Objekt ergeben: Bislang ist die bestehende Arbeitsteilung kaum transparent, keiner weiß so recht, ob sie Sinn macht, ob möglicherweise Doppelarbeiten und Widersprüche auftreten. Mehr Transparenz könnte sinnvolle Schnittstellen und Übergabepunkte sichtbar machen. Die Forschungs- und Entwicklungsprozesse würden sich produktiver gestalten.

Um die gewünschte Transparenz zu erreichen, ist es erforderlich, sich mit den *impliziten Annahmen* der Teilgebiete der angewandten Informatik auseinanderzusetzen. (Der Einfachheit halber soll im folgenden auch die praktische Informatik, mit ihren auf Organisationen abzielenden Produkten, unter diese Metapher fallen, wohlwissend, dass dies nicht mit der informatischen Tradition im Einklang steht.) Ein *Klassifikationsversuch* der verschiedenen Ansätze könnte, über implizite Annahmen hinaus, Arbeitsteilung, Doppelarbeiten, Konflikte und Widersprüche, Übergabepunkte, vielleicht auch insgesamt Defizite der verschiedenen Ansätze sichtbar machen.

Gemeinsam ist dem Gegenstandsbereich "Informatiksysteme in Organisationen" darüber hinaus, dass der erfolgreiche Einsatz von Informationstechnik nicht zuletzt davon abhängt, ob tragfähiges Wissen über Organisationen und soziale Systeme, über das Verhalten von Akteuren sowie über ökonomische Grundtatbestände bei Wissenschaftlern und Systemgestaltern vorhanden ist und in die Arbeiten eingehen. Es geht also auch um die Berücksichtigung ökonomischer, sozial- und arbeitswissenschaftlicher Erkenntnisse. Entsprechendes Wissen könnte dort hilfreich sein, wo offene Fragen, Schwachpunkte, Widersprüche und Konflikte bei den einzelnen Ansätzen auftauchen.

Dies ist ein Einstieg in eine interdisziplinäre Sichtweise der angewandten Informatik. Der Vorwurf, die interdisziplinäre Sicht sei eine Überforderung der angewandten Informatik, ist nur begrenzt haltbar: Die anwendungsnahen Teilgebiete sind auch heute schon auf eine subtile Weise interdisziplinär. Sie hantieren zumeist mit mehr oder minder plausiblen Alltagswissen. Aufgrund ihrer Attraktivität und der damit verbundenen Diffusionsstärke ihrer Produkte definieren sie Praxis allerdings viel stärker als andere, z.B. sozial- und arbeitswissenschaftliche Disziplinen, die dazu wahrscheinlich Differenzierteres zu sagen hätten. Insofern verspricht eine abgesicherte Interdisziplinarität eine Qualitätsverbesserung sowohl der angewandten Informatik wie der Praxis.

Am Ende könnte so etwas wie der Versuch stehen, Anforderungen für eine gemeinsame Plattform "Informatiksysteme in Organisationen" zu formulieren und erste Pfeiler für ein Modellkonzept zu skizzieren. Es sollte dabei um Sinnhaftigkeit, Machbarkeit und Voraussetzungen einer disziplinübergreifenden Sicht "Informatiksysteme in Organisationen" unter Wahrung der Identität der praktischen Informatik wie der Bindestrich-Informatiken gehen. Motivation sollte das Gelingen erfolgreicher technischer wie gesellschaftlicher Innovationen sein.

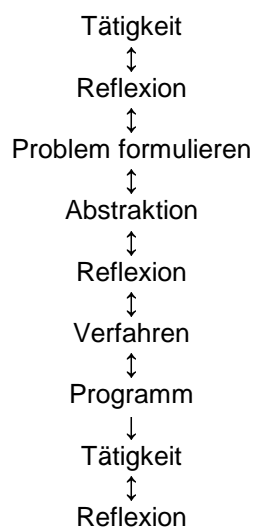
Theorie der Anwendung oder Anwendung der Theorie

Peter Schuck, Wien

Die Informatik beschreibt den Prozess der Software-Entwicklung klassisch im Dreischritt "Problem - Algorithmus - Programm". Dies hat historische Ursachen. Mit der Erfindung des Computers stand erstmals eine universelle Maschine zur Verfügung, und nun ging es darum, Anwendungsmöglichkeiten für diese Maschine zu finden und umzusetzen. Der Fokus ist heute ein anderer. Und das hat Auswirkungen auf den og. Dreischritt: Zu allererst lässt sich einwenden, dass der Schritt vom Problem zum Algorithmus ungleich viel grösser ist als der vom Algorithmus zum Programm. Waren vor 30 Jahren die fehlenden Werkzeuge zur Übertragung einer mehr oder weniger formalen Beschreibung in ein Programm noch ein wesentliches Hindernis in der Software-Entwicklung und insofern die Probleme im Schritt vom Algorithmus zum Programm technischer Art, so sind die Schwierigkeiten bei der Entwicklung von Algorithmen die gleichen wie eh und je. Während also der zweite Schritt vereinfachend gesagt die Übertragung von einer formalen Beschreibung in eine andere darstellt, befasst sich der erste Schritt mit Übersetzung eines unstrukturierten Problems in eine formale Beschreibung eines Lösungsansatzes, also dem schöpferischen Prozess der Lösungsfindung.

Doch im Schritt vom Problem zum Algorithmus ist die Lösungsfindung selbst nicht nur grob vereinfacht dargestellt, sondern ihre Beschreibung verstellt den Blick auf die zur Software-Entwicklung unbedingt nötigen Softskills.

Der Prozess der Software-Entwicklung beginnt längst *vor* dem Problem mit der Tätigkeit/Arbeit (Marx) und der *darin* einsetzenden Reflexion *darüber*. Diese Reflexion selbst ist schon Abstraktion, sie findet mittels Sprache ihren Ausdruck. Hier wird das Problem bewusst, doch selbstverständlich zunächst nicht in seinem ganzen Ausmass. In einer Wechselwirkungskette von Tätigkeit ↔ Reflexion ↔ Problemformulierung wird das Problem immer klarer. Und erst an diesem Punkt beginnt die klassische Informatik über das Problem nachzudenken: "Problemanalyse". Im Prozess der Formalisierung (immer begleitet von der Reflexion) erscheint das Problem immer deutlicher und seine (möglicherweise) algorithmische Lösbarkeit. Entweder entsteht hieraus ein Algorithmus oder der Prozess der Formalisierung muss abgebrochen werden. Vom Algorithmus zum Programm ist dann bekanntlich ein Leichtes. Wird dieses eingesetzt, so verändert es die Tätigkeit der davon Betroffenen, und der Prozess der Reflexion kann auf einer erweiterten Stufe sich fortsetzen. Es ergibt sich dann folgende Kette von Prozessschritten:



Die Reflexion hat in dieser Prozesskette eine zentrale Funktion. Es war eine der zentralen Forderungen der Usability-Diskussion, die AnwenderInnen und nicht nur die Entscheider mit in den Prozess der Software-Erstellung zu integrieren. Mit ihnen gemeinsam soll dabei die Tätigkeit reflektiert und analysiert werden, um darüber zu einem gemeinsamen Verständnis des zu lösenden Problems zu kommen und gemeinsam festzulegen, welche Funktionalität die Software erfüllen soll. Dies verändert das Verhältnis zur eigenen Arbeit nicht nur der AnwenderInnen; auch auf die InformatikerInnen kommen ganz andere Anforderungen: es geht vor allem um sprachliche Ausdrucksfähigkeit gepaart mit der Fähigkeit, die eigene Tätigkeit ständig zu überdenken und in Frage zu stellen: Selbstkritik.

Hier wäre die praktische Seite der "Reflexiven Informatik" [2, S. 21] zu verorten. Letztere verkommt ohne eine solche schnell zur bloßen Nabelschau. Die Theorie der Informatik muss also, will sie dem entgegen, auf die gegenwärtige Praxis der Informatik reflektieren und damit auch auf die gesellschaftlichen Bedingungen, unter denen diese Praxis sich vollzieht. Und um in der Reflexion der Informatik-Praxis gleich anzuschließen, möchte ich den Fokus auf einige Punkte der gegenwärtigen Praxis in der Software-Entwicklung richten:

Praxis findet ohne Theorie statt

Software wird in Wirklichkeit ganz anders produziert, als es die Disziplin und die daran geschulte Einsicht für vernünftig erachtet:

- Code-Reuse erfolgt durch Copy-und-Paste
- Wichtige Design-Entscheidungen werden noch im späten Projektprozess umgestoßen oder überhaupt erst getroffen
- Kundenbedürfnisse werden im Verkaufsprozess nicht befriedigt, sondern produziert
- Es wird die Oberfläche mit vielen zumeist ungenutzten, folglich unnötigen Features verkauft; Funktionalität und Usability sind nur zweitrangige Kriterien
- Die EntscheiderInnen der KundInnen sind nicht die AnwenderInnen der Software und kennen zumeist die Praxis der AnwenderInnen nicht
- Eine Reflexion der eigenen Tätigkeit sowohl von AnwenderInnen wie von InformatikerInnen ist nicht vorgesehen
- Software wird als fertig präsentiert, obwohl erst ein Prototyp existiert
- Aus dem Prototyp wird die fertige Version entwickelt
- Für Programm-Dokumentation gibt es weder Geld noch Zeit
- Tests finden in der Regel nicht statt
- Software-Fehler und erst recht deren Folgen werden auf die KundInnen abgewälzt
- Üblicherweise existiert kein definierter Produktionsprozess¹

Die Krise der Informatik ist nicht die Software-Krise

Die Frage, wieso sich die Praxis nicht an die Theorie hält, ist nun eben kein Problem der Praxis. Die Mittel für Y2K und die Euro-Umstellung wurden bereitwillig locker gemacht. Die Einsicht, dass diese Kosten nur einen Bruchteil ausgemacht hätten, wenn nicht vor Jahren und Jahrzehnten Design- und Entwicklungsfehler gemacht worden wären, führt heute keineswegs zu einer geänderten Softwareproduktion.

Das Papierlose Büro ist deshalb nicht Wirklichkeit geworden, weil die AnwenderInnen genau wissen, dass von einer Parnas'schen Vertrauenswürdigkeit keine Rede sein kann, und sich deshalb immer noch Sicherungs- und organisatorische Umgehungslösungen offen halten. Sie werden durch die Software täglich von der Richtigkeit dieser Maßnahmen bestätigt. Würden Architekten in gleicher Qualität Gebäude konstruieren, wären einstürzende Neubauten an der Tagesordnung.

Theorie ohne Praxis?

Die drängendste Frage, der sich eine Theorie der Informatik stellen muss, ist nicht die von einem Selbstverständnis als Irgendwas-Wissenschaft, sondern die, wieso die Methoden der Software-Entwicklung nicht in dem Maße in die Praxis umgesetzt werden, wie es vernünftig erscheint. Diese Frage kann nur die Theorie beantworten, weil die Praxis sie nicht zu stellen vermag.

Damit geht es aber um die Untersuchung der Auswirkungen der Bedingungen, unter denen Software-Produktion stattfindet, auf diese selbst. Die Bedingungen sind die der kapitalistischen Produktionsweise. Folglich geht es darum zu zeigen, wie der Produktionsprozess von dem Primat der Verwertung bestimmt wird und welche Auswirkungen das einerseits auf die Arbeit der InformatikerInnen selbst, andererseits auf die Art und die Qualität ihrer Produkte hat.

Dann wird klar, dass nicht nur die Produktionsmittel, sondern auch die Produktivkräfte selbst durch die Produktionsweise der Gesellschaft präformiert sind.²

Abschließen möchte ich in Anlehnung an Theodor W. Adornos *Minima Moralia* [1] mit vier Miniaturen zum Bereich der Informatik aus den Jahren 1991, 1998 und 2002.

¹ RUP u.dgl. werden selten eingesetzt – ISO 9000ff. ist in der Softwareproduktion gescheitert.

² Es wäre ein grobes Missverständnis, würde diese radikale Kritik mit Technikfeindlichkeit verwechselt. Peter Bittner hat im Anschluß an Habermas in [2, S. 23] vorgeführt, wie eine zu frühe Versöhnung mit der gegenwärtigen gesellschaftlichen Praxis der eigenen Kritik die Spitze nimmt.

Minima Informatica

Fensterln. Die modernen Computersysteme sind, so heißt es, benutzerfreundlich geworden. Die Symbole seien leichter verständlich und einfacher zu handhaben als zentnerschwere Manuale, in deren Gewicht sich die gesamte Entwicklung der Technologie niederzuschlagen scheint. Ohne überflüssigen Ballast sofort an die Arbeit gehen zu können, ist der Vorteil jener Technologie, die sich auf der einen Seite durch ein Nicht-Denken-Müssen und auf der anderen durch sofortige Verwertung der Arbeitskraft inbarer Münze bezahlt macht. Unwiederbringlich verloren wird mit jedem Blick durchs Fenster auf die Welt, welche selbst mehr und mehr zur Maschine und deren Scheinwelt sich beschränkt, das, was Musil noch den Realitätssinn nannte. Möglich ist alles und wirklich damit zugleich nichts. Der Blick durchs Fenster, zu oft schon als Beleg herbeizitiert, geht ins Leere. Fern jeder sinnlichen Wahrnehmung ist das, was es da zu sehen gibt. Und doch wird jene und mit ihr das Bewusstsein davon affiziert, so es nicht schon völlig gefangen darin verharrt. Wer einmal die Gelegenheit gehabt hat, Computer-Spieler, die sich heute in vielen Kinderzimmern finden, in ihrem Spiel zu betrachten, der weiß, welch grausames Geschäft da getrieben wird: die Welt ist in der Maschine und damit schon völlig negiert. Es bleibt nur noch ein gradueller Unterschied zwischen dem tausendfach geübten Massenmord des Spiels und dem Genozid.

Von Mäusen und Menschen. Die Maus, einstmals Schreckgespenst der Frauen in männlich-dummen Stammtischwitzen und jener, welche sich nach diesem Bild formen ließen, soll heute zur Vertrauten aller Sekretärinnen werden. Die Erziehungsleistung, die damit verbunden ist, wäre wesentlich bedeutender, säße da ein solches Tier. Aber auch das an die Maschine genabelte Plastikgehäuse wird gezähmt. Von der ersten Berührung über die geplante Verwendung jenes Gerätes scheint eine Parallele auf. Doch findet neben der an Saint-Exupéry erinnernden Zähmung toter Gegenstände eine Zähmung auch auf der anderen Seite statt. Die Heteronomie schlägt bis auf die Reaktionen des Gesamtsystems durch: es ist eben nur eine solche Verwendung vorgesehen. Ist das System benutzerfreundlich, so ist die Heteronomie total, andere Verwendungen werden nicht zugelassen. Die Maus ist Zuckerbrot und Peitsche in einem.

Menschen fischen. Auf der Höhe der Zeit zu sein, heißt heute sämtliche verfügbaren Kommunikationsmittel zu kennen und zu nutzen. Der vernetzte Bürger wurde zum Sinnbild für den mit allen Kommunikationsmitteln beschlagenen postmodernen Menschen. Wurde zuvor nur der als ins Netz gehend bezeichnet, der die schnelle Befriedigung seiner kurzfristigen Bedürfnisse und Leidenschaften über seine realen Interessen stellt, so bezeichnet dies heute den sich "anstöpselnden" Zeitgenossen. Die Terminologie zur Beschreibung der damit zusammenhängenden Vorgänge ist so technisch wie die Sache selbst und wirkt zurück auf das vernetzte Subjekt selbst: je intensiver die eigene Verstrickung, desto mehr erfährt es sich als technisches Anhängsel dieser Maschinerie, zu dem es sich macht. Wer jedoch glaubt, dass die Wichtigkeit der Inhalte mit dem Anwachsen der Kommunikation einhergehe, sieht sich getäuscht; noch das Belangloseste muss mitgeteilt werden. Kommunizieren zu können, heißt heute kommunizieren zu müssen; kaum eine Tätigkeit, die nicht durch Anrufe gestört wird, nichts mehr was ungeteilte Aufmerksamkeit erhält. Zappelte das volle Netz einstmals von Fischen, so zappeln heute die durch ständige "Requests" und "Interrupts" nervös gewordenen Bediener der Endgeräte.

Tradition zum Zwang. Die Tastatur, als konsequente Weiterentwicklung des Buchdrucks mit beweglichen Lettern, ist trotz vollmundiger Ankündigungen aus den Computerwissenschaften immer noch die wichtigste Schnittstelle zwischen Mensch und Maschine. Noch in der Anordnung der Tasten spiegeln sich die mechanischen Probleme der ersten Schreibmaschinen wider: es sollte möglichst eine schnelle Abfolge von Anschlägen verhindert werden. Seit fast hundert Jahren wird nun in Kursen geübt, was ursprünglich verhindert werden sollte: Text möglichst schnell zu tippen. Die Dressur des Menschen gelang so gut, dass eine Umreihung der Tasten längst ausgeschlossen ist. So zeigt sich hier schon die große Fessel der Computertechnologie - Kompatibilität. Es ist dies die Forderung, das Bestehende so fortzuführen und um neue Funktionen zu erweitern, dass sich diese möglichst nahtlos in das bestehende Ganze einfügen. Gerade an der Tastatur wird das Scheitern solcher Forderung deutlich. Deren Erweiterungen bestehen in zusätzlichen Tasten für zusätzliche Funktionen. Diese Tasten werden am Rande des Tastenfeldes positioniert und nötigen den Fingern weitere Wege auf. Doch befinden sich diese auf einer Tastatur eines anderen Herstellers, so finden sie sich nicht mehr zurecht. Und die als Inkarnation der Benutzerfreundlichkeit gefeierte Maus ist in Wirklichkeit Ausdruck jener Ignoranz, die übersieht, dass bereits beide Hände mit der Bedienung der Tastatur vollauf beschäftigt sind.

Literatur

[1] Adorno, Theodor, W.: Minima Moralia, Frankfurt a.M. 1951.

[2] Bittner, Peter: Informatisches Handeln und Kritische Theorie. In: Frieder Nake, Arno Rolf, Dirk Siefkes (Hrsg.): Informatik – Aufregung zu einer Disziplin. Tagung Heppenheim 2001. Uni Hamburg, FB Informatik, Bericht 235/01, S. 21-26.

Fragen an die Theoretiker, aus der Praxis eines internen Dienstleisters heraus formuliert

Kai Seim, T Systems, Darmstadt

Vorab

Zur Vorbereitung auf die Tagung habe ich versucht, all die Papiere wenigstens einmal zu lesen, die bis zum 20. Februar vorlagen. Dabei ist mir ein aus meiner Sicht immenses Defizit an Wahrnehmung der Praxis der Anwendungen der Informatik aufgefallen. Da ich für mich nicht den Anspruch erhebe(n kann), Theoriearbeit (im Sinne universitärer oder anders gearteter akademischer Beschäftigung mit dem Thema) zu leisten, andererseits aber an einem Dialog zwischen (meiner) betrieblichen Praxis und der (akademischen) Theorie interessiert bin, versuche ich im Folgenden Fragen zu formulieren, in der Hoffnung, damit zumindest einen Anfang für einen Dialog zu machen.

Theorieproduktion

Prozesse

Wie reagiert die universitäre Forschung darauf, dass sehr viel Theorie der Software-Entwicklung (i.S. des Software-Engineering) ausserhalb aller universitären Forschung entwickelt wird?

So entstehen Prozeßmodelle zur Softwareentwicklung, die Eingang in die betriebliche Praxis finden, gerade nicht an Universitäten, sondern werden – wie z.B. der RUP (ungeachtet der auch hier fehlenden Qualitäten) – von (us-amerikanischen) Unternehmen entwickelt. Das V-Modell ist eine Erfindung der Bundeswehr, resp. des BMdV.

Prozeßmodelle für betriebliche Funktionen, wie ITIL, entstehen in einer staatlichen Organisation, die heute Teil des britischen Wirtschaftsministeriums ist, die gerade *nicht* aus dem universitären Umfeld stammt. Generell ist meine Frage: Wo bleibt die Forschung an generischen Prozessmodellen für die "Wertschöpfung" im Informatikbereich?

Einschub: ich denke an Prozesse wie Software-(Neu-) Entwicklung, Software-Wartung, Betrieb von Systemen (Blech, Software, Strippe, Jobketten, Sicherungsverfahren, ...), Implementierung von Systemen, Bearbeitung von Störfällen, usw., die alle zumindest eine enorme wirtschaftliche Bedeutung haben, in denen aber auch viele Informatiker ihre tägliche Arbeit verrichten.

Insbesondere im Bereich des Systembetriebes gibt es mittlerweile einige (für mich) interessante Arbeiten, die sich mit der Qualität von Betrieb beschäftigen und der Frage, wie diese messbar gemacht werden kann. Was mir aber fehlt, ist die Verbindung zur Diskussion um generische Prozesse im Betriebsumfeld. Die Arbeiten behandeln (fast) alle eher systemnahe Fragen, befassen sich mit neuen Protokollen und deren Weiterentwicklung usw.

Hybridisierung?

Siefkes schreibt über Hybridisierung¹ und stellt fest, dass Informatiker sich meist mit der Symbolwelt beschäftigen, die Maschinen- und Menschenwelt geraten in den Hintergrund. Dazu möchte ich "dagegen" oder besser daneben setzen, dass wir "Praktiker" (mir fällt kein besserer Begriff ein) uns die Mißachtung von Mensch und Maschine um den Preis des auch persönlichen Mißerfolgs (Zielerreichung gefährdet, und damit auch der Zielbonus) nicht leisten können. Wir entwickeln, warten und betreiben Systeme im betrieblichen (und damit kapitalistischen) Alltag. Wenn unser System steht, so können Hunderte nicht mehr arbeiten. Wenn wir zu spät (und dann womöglich noch unausgereifte) Software ausliefern, ist jede Diskussion über Benutzer- oder gar Betroffenenbeteiligung u.ä. unnötig. Und die Verträge, die z.B. wir (Lebensversicherung) verwalten, haben schlußendlich ein klar definiertes Deckungskapital, dass mit eindeutiger Verzinsung angelegt sein will. Die dafür nötigen Verwaltungsakte sind sicherlich nicht mehr so handfest wie früher die Akte, in die der Mathematiker "zu Fuß" die nötigen Werte eingetragen hat. Aber auch heute agieren Menschen miteinander. Mir hat dafür bisher noch am besten das Bild des "kommunikativen Handelns" gefallen, das wir Informatiker mit unseren Systemen normieren. Und in dieser Normierungsarbeit liegt die große Kraft der Informatik, zumindest der von Informatikern entwickelten Systeme. Diese Normierung beschleunigt und verbilligt die Wertschöpfung derart, dass damit – Professionalität vorausgesetzt – erheblicher Konkurrenzdruck erzeugt wird.

¹ Aufruf zur AG „Informatik als Hybridwissenschaft“, dieser Band.

Für die Gestaltung von Informatiksystemen hat mir dabei bisher immer ein Verständnis von Prozessen geholfen, das aus der Systemtheorie Luhmanns hergeleitet werden kann.

Tot/Lebendig – Trivial/Innovativ - DV-Systeme als Mittel, um den trivialen Anteil von Prozessen (massiv) zu erhöhen, mit den entsprechenden Effizienzgewinnen. Und andererseits den Verlusten an Flexibilität (denn: es sind ja maschinell abgebildete Normen kommunikativen Handelns).

Hierzu nehme ich kaum Forschung wahr (was ja auch an meiner verengten Sicht liegen kann). Was ich hier fragen möchte: Wo bleibt die Weiterentwicklung solcher Ansätze, wie der von Budde/Züllighoven "Softwarewerkzeuge in einer Programmierwerkstatt", in dem Heidegger intelligent "beklaut" wurde und damit einsetzbar in Diskussionen zwischen Systementwicklern. Wann macht jemand dasselbe mit Habermas, oder Luhmann?

Änderung der Ausbildung?

Später in seinem Text argumentiert Siefkes für eine Änderung der Ausbildung. Puppe stellt sich wiederum in seinem Text dagegen mit der These, Informatiker hätten die Aufgabe, Werkzeuge (zur Erstellung von Systemen) zu realisieren, und verweist ansonsten witziger Weise auf Kent Beck. Gesetzt den Fall, Puppe wäre zu folgen (wogegen ich im weiteren argumentieren möchte), hätte dies m.E. mindestens zwei sofortige Konsequenzen:

1. Mangels Masse verdünnt sich die Informatikforschung (und Theoriebildung) in einige wenige Konzerne, die noch Basissysteme (im Sinne von Betriebssystem, Datenbanksystem, Compiler etc.) herstellen und vertreiben.
2. Es fehlt immer noch die Aussage, wie denn die große Masse der Informatiker denn nun Systeme realisieren und später dann betreiben soll. Gibt's denn für die Vertragsverwaltung von LV-Verträgen mittels DV keine Hilfe seitens der Theoretiker? Bin ich denn ganz alleine im dunklen Keller der Praxis und muß die Lippen zum Pfeifen spitzen?

Im Zweifel unterliegt Puppe dem alten Mißverständnis (das in Dreijahreszyklen auch durch die Computerwoche geistert; der sog. CIO), dass Informatiker, die mehr können als Programmverifikation mittels Hoarekalkül, ganz schnell keine Informatiker, sondern (z.B.) Vertragsverwalter werden. Die Praxis zeigt es täglich: dem ist nicht so. Und leider gibt es viel zu wenige, die den Hoarekalkül in ihrem Studium ausreichend zur Kenntnis genommen haben und für die Entwicklung qualitativ hoch stehender Software (Software aus Eisen statt aus Plastik, um G. Wohland zu zitieren) einsetzen können.

Dieses alles fehlt heute (fast) jedem Absolventen:

1. Projektmanagement (als Teil von Organisationstheorie)
2. Teamwork (Präsentieren, Moderieren, Zuhören, Protokollieren)
3. Betriebswirtschaftliche Grundkenntnisse (Was ist eine Buchung? Was ist ein Rechtstitel?)
4. Kulturelle Grundkenntnisse (Heidegger? Wittgenstein? Marx? Habermas?)

Verantwortungsdiskussion

Ist Informatik eine Frage der Moral? Oder die Theoriearbeit als Informatiker?

Puppe beklagt in seiner Zusammenfassung, dass (eine bestimmte Sorte von) Informatiker(n) aufgrund der massiven Auswirkungen, die informationsverarbeitende Systeme (in unserer Gesellschaft) haben, dazu neigen, sich für viele der damit verbundenen Probleme zuständig zu erklären. Diese Kritik möchte ich gerne unterstützen.

Viele Artikel, die ich gelesen habe, verwenden moralische Kriterien. Um es (hoffentlich) klar zu stellen: Moral muß sein! Sie stört aber sehr (durch Erzeugen blinder Flecken) bei der Theoriearbeit. Dort gehört sie einfach nicht hin. So ist mir z.B. unklar, welchen Beitrag eine Diskussion sogenannter Genderaufladungen bringen kann. Böse formuliert wirkt das auf mich ähnlich sektenhaft wie die KI-Euphorie der späten 80er / frühen 90er. Aus betrieblicher Praxis kann ich zwar bestätigen, dass z.B. bei Gerling E&L in der Breite Frauen eher schlechter bezahlt werden als Männer. Dies gilt aber nicht für die unteren Gehaltsgruppen (hier verdienen die Männer weniger) und nicht für die Führungskräfte (100% der Abteilungen im Bereich Informatik werden von Frauen geleitet, deren Chef wiederum ein Mann ist :-).

Ist jemand verwundert?

Neue Arbeitsmethoden und ihre Folgen

Jörg Werda, Deutsche Lufthansa, Hamburg

Es gibt viele Schlagworte der modernen Arbeitswelt: "Lean Production", "Lean Management", "Flexibilisierung", "Teamorientierung" und "Globalisierung", um nur einige zu nennen. Diese Begriffe haben einen hohen Werbeeffekt, um die Akzeptanz neuer Arbeitsmethoden in der Gesellschaft, aber insbesondere bei den Arbeitnehmern⁵² zu erhöhen, aber auch, um andere Arbeitsmethoden als veraltet und unmodern darzustellen. Dennoch sind sie, wie jede gute Werbung, nicht einfach aus der Luft gegriffen, sondern haben reale Hintergründe.

Arno Rolf hatte die Hoffnung, daß insbesondere durch Fortschritte in der Informatik neue Gestaltungsmöglichkeiten entstehen, die auf eine Humanisierung der Arbeitswelt zielen [Rolf 2001, S.19]. Er hoffte auf kleine, autonome, Organisationsformen [ebd.], ich füge hinzu, er hofft also auf kleine Teams, die ohne einen großen unproduktiven "Wasserkopf" selbstorganisiert und eigenverantwortlich ihre Arbeit verrichten, und das aus einer dafür erforderlichen ganzheitlichen Sicht [ebd.]. Ja führt dies vielleicht sogar dazu, daß wir bekümmerte Macher und gestaltungsorientierte Interpretierer [Rolf 1992, S.44] erhalten? Es klingt auf jeden Fall nett. Ist dies nicht genau das, was alle Kritiker der Lohnarbeit schon immer wollten? Ist dies nicht ein Stück weit eine Aufhebung der entfremdeten Arbeit? Wer sollte also etwas dagegen haben.

Das gleiche gilt auch für die Flexibilisierung der Arbeit. Wer ist nicht gerne vielseitig einsetzbar, sozusagen multifunktional. Wer möchte schon in verkrusteten Arbeitsstrukturen arbeiten und jeden Tag für den Rest seines Lebens von 7:00 bis 15:30 in die Fabrik gehen und eintönige Arbeit machen? Ist es da nicht besser, flexible Arbeitszeiten zu haben, zu kommen und zu gehen, wann man will? Wer kann dazu schon ‚Nein‘ sagen?

Auch die Globalisierung stellt neue Anforderungen. Es wird ein Kontakt mit Gruppen in anderen Kulturen notwendig, und es geht bei der internationalen Vernetzung daher um den Erwerb spezieller, auch internationaler sozialer Kompetenzen [Rolf 2001, S.20]. Andere Kulturen zu verstehen und "multikulturelle" Kompetenzen bei der Arbeit zu erlangen, ist sozusagen der letzte Baustein zur Zufriedenheit des aufgeklärten Arbeitnehmers. Ist dies die schöne, neue Arbeitswelt?

Nun, Arno Rolf zitiert selbst, inzwischen ernüchtert, Walter Volpert [1992], der zu Recht feststellt, daß all diese zusätzlichen Anforderungen und Qualifikationen zuerst einmal erbracht werden müssen. Er fragt, ob Arbeitnehmer hierdurch nicht einfach zusätzlich ge- und überfordert werden. In erster Linie ist Lean Production nämlich die Einsparung des "Verwaltungswasserkopfs", also eine Einsparung an Personal. Die entsprechenden Verwaltungsaufgaben werden jetzt anderen Arbeitnehmern zugewiesen, häufig ohne eine entsprechend bessere Bezahlung, fast nie werden die Einsparungen im Verwaltungsbereich auf die Gehälter der übrigen Arbeitnehmer vollständig verteilt.

Auch Flexibilisierung bedeutet in erster Linie den Verzicht auf Überstunden- und Spätarbeitszuschläge. Wer flexibel arbeitet, macht keine Überstunden, sondern bummelt Überstunden ab, wenn der Auftragseingang schlecht ist und "klotzt ran", wenn "Not am Mann" ist. Dies sind jedoch beides keine Faktoren, die der Arbeitnehmer selbstbestimmt entscheidet. Flexibilisierung von Arbeitszeiten und Einsatzorten ermöglichen es, mit geringerem Personalaufkommen die gleiche Arbeit zu verrichten, weil es nicht mehr notwendig ist, unproduktive Arbeitskraftreserven auf Vorrat für Spitzenzeiten aufzubauen.

Wie im Rahmen des Seminars festgestellt wurde, hat es ausgehend vom Taylorismus eine kontinuierliche Weiterentwicklung, sozusagen eine Verfeinerung der Arbeitsmethodik gegeben. Es zeigt sich, daß alle genannten modernen Arbeitsmethoden dafür sorgen, daß die Arbeit produktiver und rationeller gestaltet wird, daß mit weniger Personalkosten das Gleiche produziert oder mit gleichen Personalkosten mehr produziert werden kann. Die Profitabilität eines Unternehmens erhöht sich also. Dies ist auch die entscheidende Folge moderner Arbeitsmethoden und der Grund, warum sie eingeführt werden. Ist jemand verwundert?

⁵² Warum heißen Arbeitnehmer eigentlich Arbeitnehmer, obwohl sie ihre Arbeitskraft geben, und der Arbeitgeber ihre Arbeit nimmt?

Literatur

- [Coy 1992] Coy, Wolfgang et al. (Hrsg.): *Sichtweisen der Informatik*. Braunschweig, Wiesbaden: Friedr. Vieweg & Sohn, 1992.
- [Kutzner 2001] Kutzner, Markus: *Die subjektwissenschaftliche Stufe der Arbeit*. Lehrveranstaltung "Theorie(n) der Informatik", Dirk Siefkes, Technische Universität Berlin, 2001/02.
- [Rolf 1992] Rolf, Arno: *Sichtwechsel – Informatik als Gestaltungswissenschaft*. In [Coy 1992], S. 33-47.
- [Rolf 2001] Rolf, Arno. *AG Gestalterische Aufregung: Anregungen*. In: Frieder Nake, Arno Rolf, Dirk Siefkes: *Informatik: Aufregung zu einer Disziplin*. Tagung Heppenheim 2001. Uni Hamburg, FB Informatik, Bericht 235/01, S. 15-16.
- [Volpert 1992] Volpert, Walter: *Erhalten und gestalten – von der notwendigen Zähmung des Gestaltungsdrangs*. In [Coy 1992], S. 171-180.

Skizze einer Theorie der Informatik-Anwendungen

Bericht zur AG Theorie der Anwendungen der Informatik

Peter Brödner, Kai Seim, Gerhard Wohland

1 Einführung: Das Scheitern schöner Absichten

Im Unterschied zu klassischen Anwendungsfeldern technischer Systeme der Energie- und Stoffumwandlung warten Informatik-Anwendungen mit allerlei Auffälligkeiten und Überraschungen auf. Es sind dies Auffälligkeiten, die sich trotz aller Bemühungen um systematisches und methodisch kontrolliertes Vorgehen bei der Gestaltung und Einführung informationstechnischer Systeme gegen die Erwartungen der Beteiligten – also von Systementwicklern, Systemanwendern und -betreibern wie auch Benutzern – immer wieder zutage treten.

Zunächst einmal ist zu beobachten, dass sich bei den Informatik-Anwendungen Fälle von Software-Havarien oder gar von gänzlich gescheiterten Projekten auffällig häufen. Zwar treten auch in klassischen Bereichen der Technik gelegentlich schwere Designfehler oder misslungene Projekte in Erscheinung, doch bei weitem nicht mit der Regelmäßigkeit und Häufigkeit wie bei Informatik-Anwendungen. Zugespielt gesagt: In den Informatik-Anwendungen ist der misslungene Systementwurf oder die Anwendungshavarie die Regel, in klassischen Bereichen der Technik die Ausnahme.

Mit Anwendungshavarie der Informationstechnik ist dabei gemeint, dass Konzeption und Anwendung informationstechnischer Systeme zwar meist nicht zur Gänze scheitern (was noch häufig genug vorkommt, indes nur selten zugegeben wird), aber doch mit bedauerlicher Regelmäßigkeit die angesetzten Kosten- und Zeitbudgets um ein Vielfaches überschreiten und zugleich hinter den erwarteten Rationalisierungseffekten deutlich zurückbleiben. Bekannt gewordene Großhavarien sind beispielsweise der amerikanische Pharmagroßhändler FoxMeyer (gescheiterte Einführung eines ERP-Systems (Enterprise Resources Planning) mit anschließender Pleite; Scott/Vessey 2002) oder die gänzlich gescheiterte Entwicklung eines neuen integrierten Buchungssystems für Hotels, Mietwagen und Flüge (bei der in dreieinhalb Jahren 125 Mio. USD ohne verwendbare Ergebnisse in den Sand gesetzt wurden, Oz 1994). Diese Beobachtungen schlagen sich auch im sogenannten *Produktivitätsparadox* der Informationstechnik nieder, demzufolge sich trotz jahrzehntelanger beispielloser Investitionen in Informationstechnik keine merklichen zusätzlichen Produktivitätsverbesserungen verzeichnen lassen (vgl. auch Brödner 1997, Landauer 1995).

Einem Bonmot Ortega y Gasset zufolge ist "Technik die Anstrengung, Anstrengungen zu vermeiden". Und bislang, im Bereich klassischer Einsatzfelder von Technik (der Energie- und Stoffumwandlung) hat das auch wirkungsvoll funktioniert: In den letzten 150 Jahren, seit den Tagen des kommunistischen Manifests, hat sich die Produktivität in entwickelten Industrieländern verzwanzigfacht. Demgegenüber hat die Informationstechnik in den letzten 50 Jahren keine spürbare zusätzliche Steigerung der Produktivität bewirkt.

Wie ist das möglich? Diese Auffälligkeiten bedürfen der Erklärung. Für uns sind derartige Auffälligkeiten und Überraschungen, die allen Erwartungen hartnäckig Hohn sprechen, Ausdruck von Theoriemangel. Sie veranlassen uns, über eine Theorie der Informatik-Anwendungen zu reflektieren, die wir im folgenden entfalten. Eine Theorie hat dabei den doppelten Sinn, erstens die genannten Auffälligkeiten systematisch zu erklären und zum zweiten, gestützt auf diese Erklärungen, Ansätze zu Handlungsanleitungen zu gewinnen, die künftige Praxis auf Dauer zu verbessern.

Wir beginnen im nächsten Abschnitt (2) zunächst mit der genaueren Darstellung empirischer Befunde zum Produktivitätsparadox, um die hier einfürend konstatierten Feststellungen und den Ernst der Lage zu untermauern. Im 3. Abschnitt entwickeln wir dann unsere theoretischen Überlegungen auf Basis von einigen sehr grundlegenden Unterscheidungen, die uns als "Denkwerkzeuge" helfen, die Besonderheiten der informatischen Gegenstände, ihrer Zielgebiete und des Umgangs mit ihnen zu erklären. Im 4. Abschnitt betrachten und analysieren wir dann mit Hilfe unserer theoretischen Überlegungen den Einsatz von IT-Systemen in Wertschöpfungsprozessen als besonders wichtigen exemplarischen Fall von Informatikanwendungen. Dabei wollen wir den Wert der Theorie zur Erklärung bisheriger Havarien unter Beweis stellen und zugleich Ansätze zu einer verbesserten Praxis aufzeigen. Im 5. Abschnitt formulieren wir dazu Prüfkriterien für den Einsatz von IT-Systemen.

2 Evidenz: Das Produktivitätsparadox der Informationstechnik

Das Produktivitätsparadox der IT verweist auf den erstaunlichen Sachverhalt, dass trotz jahrzehntelanger massiver und im Zeitverlauf wachsender Investitionen in Informationstechnik keine merklichen Zuwächse der Arbeitsproduktivität (Produktionswert je Beschäftigtenstunde) über übliche Produktivitätsfortschritte hinaus festzustellen sind. So haben sich etwa in den drei Dekaden von 1960 bis 1990 die realen IT-Investitionen in den USA nach anfänglich fast konstantem Verlauf bis 1975 auf niedrigem Niveau (ca. 20 Mrd. USD p.a.) bis zum Jahre 1990 auf 220 Mrd. USD mehr als verzehnfacht. Im selben Zeitraum stieg die Produktivität in den produzierenden Sektoren mit relativ niedrigen gleich bleibenden jährlichen Zuwächsen, während in den Dienstleistungssektoren die Produktivität seit 1972 stagnierte (Brynjolfsson 1993). Das hat den Nobelpreisträger für Ökonomie Robert Solow zu dem berühmt gewordenen Ausspruch veranlasst: "You can see the computer age everywhere except in the productivity statistics" (Solow 1987).

Daran hat sich bis heute nicht viel geändert. Noch immer lässt sich auf makroökonomischer Ebene kein zuverlässiger Zusammenhang zwischen IT-Investitionen und Produktivitätswachstum nachweisen, obgleich (wiederum in den USA) seit 1991 die IT-Investitionen die Investitionen in Produktionstechnik übersteigen (Stewart 1997). Nun gab es in der zweiten Hälfte der 90er Jahre in den USA einen deutlichen Produktivitätssprung von jahresdurchschnittlich 1,4% in den Jahren 1972 bis 95 auf jahresdurchschnittlich 2,5% in den Jahren 1995 bis 2000. Schon glaubte man feststellen zu können, dass sich nun endlich, wenn auch mit jahrzehntelanger Verzögerung, die gigantischen IT-Investitionen auch auf die gesamtwirtschaftliche Produktivität positiv auszuwirken beginnen. Doch weit gefehlt.

Wie die jüngste vom McKinsey Global Institute (MGI) durchgeführte Produktivitätsanalyse in den USA aufzeigt, ist dieses außergewöhnliche Produktivitätswachstum ausschließlich auf besondere Entwicklungen in den sechs Branchen Groß- und Einzelhandel, Wertpapierhandel, Elektronik, Maschinen (einschl. Computer) und Telekommunikationsdienste zurückzuführen. Diese gehen aber im wesentlichen auf Restrukturierungen der Wertschöpfungsprozesse zurück und hängen nur zu geringen Teilen mit erhöhten IT-Investitionen zusammen. Alle übrigen 53 Branchen zusammen (mit rund 70% des Bruttoinlandsprodukts (BIP)) tragen nichts mehr zum Produktivitätswachstum bei, da einzelnen Bereichen mit geringfügiger positiver Entwicklung solche mit negativer Produktivitätsentwicklung – trotz teilweise überdurchschnittlich hoher IT-Investitionen – gegenüberstehen (vgl. Abb. 1; MGI 2001).

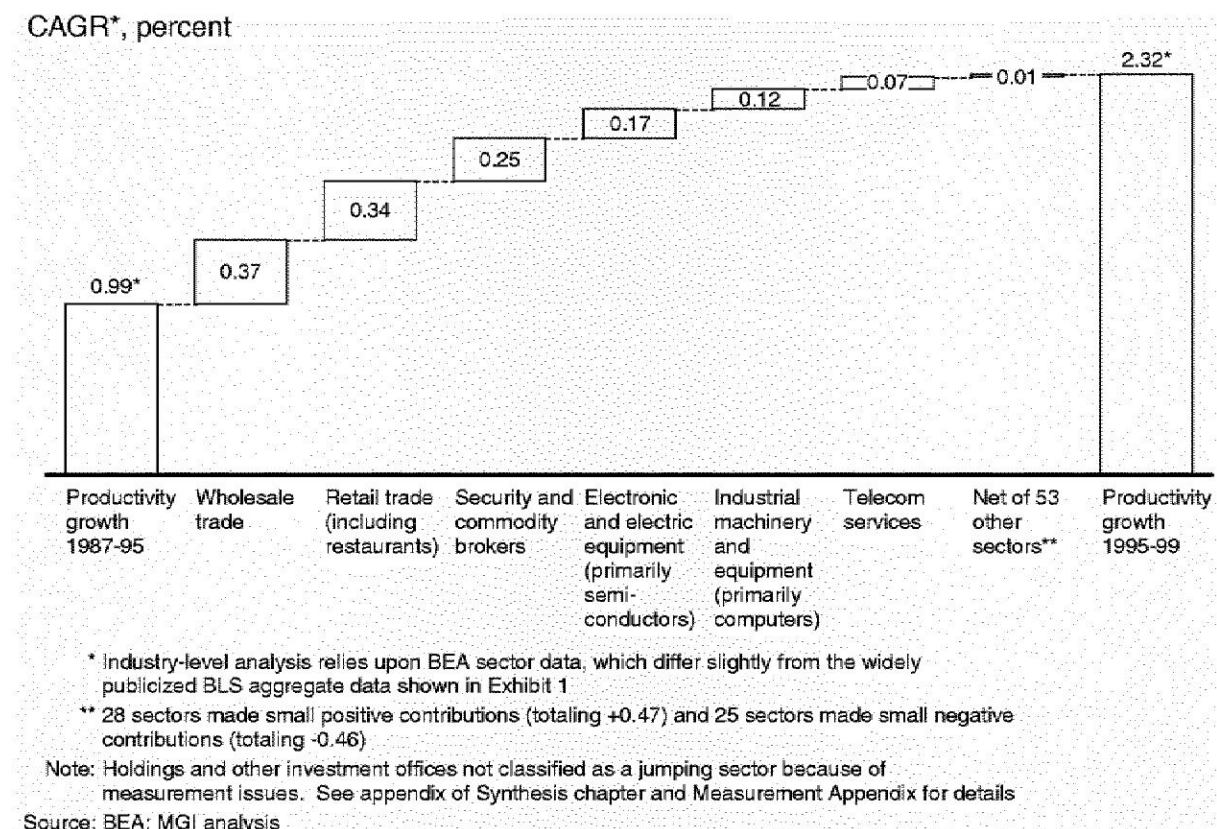


Abb. 1: Beiträge einzelner Wirtschaftszweige zum Produktivitätswachstum der USA

So sind die außerordentlichen Produktivitätssteigerungen im Groß- und Einzelhandel im wesentlichen auf Prozessverbesserungen durch restrukturierte Logistikketten, große Verkaufseinheiten und automatisierte Lagerhaltung in Distributionszentren, zu geringeren Teilen auch auf den breiten Einsatz von Barcode-Lesern und EDI zurückzuführen. Die Nutzung der IT ist hier Bedingung, aber nicht Ursache der Produktivitätssteigerung. Verbesserungen des Halbleitersektors und der Computerherstellung wurzeln hauptsächlich in der Leistungssteigerung der Chips und der Systemeinheiten sowie im raschen Wachstum des Internet. Nur im Wertpapierhandel beruhen die Produktivitätszuwächse wesentlich auf dem Wachstum von Online-Handel und IT-Einsatz in der Auftragsabwicklung.

In den letzten Jahren hat nun die wissenschaftliche Debatte um das IT-Produktivitätsparadox in den USA eine erstaunliche Wende genommen. Wenn es offensichtlich so schwierig ist, auf makroökonomischer Ebene positive Wirkungen der IT-Investitionen nachzuweisen – weil in der Tat zu viele, möglicherweise gegenläufige Einflüsse wirksam werden können oder auch Messprobleme geltend gemacht werden –, können möglicherweise mikroökonomische Analysen weiterhelfen. Die bereits zitierte jüngste Produktivitätsanalyse von MGI ist insofern Ausdruck dieses Perspektivwechsels, als sie Unterschiede in den Branchen analysiert und dort wesentliche unternehmerische Strategien untersucht, die die Unterschiede in der Produktivitätsentwicklung zu erklären vermögen.

In ihrer jüngsten Studie haben Brynjolfsson und Hitt (2001) aufgrund ökonomischer Analysen von Daten aus rund 400 großen US-Unternehmen zeigen können, daß

- IT-Systeme die Leistungsfähigkeit von Unternehmen nur dann steigern können, wenn deren Einführung mit Dezentralisierung, objektorientierter Reorganisation und Investition in Humankapital verbunden wird,
- Unternehmen mit dezentralisierten Organisationsstrukturen höhere Produktivität in der Nutzung von IT-Systemen erreichen als solche, die nur in IT investieren,
- "intangible assets", z.B. kollektive Handlungskompetenz, den realisierten Nutzen von IT-Systemen stark beeinflussen,
- der Aufwand für organisatorische Erneuerung und Qualifizierung, etwa bei Einführung von ERP-Systemen, um den Faktor 4 höher ist als die Ausgaben für Hard- und Software.

In ähnlichem Sinne verwies auch die OECD erst kürzlich auf eindeutige Befunde dafür, dass mit der Nutzung von IT-Systemen zwar Produktivitätssteigerungen einhergehen können, diese jedoch ohne Reorganisation der Arbeit ausbleiben. Insbesondere korrelieren mit dem produktiven Einsatz von Informationstechnik Veränderungen der Arbeitsorganisation wie flachere Hierarchien, teilautonome Arbeitsgruppen und die Übertragung betrieblicher Verantwortlichkeiten auf die Arbeitnehmer (OECD 2001).

Auf der Ebene des unternehmerischen Umgangs mit IT-Systemen trennt sich offensichtlich die Spreu vom Weizen. Bei deren Gestaltung, Einführung und Nutzung, mithin bei der Art und Weise ihrer sozialen Einbettung in die jeweilige Organisation entscheidet sich, ob im Gebrauch ökonomischer Nutzen aus den Systemen gezogen und letztlich höhere Produktivität erreicht werden kann oder nicht. Häufig wird dieser Zusammenhang jedoch ignoriert und so bestimmen IT-Havarien meist das Bild – in einem Ausmaß, das man bei früheren Technikformen nicht kannte.

Beispielsweise zeigt sich beim Einsatz von Produktionsplanungs- und -steuerungssystemen (PPS bzw. ERP), dass sieben von zehn Unternehmen einer rein technikzentrierten Perspektive auf den Systemeinsatz folgen – mit außerordentlich schädlichen Folgen für ihre wirtschaftliche Leistung: Implementierungsprojekte überziehen regelmäßig die Zeit- und Kostenbudgets um ein Vielfaches, und trotz hoher Kosten verbessern sich wettbewerbsentscheidende Leistungsparameter wie Durchlaufzeiten, Bestände oder Produktivität kaum. Der Implementierungsprozess konzentriert sich ganz überwiegend auf die Bestimmung der Systemanforderungen und -konfiguration, während Fragen der Nutzung und kollektiven Aneignung der Systeme kaum in Betracht gezogen werden. In der Folge bleiben viele Funktionen ungenutzt, notwendiges Wissen über Geschäftsprozesse und Handlungsfolgen im Organisationszusammenhang ist unter den verschiedenen Akteuren unzureichend vermittelt und überflüssige Datenhalden werden angesammelt.

Nur eine kleine Minderheit von Unternehmen setzt dagegen mit großem wirtschaftlichen Erfolg an organisatorischen Veränderungen an, indem sie Wertschöpfungsprozesse konsequent auf den Kundennutzen ausrichten und objektorientiert restrukturieren, eine dementsprechend angepasste Konzeption und Nutzung der gleichen IT-Systeme als Arbeitsmittel und Medium der Kooperation entwickeln sowie zur Bewältigung der Veränderungen und zur Aneignung der Systeme im Handlungsfeld kollektive Lernprozesse organisieren. Nur wenn die Arbeitsweise, mithin die Regeln

gemeinsamen Handelns, im Gebrauch der Systeme verändert werden, lassen sich deren Nutzenpotentiale aktivieren (Davenport 1998, Maucher 1998, 2001).

Woran liegt es nur, dass mit dem Einsatz von IT-Systemen verbundene Produktivitätserwartungen so häufig enttäuscht werden? Die sich trotz enormer Anstrengungen und beachtlicher Fortschritte im Software-Engineering (vgl. z.B. Sommerville 2001) immer wieder neu auftuende Kluft zwischen Wunsch und Wirklichkeit bei der effizienten Nutzung von IT-Systemen bedarf der Erklärung, der wir uns im folgenden zuwenden.

3 Zur Theorie der Informatik-Anwendungen: Der Computer als semiotische Maschine

"A computer is what one does with it."

J. M. Carroll

Im Laufe der nun gut fünfzigjährigen Geschichte praktischer Computertechnik hat es ganz unterschiedliche Auffassungen von deren Natur gegeben: Die einen sehen den Computer als Automaten, als universelles Mittel der Rationalisierung durch Automatisierung von Kopfarbeit, andere sehen darin eher ein Werkzeug, das als benutzungsgerecht und aufgabenangemessen gestaltetes Arbeitsmittel menschliche Arbeit unterstützen und effizienter machen soll, wieder andere betrachten ihn vor allem als Medium der Kommunikation und Kooperation (Coy 1995). Es muss schon ein merkwürdiges technisches Artefakt sein, das derart unterschiedliche Sichtweisen erlaubt, die gleichwohl alle eine gewisse Berechtigung vorweisen können. Sie alle betonen aber nur einen besonderen Aspekt der datenverarbeitenden Maschine, ohne deren wirkliche Natur, ihre Wirkungsweise im Gebrauch, im Vergleich zu anderen Maschinen angemessen zu verstehen. Die Entwicklung einer angemessenen Theorie der Informatik ist also dringend geboten (um auf Coys Frage von 1989 erneut zu antworten). Bevor wir dieser Aufgabe näher treten können, müssen freilich noch einige dafür wesentliche begriffliche Unterscheidungen getroffen werden.

Beobachten und Denken setzen Unterscheiden voraus. Innovatives Denken beruht auf neuen Unterscheidungen; es nimmt Überraschungen auf. In diesem Sinne sollen die nachstehenden begrifflichen Unterscheidungen als "Denkwerkzeuge" helfen, die angesprochenen Probleme von Informatik-Anwendungen besser zu verstehen und dabei ein angemessenes Bild vom Computer zu gewinnen.

3.1 Einige grundlegende Unterscheidungen

Die Dynamik wirtschaftlichen Handelns steckt voller Überraschungen. Um darin erfolgreich bestehen zu können, sind angemessenes Beobachten und Denken gefordert. Die für wirtschaftliches Handeln und Organisationsentwicklung in dynamischer Umgebung wichtigste Unterscheidung ist die zwischen **tot** und **lebendig**. Auf ihr basieren weitere Unterscheidungen:

*Unterscheidung **kompliziert** (trivial, Standard) versus **komplex**:*

Kompliziertheit ist ein Maß für die Unwissenheit eines Beobachters. Durch einsichtiges Lernen wird Kompliziertes einfacher, bis es letztlich durchschaut ist und als das erscheint, was es ist: trivial oder ein Standard. Ein mathematischer Beweis, ein Computerprogramm oder eine Landkarte sind Beispiele für Kompliziertes.

Komplexität ist ein Maß für das Überraschungspotential eines lebendigen Systems, für seine Autonomie und Kontingenz. Durch Kommunikation werden lebendige Systeme vertraut. Sie verlieren aber nie ihre Fähigkeit, durch unerwartetes Verhalten, etwa durch Ideen, zu überraschen. Komplexität lässt sich nicht durch Lernen reduzieren.

*Unterscheidung **Daten** versus **Information**:*

Information ist "jeder Unterschied, der (im praktischen Handeln) etwas ausmacht" (Bateson 1980). Ein Unterschied ist eine Relation zwischen Dingen (z.B. zwischen Erwartungen und Ergebnissen von Handlungen, zwischen wahrgenommenen Umweltgegebenheiten), nicht etwa eine Eigenschaft von Gegenständen oder Vorgängen. Diesem handlungsbezogenen, semantischen Begriff von Information zufolge entsteht Information durch die Interpretationsleistung handelnder Menschen, indem sie einem wahrgenommenen Unterschied Bedeutung zuweisen. *Bedeutung* ist mithin das, was einen Unterschied in einem Handlungskontext ausmacht. Sie ist Ergebnis einer Interpretation, mithin ein Konstrukt des aktiven Gehirns, nicht etwas in der Welt Gegebenes.

Daten sind demgegenüber Funktionen oder Signale, die aufgrund gegebener oder unterstellter Abmachungen Information darstellen (DIN 44 300). Information existiert in der Lebenswelt der Menschen. Sie wird aktiv erzeugt und kann weder übertragen noch gespeichert werden. Daten sind tote Strukturen. Sie können in einem Computer als dessen Zustände oder Signale gespeichert und verarbeitet und zwischen Rechnern übertragen werden.

Unterscheidung Wissen versus Können:

Können ist die praktische Handlungskompetenz von Menschen, gekennzeichnet durch ihre Fähigkeit (und Fertigkeit) in einer Situation angemessen zu handeln, um ihre Intentionen und Interessen zu verwirklichen; es ist erfolgreiches situiertes Handeln.

Wissen ist durch Begriffsbildung ins Bewusstsein gehobene Erfahrung (indem neue Situationen durch Unterscheiden und Benennen mit bekannten Bedeutungen in Zusammenhang gebracht werden). Mit Hilfe relevanter Begriffe wird Erfahrung geordnet und expliziert.

Wesentlich für das Verständnis von Können und Wissen ist deren *dynamische Beziehung*, die Art und Weise, wie sie einander wechselseitig hervorbringen: Vorgängig ist stets die natürliche Handlungskompetenz, das Können. Erst durch besondere Anstrengungen (etwa durch Reflexion, Begriffsbildung oder Experimentieren) lassen sich Aspekte des Handelns (aber niemals zur Gänze) in explizites, theoretisches Wissen transformieren. Durch dessen Aneignung für praktische Zwecke, d. h. durch Interpretation seiner Funktionen im Handlungskontext, wird das explizite, abstrakte Wissen wieder in einen – freilich eben dadurch veränderten – Praxiszusammenhang gestellt (rekontextualisiert), ein Vorgang, der seinerseits Können erfordert. Dieser Dialektik der (stets partiellen) Explikation von Erfahrung in Wissen und der Aneignung von Wissen als erweitertes Können zufolge ist das in Zeichen oder Artefakten vergegenständlichte, dekontextualisierte Wissen "geronnene Erfahrung" und wird durch Aneignung zu-gleich Teil einer veränderten Praxis (Brödner 1997).

Neben diesen sehr grundlegenden Unterscheidungen sind nun noch solche von Bedeutung, die die Arbeitsweise von Organisationen in ihrem zunehmend veränderlichen Umfeld betreffen. Wegen der gesteigerten Marktdynamik bleibt weniger Zeit, Probleme zu lösen. Zumindest für operative Aufgaben wäre die Einbeziehung einer Zentrale zu langsam. Heute müssen die meisten operativen Probleme dort gelöst werden, wo sie entstehen, vor Ort, in der Peripherie.

Wo Probleme gelöst werden, entstehen Können und neues Wissen. So entwickelt sich neben dem Wissensbereich der Zentrale eine Vielfalt von dezentraler Kompetenz. Dieses "Gegenwissen" ist für konventionelle Organisationen eine Bedrohung. Für flexible Organisationen ist das dezentrale Gegenwissen der Brennstoff für konstruktive Widerständigkeit und damit für Innovation.

Unterscheidung Zentrum versus Peripherie:

Auf ein Unternehmen wirken zwei Zwänge, der Markt und die Geldgeber (Abb. 2). In träger Umgebung wirken beide Zwänge auf die Führung. In dynamischer Marktumgebung kann der Zwang des Marktes nicht mehr tief ins Unternehmen eindringen. Die entsprechenden Probleme müssen ohne die Zentrale bearbeitet werden. Neben dem Wissensbereich Zentrum entsteht ein weiterer – die Peripherie. Beides sind also weder Funktionen noch Orte:

Zentrum ist die Zusammenfassung aller Tätigkeiten, die unter dem Druck der "Shareholder" ablaufen.

Peripherie ist die Zusammenfassung aller Tätigkeiten, die unter dem Druck des Marktes ablaufen.

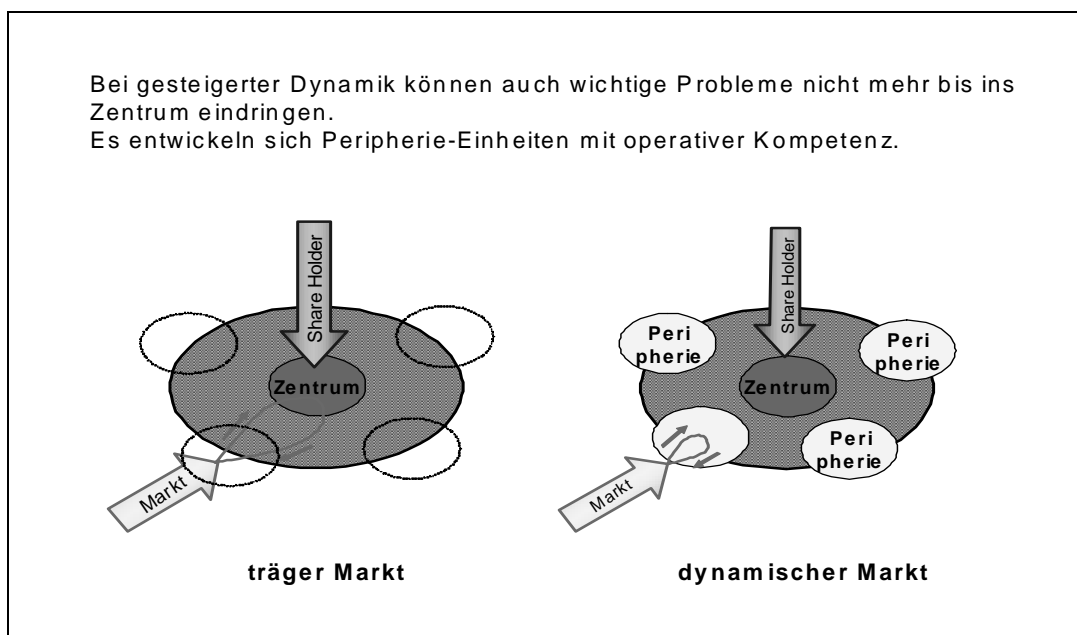


Abb. 2: Ausdifferenzierung von Zentrum und Peripherie

Unterscheidung *Methode* versus *Theorie*:

Methoden sind Systeme von Regeln für menschliches Handeln. Sie fangen erfolgreiches Handeln ein und machen es wiederholbar. Ihre Anwendbarkeit muss aber erst geprüft werden; es gibt keine Regel, wie man Regeln richtig anwendet. Methoden leben von Wiederholung, dem Trivialen im Geschehen. Methoden sind Werkzeuge für das Triviale.

Theorien sind Systeme konsistent aufeinander bezogener Begriffe und Argumente (Aussagen). Mit erlaubten Denkkoperationen kann die Theorie nicht verlassen werden, sie ist selbstreferentiell geschlossen. Theorie ist Werkzeug zur Bewertung von Ideen des Problemlösens. Ohne solche Ideen ist Theorie nutzlos. Theorie ist wie ein Scheinwerfer, der etwas durch Beleuchtung sichtbar macht: Mit Mitteln der Theorie können in der Praxis identifizierte Probleme durch den kreativen Akt der Transformation und Modellbildung explizit ausgedrückt werden. Sie stellt formal gesicherte Lösungsmethoden bereit, deren Ergebnisse aber erst noch für die Praxis interpretiert werden müssen. Aus Theorie folgt nur Theoretisches, sonst nichts, schon gar nichts Praktisches. Theorien sind Werkzeuge für das Komplexe.

Unterscheidung *Führung* versus *Steuerung*:

Steuerung setzt Hierarchie in der Organisation voraus. Kann diese durch Wissensvorsprung erzeugt werden, ist Steuerung effektiv. Wird sie entgegen einem Wissensgefälle praktiziert, entsteht durch Steuerung nur Havarie. Komplexe Projekte verfügen bezüglich ihres Themas über maximales Können und Wissen. Sie sind geeignet, einmalige Probleme zu bearbeiten, können aber nicht gesteuert werden. Durch hohe Dynamik der Umgebung entsteht Selbstorganisation in der Peripherie von Organisationen. Selbstorganisation ist konservativ und deshalb in dynamischer Umgebung nur kurz lebensfähig. Deshalb ist *Führung* in dynamischer Umgebung konstruktive Störung von Selbstorganisation. Komplexe Projekte sind immer selbstorganisiert.

3.2 Kennzeichen semiotischer Maschinen

Mit diesen Unterscheidungen gelingt es nun, den eigentlichen Gegenstand von Informatik genauer zu kennzeichnen. Computer sind, im Unterschied zu klassischen stoff- und energieumwandelnden Maschinen, datenverarbeitende Maschinen. Ihre per Algorithmus erzeugten bzw. veränderten Zustände oder Signale lassen sich als Träger von Zeichen deuten; darauf beruht ihre universale Einsetzbarkeit und große interpretative Flexibilität. Ihr Zweck ist die Maschinisierung von Kopfarbeit, die maschinelle Ausführung wohl bestimmter Transformationen von Zeichen, intern durch Signale verkörpert. Ihr Einsatz verlangt Berechenbarkeit, die vollständig festgelegte, eben algorithmische Beschreibung dieser Transformationen (Turing), die sich in Form von eindeutig bestimmten, wiederholbaren Folgen von Signalveränderungen (sprich: Folgen von Maschinenzuständen) abbilden lassen.

Kopfarbeit ist gesellschaftlicher Natur und beruht auf Bildung und Gebrauch von Zeichen. Mit der Bildung von Zeichen und intersubjektiv geteilten Interpretationsschemata weisen wir durch unser Tätigsein Dingen oder Vorgängen Bedeutung zu. Ohne dass wir Dinge oder Vorgänge als Zeichen abbilden, haben sie für uns keine Bedeutung, können wir kognitiv mit ihnen nicht umgehen, vermögen wir nicht einmal zu denken. Semiotisierung ist folglich notwendige Bedingung dafür, in der Welt sinnvoll handeln zu können (Nake 1997).

Dass der Gebrauch von Computern im Umgang mit Zeichen Sinn macht und einen Nutzen (Gebrauchswert) stiftet, ist mithin an mindestens drei Voraussetzungen gebunden: 1. muss sein Einsatzzweck in Zeichenprozessen genau bestimmt sein, 2. müssen die dazu erforderlichen geistigen Tätigkeiten formalisiert und algorithmisch beschrieben werden (Berechenbarkeit) und 3. müssen die durch den Systemeinsatz gestellten Handlungsanforderungen durch kollektive Aneignung der neuen Prozesse bewältigt werden. Dazu bedarf es der menschlichen Handlungskompetenz und Interpretationsleistung, die eigene Praxis zu reflektieren und darin Zeichenprozesse mit algorithmisch fassbaren geistigen Tätigkeiten zu bestimmen. Im praktischen Umgang mit dem Computer erhalten die verarbeiteten Zeichen durch den Handlungskontext ihren Sinn und stiften dessen Gebrauchswert; aufgrund der Berechenbarkeit ihrer Funktionen werden sie in der Maschine auf bloße Signale reduziert. So vermittelt die algorithmische Beschreibung zwischen Gebrauchswert und Maschine, zwischen Sinn und Signal.

Damit knüpfen wir an frühere, gut entwickelte Sichtweisen an, insbesondere an Perspektiven des Peirceschen relationalen Zeichenbegriffs (Eco 1976, Nöth 2000, Peirce 1983), des symbolischen Interaktionismus (Blumer 1969, Mead 1987, Mill 1998) und der technischen Semiotik (Andersen 1990, Nadin 1988, Nake 2001, Nake/Grabowski 2001), in denen wir freilich nicht stehen bleiben wollen. Die weiter gehende Analyse zeigt: Computersysteme werden entwickelt und benutzt, indem Aspekte

menschlicher Praxis im Lichte bestimmter Zwecke und aufgrund einer Verwandlung von Können in begriffliches Wissen modelliert werden (*Modellbildung*). Dabei werden grundsätzlich die folgenden Schritte der Trivialisierung und Formalisierung durchlaufen (Andelfinger 1997, Krämer 1988, Nake 1997):

- *Semiotisierung*: Abbildung des Wirklichkeitsausschnitts in Zeichen (als Voraussetzung für Reflexion und Kommunikation),
- *Formalisierung*: Abstraktion von kontextgebundenen Interpretationen (Verwendung standardisierter Zeichen),
- *Algorithmisierung*: Beschreibung von Handlungsabläufen als formal berechenbare Verfahren mittels der Standardzeichen (Wiederholbarkeit).

Formale Operationen beziehen sich dabei nur auf die Form und die Anordnung von Zeichen und bewirken deren physische Transformation ohne Ansehen ihrer Bedeutung. Softwaresysteme als Ausdruck dekontextualisierten Wissens lassen sich somit für alles Formalisierbare herstellen, und ihr Sinn muss im Gebrauch erst noch erschlossen werden. Durch Aneignung für ihren Gebrauch, d.h. durch Interpretation ihrer Funktionen im Handlungskontext, werden die Softwaresysteme wieder in einen – freilich eben dadurch veränderten – Praxiszusammenhang gestellt. So verkörpern sie dekontextualisierte formale Modelle und sind zugleich Teil des neuen Handlungskontextes einer veränderten Praxis. Diese Dialektik der Gestaltung von Form aus Reflexion von Praxis und der Aneignung von Form für veränderte Praxis verbindet die soziale Welt menschlicher Akteure mit der formalen Welt der Systeme.

Als *semiotische Maschinen* bilden Computer eine eigene Klasse von Maschinen, die sich von der Klasse der mechanischen, elektrischen, chemischen und biologischen Maschinen bzw. Prozesse grundsätzlich unterscheiden. Gemeinsam ist beiden Klassen technischer Artefakte zunächst ihre enge Verwandtschaft zur Sprache, indem sie wohl bestimmte Funktionen verkörpern, die durch Menschen in deren Handlungskontext zu interpretieren sind (die funktionale "Sprache" der Artefakte). Dabei sind die Wirkungen durch die Eingaben determiniert. Um sinnvolle Eingaben machen und die eingetretenen Wirkungen interpretieren zu können, müssen Handlungen in der funktionalen Sprache der Artefakte ausgedrückt werden. Das gilt zunächst für alle technischen Artefakte, vom Faustkeil bis zum Computer.

Die fundamentalen Unterschiede zwischen beiden Maschinenklassen liegen in deren Wirkbereichen, Funktionsweisen und Zwecken. Der *Wirkbereich* von Arbeits- und Kraftmaschinen sowie artifiziellen chemischen und biologischen Prozessen erstreckt sich auf die Natur und greift in Prozesse der Energie- und Stoffumwandlung ein, während der Wirkbereich semiotischer Maschinen ganz im Prozess der sozialen Interaktion liegt und auf die Verarbeitung von Daten in Zeichenprozessen zielt. Mit semiotischen Maschinen wird folglich die soziale Welt der Zeichenprozesse nirgends verlassen.

Dementsprechend beruht die *Funktionsweise* von Maschinen und Prozessen der Energie- und Stoffumwandlung auf natürlichen Effekten als Ergebnis von Naturerkenntnis und ihr *Zweck* ist die Nutzung der Naturkräfte. Die Funktionsweise semiotischer Maschinen beruht dagegen auf expliziten Handlungsvorschriften durch Formalisierung von Zeichenprozessen als Ergebnis von Kopfarbeit, und sie dient der Organisation und Koordination kollektiven Handelns. Die Funktionen werden realisiert durch Software, die damit doppelten Charakter annimmt: Einerseits ist sie (wenn auch schwierig) lesbarer Text, andererseits maschinell ausführbarer Operationscode, mithin eine Maschine. Software ist folglich Maschine und deren Beschreibung zugleich. Software vermittelt damit zwischen der Welt der Signale innerhalb semiotischer Maschinen und den in der Lebenswelt der Menschen gebrauchten Zeichen außerhalb.

Infolgedessen ist die interpretatorische Flexibilität technischen Handelns bei den Maschinen und Prozessen der Energie- und Stoffumwandlung gebunden an und beschränkt durch Naturbedingungen, während sie bei semiotischen Maschinen auf Gewohnheiten und Vereinbarungen beruht, die sich ändern können und zudem in Wechselwirkung mit den implementierten Handlungsvorschriften stehen. Als semiotische, in Zeichenprozessen operierende Maschinen gehen Computer eine innige Verbindung mit sozialen Systemen ein, indem sie einen Teil von deren kommunikativer Struktur bilden: Ihr Einsatz erfordert zunächst, auf angemessene, zum Handlungskontext passende Weise bestimmte Aspekte der sozialen Struktur – u.U. auch implizite Regeln – zu formalisieren. Im praktischen Umgang mit ihnen muss dann den darin angelegten Handlungsanforderungen genüge getan werden. Dabei bilden sich zugleich die Regeln einer neuen, systemunterstützten Praxis heraus, wodurch das kommunikative Handlungsfeld insgesamt neu strukturiert wird. Folglich unterliegen semiotische Maschinen allen Problemen der "doppelten Hermeneutik" von Zeichenprozessen in sozialen Systemen (d.h., dass das über die sozialen Systeme gewonnene und maschinell vergegenständlichte Wissen Teil ihrer Regeln und Ressourcen wird, die sich eben dadurch verändern; Giddens 1988). Sie erfordern mithin hohe Anstrengungen der Abstimmung und Koordination bei Einführung und Gebrauch sowie für die

Entwicklung gemeinsamer Handlungsrouinen. Insbesondere erheischt der Umgang mit semiotischen Maschinen die Entwicklung eines hinreichend geteilten Interpretationsrahmens sowie einer geteilten Begriffswelt unter den beteiligten Akteuren.

Heißt es im Falle stoff- und energieumwandelnder Maschinen, sich den Naturkräften und -effekten zu unterwerfen, sich ihrer naturbestimmten Funktionsweise zu fügen – oder auf ihr Mitwirken zu verzichten –, so ist im Falle semiotischer Maschinen deren Funktionsweise ganz und gar dem Spiel der Akteure überlassen, den vorgefundenen Spielregeln ihrer kollektiven Sinnggebung, Machtausübung und Wertschätzung (Signifikation, Domination, Legitimation sensu Giddens) anheim gegeben. Die semiotischen Maschinen zugrunde liegende Modellbildung, Formalisierung und Interpretation sind dabei lediglich Ausdruck der Regeln und Ressourcen kollektiven Handelns, die sie selektiv spiegeln und die sie im Gebrauch zugleich verändern, indem sie mit ihnen zu einer neuen Praxis verwachsen. Es ist daher nicht Zufall, sondern Konsequenz der Dynamik kollektiven Handelns und seiner Zeichenprozesse, dass Rationalisierungseffekte nur eintreten können, wenn sie ausdrücklich intendiert sind und kollektive Handlungsschemata, die sie hervorzubringen vermögen, explizit vereinbart und erprobt werden.

Zugespitzt gesagt: Computer sind *nur* semiotische Maschinen. Anders als stoff- und energieumwandelnde Maschinen und Prozesse bringen sie keine Naturkräfte ins Spiel, sondern sind auf Operationen in Zeichenprozessen der sozialen Interaktion beschränkt. So leisten sie im Prinzip nichts Neues, was nicht herkömmliche Berechnungsverfahren, Produktbeschreibungen, Verfahrensvorschriften oder Dokumentablagen auch schon bewirkt hätten. Was aber mit Computersystemen und ihrer weltumspannenden Vernetzung im Internet neu in die Welt gesetzt wurde, ist ein universal nutzbares *instrumentelles Medium*, das alle bisherigen Medien (Texte, Bilder, Audio, Video) integriert und zugleich Werkzeuge zur Bearbeitung der digitalen Objekte im Medium bereitstellt. Damit werden grundlegend neue Handlungsmöglichkeiten eröffnet: die Bildung virtueller Teams und virtueller Arbeitsräume, E-Mail-Kommunikation oder synchroner Ideenaustausch unter Abwesenden. Dabei werfen sie neue Gestaltungsaufgaben auf, so etwa die Gewährleistung von Privatheit oder von "Awareness" der digitalen Gegenstände und Werkzeuge, der Kooperationspartner und ihrer sozialen Beziehungen in virtuellen Arbeitsräumen, die Entwicklung geteilter Interpretationsschemata, allem voran freilich die Klärung der Frage nach der Angemessenheit der zur Formalisierung erforderlichen "Trivialisierung" von Kopfarbeit und ihren Zeichenprozessen.

Ihre besondere Funktionsweise als frei programmierbare semiotische Maschinen erlaubt es, sie universell, in beliebigen Zeichenprozessen mit trivialen, formalisierbaren Anteilen einzusetzen. Das Spektrum reicht von der Manipulation mathematischer Objekte, die von vornherein als explizit beschriebene Gegenstände von Kopfarbeit konstruiert sind (und von denen die Entwicklung semiotischer Maschinen ihren Anfang nahm), bis hin zur Maschinisierung beliebiger anderer Zeichenprozesse von Kopfarbeit, soweit sie trivialisiert und algorithmisch gefasst werden können. Von besonderer wirtschaftlicher Bedeutung sind indessen wertschöpfende Kopfarbeitsprozesse, auf die wir im folgenden unsere Aufmerksamkeit richten.

Dass Computer semiotische Maschinen sind (und damit eine besondere Klasse von Maschinen bilden), hat erhebliche Konsequenzen für die Realisierung und Beurteilung von Rationalisierungseffekten durch den Einsatz von IT-Systemen in Wertschöpfungsprozessen (und dass sie darin herkömmlichen Maschinen gewöhnlich gleich gesetzt werden, gehört zu den sich nicht abnutzenden Irrtümern von Informatikern und Systemanwendern). Im Falle von energie- und stoffumwandelnden Maschinen werden durch deren funktionale Gestaltung Naturkräfte genutzt, durch die höhere Mengenleistungen bei meist auch höherer Qualität erreicht werden können. Beispielsweise können durch Verwendung besserer Schneidstoffe und passende Maschinengestaltung die Schnittgeschwindigkeit spanender Werkzeugmaschinen und damit deren Stückausbringung verdoppelt werden – ein Produktivitätsfortschritt um 100%.

Ganz anders verhält es sich mit semiotischen Maschinen: Hier werden statt der Nutzung von Naturkräften in energetischen oder materiellen Prozessen Teile von Zeichenprozessen algorithmisiert, die ganz in der sozialen Sphäre von Kommunikation und Interpretation angesiedelt sind. Welche Wirkungen deren Einsatz zeitigt, hängt daher von Verlauf und Ergebnis komplexer und umfassender Verhandlungs- und Verständigungsprozesse darüber ab,

- welche Zeichenprozesse überhaupt handlungs- (bzw. geschäfts-)relevant sind,
- welche Teile davon sinnvoll formalisiert und algorithmisiert werden können ("das Triviale im Komplexen finden"), ohne Handlungsspielräume unangemessen zu beschränken,
- wie resultierende Zeichen (d.h. algorithmisch erzeugte Daten) im Handlungskontext zu interpretieren sind,

- wie sich Handlungsfelder verändern müssen, um Nutzen aus dem Computereinsatz zu ziehen.

Informationstechnik erweist sich damit als Medium des Organisierens. Um effektiv zu sein, müssen IT-System und Organisation in Ko-Konstruktion entwickelt und angeeignet werden (vgl. in ähnlicher Perspektive Dahlbohm/Mathiassen 1993, Floyd 1992). Ob der Einsatz eines IT-Systems ökonomischen Nutzen stiftet oder nicht, ist daher nicht in erster Linie eine Frage der Leistungsfähigkeit der semiotischen Maschine (es sei denn, es sind komplizierte, rechenintensive mathematische Objekte im Spiel), sondern der Art und Weise, wie sie konzipiert, eingesetzt und genutzt und wie damit die zugrunde liegenden Zeichenprozesse organisiert werden. Paradoxerweise stellt die effektive und effiziente Nutzung von IT-Systemen umso höhere Anforderungen an deren aufwendige Aneignung und die Entwicklung individueller und kollektiver Handlungskompetenz im Gebrauch, je komplizierter und funktionsreicher sie sind.

Insgesamt wirft damit die besondere Natur semiotischer Maschinen zwei Grundprobleme auf, die bislang, in Verkennung ihrer besonderen Natur als Medien des Organisierens, meist übersehen oder zumindest vernachlässigt wurden: das Problem der Trivialisierbarkeit von Kopfarbeit und ihrer Zeichenprozesse und das Problem der kollektiven Aneignung der neuen systemunterstützten Arbeitsweise. Es kommt darauf an, die richtigen Fragen zu stellen, denn falsche Fragen induzieren falsche Antworten und führen zu unbrauchbaren Systemen. Im Falle komplexer Kopfarbeit ist vor allem danach zu fragen, welche Anteile daran trivialisierbar sind (ohne den Handlungsspielraum einzuschränken), und im Falle trivialer Kopfarbeit geht es darum zu verstehen, wie deren Zeichenprozesse funktionieren (um sie modellieren zu können).

Wenn es beispielsweise durch geschickte Nutzung eines PPS-Systems – etwa aufgrund belastungsorientierter Auftragsfreigabe – gelingt, die Wiederbeschaffungszeit bestimmter Teile zu halbieren, tritt bei sonst gleich bleibenden Spielregeln zunächst die kontraproduktive Wirkung erhöhter Lagerbestände ein. Erst wenn auch die aus Sicherheitsgründen nötigen Mindestbestände und die Bestellmengen entsprechend reduziert werden, lassen sich geringere Lagerbestände realisieren.

Die belastungsorientierte Auftragsfreigabe (Wiendahl 1987) ist selbst ein gutes Beispiel für die Notwendigkeit, kollektiv ein ganz neues Verständnis von der Funktionsweise einer Werkstattfertigung zu entwickeln. Diese neue Sichtweise ist Voraussetzung dafür, in dieser Struktur mögliche Verkürzungen von Durchlaufzeiten zu realisieren, wozu das PPS-System bei passender Funktionalität nicht mehr beitragen kann, als die nötigen aktuellen Prozessdaten zu liefern (weitgehend unabhängig von seiner Rechenleistung).

Ein weiteres erhellendes Beispiel liefert die Erstellung einer Druckvorlage in der Buchproduktion. Die rechnerunterstützte Erstellung der Druckvorlage stellt wesentlich erhöhte Anforderungen an die wechselseitige Abstimmung und Koordination der Einzeltätigkeiten. Genügte es bei konventioneller Buchproduktion, die inhaltlich und sprachlich korrekten Manuskriptteile zum vereinbarten Zeitpunkt abzuliefern (um sie von einem Fachmann in einheitliche, ästhetisch ansprechende Form zu bringen), so müssen nun, um das IT-Medium effektiv und effizient nutzen zu können, zusätzlich auch zahlreiche formale Vereinbarungen getroffen werden, so beispielsweise

- Länge der Teile in Zeichen,
- Schreibweisen wiederkehrender Elemente (z.B. Literaturangaben und -verzeichnis),
- Formatangaben, Schrifttype und –größe (zur Vermeidung von Doppelarbeiten),
- Farbräume und deren vielfältige Parameter.

Die Rationalisierungseffekte hängen dabei fast ausschließlich von der Qualität und Zuverlässigkeit solcher Vereinbarungen ab und so gut wie nicht von der Leistungsfähigkeit des IT-Mediums. Jede Abweichung von den Vereinbarungen verursacht zusätzlichen Aufwand.

4 Naive Schönheiten: IT-Systeme in Wertschöpfungsprozessen

Organisationen sind Einheiten kollektiven Handelns. Sie werden gebildet, um für andere eine Leistung zu erbringen, um öffentlichen Nutzen zu stiften oder als Unternehmen Wertschöpfung mit Gewinn zu betreiben. Als soziale Systeme sind sie komplexe Gebilde und in der Lage, sich den Anforderungen ihrer Umgebung anzupassen.

Massenmärkte mit wenig Kundenorientierung und geringer Dynamik stellen geringe Anforderungen an Flexibilität und Kreativität des Handelns. Hier lassen sich Organisationen – wie der relative Erfolg des lange vorherrschenden Taylor-Modells der Produktion beweist – weitgehend trivialisieren. Leistungen werden arbeitsteilig nach genauen, theoretisch erfassten Vorschriften erbracht und kontrolliert. Arbeitsprozesse sind entsprechend in hohem Maße semiotisiert. Der Erfolg lebt von der Wiederholbarkeit der Leistungen. Derart hoch trivialiserte Organisationen bieten beste

Voraussetzungen für die Formalisierung und Algorithmisierung von Arbeitsprozessen. Folgerichtig ist die Vision der CIM-Fabrik Kulminationspunkt und Krönung des Taylor-Modells der Organisation.

Rasch wachsende Dynamik der Märkte und hohe Flexibilitätsanforderungen der Kundenorientierung verurteilen dieses Entwicklungsmodell indes zum Scheitern. Je dynamischer die Märkte und je vielfältiger die Kundenanforderungen, desto höher muss sich die Komplexität der Organisation einstellen. Und komplexe Organisationen lassen sich nur in Teilen trivialisieren. Eben hierin liegt das Problem: Zum einen ist Trivialisierung notwendige Voraussetzung für den Einsatz semiotischer Maschinen, zum anderen darf aber die notwendige Komplexität durch Trivialisierung nicht reduziert werden. So stehen komplexe Organisationen unausweichlich vor der überlebenswichtigen Frage herauszufinden, welche Teile formalisiert werden dürfen, ohne die notwendige Komplexität und die Möglichkeiten flexiblen und kreativen Handelns einzuschränken. An dieser Frage entscheidet sich, ob IT-Systeme das traurige Schicksal naiver Schönheiten ereilt (die gewöhnlich als "gefallene Mädchen" enden).

Die Dynamik der Märkte zwingt dazu, im Hinblick auf Wettbewerb und Kundenorientierung zwischen unterschiedlichen Kompetenzen der Wertschöpfung zu differenzieren: *Kernkompetenzen* sind die Fähigkeiten eines Unternehmens, die es im Wettbewerb von anderen unterscheiden. Mit diesen Fähigkeiten konkurriert das Unternehmen um Kunden und entwickelt seine Geschäftsfelder. Kernkompetenz kann nicht über eine ökonomische Schnittstelle zugekauft werden. Der Zulieferer hätte nur einen Kunden und damit keinen Markt. Wird es dennoch versucht, erhält man trotz größter Mühe nicht die gewollte Leistung.

Schalenkompetenzen sind demgegenüber Fähigkeiten, die sich allgemein, für viele Konkurrenten, als notwendig erwiesen haben. An die Schalenkompetenz stellen die Unternehmen einander ähnliche Anforderungen. Deswegen gibt es für Schalenkompetenzen Zulieferer. Diese Zulieferer sind Spezialisten für bestimmte, allgemein benötigte Leistungen. Sie konkurrieren untereinander und können deshalb eine höhere Qualität und niedrigere Kosten anbieten als die interne Herstellung.

In trägen (regulierten) Märkten ist es möglich, durch Steigerung der Wertschöpfungstiefe Kosten zu senken. Die dadurch bewirkte Reduktion der Beweglichkeit störte nicht, da die Märkte träge waren. Die Dynamik moderner Märkte kann so stark ansteigen, dass die Tiefe der Wertschöpfung bis auf die Kernkompetenz abgespeckt werden muss, um konkurrenzfähige Flexibilität zu erreichen. Je geringer die Leistungstiefe einer Wertschöpfung, um so leichter kann ihre Flexibilität gesteigert werden.

Das wirksamste Mittel, die Leistungstiefe zu reduzieren, ist das Auslagern von Prozessteilen aus dem Hauptprozess der kundenbezogenen Wertschöpfung. Die ausgelagerten Prozessteile müssen so an den Hauptprozess gekoppelt werden, dass die gewonnene Flexibilität nicht wieder reduziert wird. Ökonomische Schnittstellen sind dafür am besten geeignet. Sie können allerdings nur für Schalenkompetenz verwendet werden, sonst verschlechtern sich Qualität, Flexibilität und Kosten der Wertschöpfung.

Wächst die Marktdynamik, verliert ein Unternehmen darüber hinaus seine zentrale Steuerbarkeit. Es wird vom Markt dezentralisiert. Als Reflex auf die Marktdynamik bildet sich dezentrale Selbstorganisation, da sie wesentlich leichter mit der Marktdynamik umzugehen vermag. Dies geschieht ungefragt und automatisch, oft gegen den Willen der Zentrale ("heimliche" Transformation).

Die daraus erwachsende neue Funktion der zentralen Führung ist die innovative Störung der notwendig konservativen Selbstorganisation der marktnahen peripheren Bereiche. Im Gegensatz zur Dezentralisierung von Leistungsprozessen geschieht diese Zentralisierung von Innovation nicht von alleine; sie ist Ergebnis von bewusster Entwicklung oder sie bleibt aus. Die für ein Unternehmen notwendige Innovation kann nicht in selbstorganisierte Wertschöpfung integriert werden. Sie muss zentralisiert werden und von dort aus als dosierte konstruktive Störung auf die Peripherie einwirken (vgl. Kap. 3).

Ein Teil dieser bewussten Entwicklung ist eine neue Arbeitsteilung zwischen zentraler Datenverarbeitung (DV) und peripheren Arbeitsprozessen, besonders was die wertverbrauchenden Dienstleistungen betrifft. Der nicht-innovative Alltag dieser informationstechnischen Dienstleistung muss in die selbstorganisierte Wertschöpfung der Peripherie verlagert und dort integriert werden. Das fördert und erleichtert die dortige Selbstorganisation und vermeidet die Zuordnung von Kosten durch aufwendige Kostenrechnung. Die dafür notwendige Qualifikation muss durch Führung und durch technologische Innovation geschaffen werden (vgl. Abb. 3).

Innovation von Produkten und Prozessen kann von der Peripherie nicht geleistet werden, weil dadurch eingespielte und bewährte Prozesse gestört werden. Deshalb muss Innovation als strategische Aufgabe zentralisiert werden. Ohne diese Zentralisierung kann in der Peripherie keine

Selbstorganisation entstehen. Das bezieht sich nur auf die Funktion: Ein Mitarbeiter der Peripherie kann natürlich innovativ denken und handeln, beispielsweise kann er den Anstoß geben für Projekte zur Erneuerung von IT-Systemen.

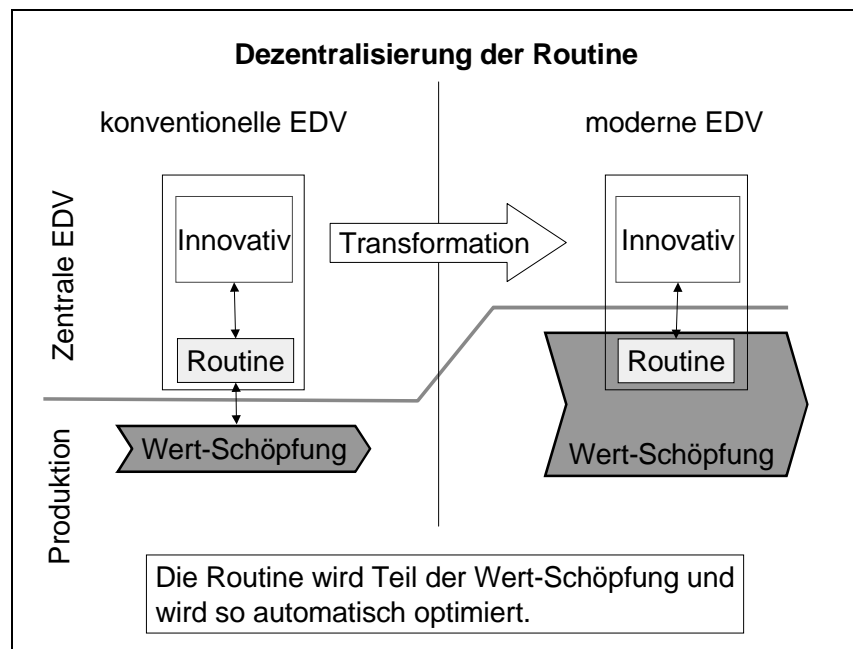


Abb. 3: Verlagerung von Alltagsroutinen aus der zentralen DV in die periphere Wertschöpfung
 Damit der moderne Teil der Organisation, der in aller Regel unbewußt als Reaktion an der Peripherie auf den zunehmenden Marktdruck entstanden ist, dauerhaft in einer Organisation etabliert werden kann, bedarf es eines "Schutzraumes", in dem solche neuen, und damit in aller Regel auch empfindlichen Organisationsteile wachsen können. Diese Schutzräume können u.a. hergestellt werden durch:

- einen Sponsor: ein Manager, der die notwendigen Ausnahmen durch Machteinsatz schützen kann,
- ein Bündnis mit dem Kunden, der von der Führung die dauerhafte Existenz der für ihn interessanten Teile einer Organisation einfordert (mit der Drohung des Wechsels),
- Zusammenrottung als Methode: Gründung einer "Community" der Experten.

Insgesamt sind bei dieser grundsätzliche Reorganisation der Wertschöpfung allerdings die aus dem Erhalt der notwendigen Komplexität der Organisation resultierenden "Tabuzonen" für den Einsatz von IT-Systemen zu beachten. Grundsätzlich sind alle Bereiche hoher Kreativität und damit hoher Selbstorganisation solche Tabuzonen. Im Zuge der Reorganisation verändern sie sich nach Lage und Umfang. Zwei Typen von Tabuzonen sind dabei zu unterscheiden.

Die erste Tabuzone – der Konflikt: Ein Konflikt ist ein Bereich hoher, nach innen gerichteter Kreativität. Er erzeugt ein Subsystem hoher und sehr stabiler Selbstorganisation. Diese Tabu-Zone ist schon lange bekannt. Trotzdem wird immer wieder versucht, vorhandene Konflikte mit der geliehenen "Autorität der EDV" zu befrieden. Der Effekt ist fast immer die "Elektrifizierung" des Konflikts und damit seine Fortsetzung auf höherer Ebene. Erst wenn es gelingt, destruktive Konflikte auf das Niveau einer kreativen Spannung (z. B. zwischen Vertrieb und Fertigung) zu bringen, kann DV nützlich sein, weil sie aus trivialisierten Routinen gewonnenes kodifiziertes Wissen verfügbar macht.

Die zweite Tabuzone – kreative Arbeit: Diese Zonen sind Bereiche hoher, nach außen gerichteter Kreativität. Die entsprechende Selbstorganisation bildet sich dort, wo sich konkurrierende Unternehmen durch die Enge des Marktes "nahe" kommen. So nahe, dass die Konkurrenten auch dezentral zu spüren sind. Dieser wirtschaftliche "Nahkampf" ist nur mit hoher dezentraler Kreativität zu führen. Er kann nicht tayloristisch formalisiert werden.

Im Vergleich zu den trivialen Bereichen der Organisation sind die kreativen Tabuzonen oft klein und unscheinbar und von Schlendrian schwer zu unterscheiden. Besonders dann, wenn sie durch "heimliche" Transformation entstanden sind. Trotzdem sind sie überlebenswichtig. In ihnen werden die Kernkompetenzen und Alleinstellungen erzeugt, mit denen das Unternehmen konkurrenzfähig bleibt. Werden diese Zonen mit DV-Systemen "belästigt", entstehen die Software-Havarien.

DV-Systeme, die am komplexen Ende eines Problems eingesetzt werden dürfen, müssen neutral sein. Das heißt, sie bilden die zu lösende Aufgabe und damit verbundene Arbeitsabläufe nicht ab. Eine Zeitung hat beispielsweise die Aufgabe, gute Artikel zu schreiben. Ein Ingenieurbüro hat die Aufgabe, gute Dokumentationen seiner Entwürfe zu schreiben. Ein Dichter möchte gute Bücher schreiben. Allen diesen Aufgaben ist gemeinsam, dass es um die Erstellung qualitativ anspruchsvoller Texte geht. Die Lösung sind Menschen, die schreiben können. Auch ein "intelligentes" Textverarbeitungssystem ist keine Lösung, ist aber in allen Kontexten anwendbar. In diesem Sinne sind auch verschiedenste Probleme an ihrem komplexen Ende ähnlich: So muss unterschieden werden zwischen DV-Systemen, die bestimmte triviale Arbeitsaufgaben und -abläufe modellieren und daher notwendig verschieden sind, und solchen, die bei komplexen Aufgaben als neutrales, einheitliches Medium genutzt werden (vgl. Abb. 4).

Am komplexen Ende sind auch verschiedenste Kompetenz-Themen einander ähnlich

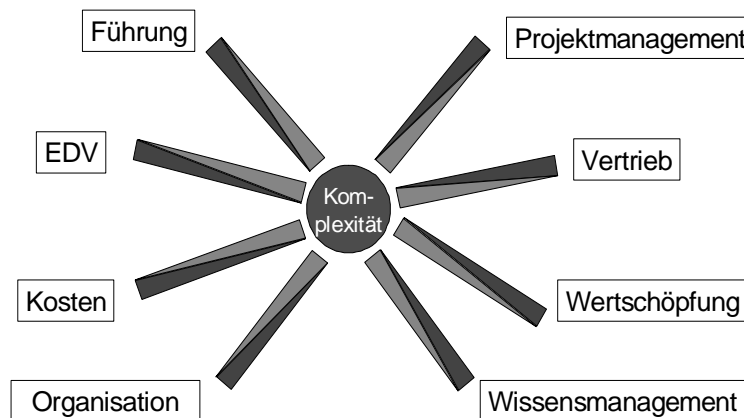


Abb. 4: Ähnlichkeiten und Unterschiede der Eigenschaften von DV-Systemen in komplexen und trivialen Einsatzfeldern

Die Informatik leidet darunter, dass ihren Anwendungen die Lösung fast aller Probleme zugetraut wird. Wenn die Informatik darauf hereinfällt, wird sie zur Ursache von Havarien. Um dies zu vermeiden, muss zunächst das Bild von der Organisation durch die Brille der kollektiven Reflexion "scharf" gestellt werden. Das erfordert, die Organisation nach Möglichkeit in trivialisierbare und nicht trivialisierbare Bestandteile und Prozesse zu zerlegen. Die letzteren gefährden den Erhalt notwendiger Komplexität, falls sie in das Prokrustesbett von IT-Systemen gezwängt werden sollen. Für die trivialen und formalisierbaren Teile können dagegen konventionelle Verfahren und Systeme angewendet werden. Die Struktur der Arbeitsprozesse wird dabei mit der Struktur eines IT-Systems zur Deckung gebracht und fest verkoppelt. Eine Havariegefahr ist hier sehr gering.

Für den Einsatz in komplexen Teilen der Organisationen sind nur IT-Systeme erlaubt, die lediglich als Medium der Kommunikation und Kooperation und des interaktiven Umgangs mit explizitem Wissen gebraucht werden und nicht mit dem Arbeitsprozess verkoppelt werden können. Dabei handelt es sich um so genannte neutrale Systeme wie zum Beispiel Text-, Hypertext- oder Mail-Systeme. Freilich gilt auch für deren produktive Aneignung das oben über semiotische Maschinen Gesagte, insbesondere die Notwendigkeit der Entwicklung neuer Spielregeln im Umgang mit dem neuen Medium.

5 Folgerung: Wie sich Havarien vermeiden lassen

Als Fazit der hier angestellten Überlegungen lässt sich festhalten: IT-Systeme können auch in den kreativen Tabuzonen eingesetzt werden, freilich nur dann, wenn sie den nachstehenden Merkmalen genügen.

Neutralität statt Flexibilität: Selbstorganisation und konventionelle DV sind Feinde.

Auch die flexibelste DV kann nicht auf Überraschungen reagieren. Wird Organisation mit DV verbunden (kausal gekoppelt), so kann auch die Organisation nicht mehr auf Überraschung reagieren. Die Selbstorganisation kann sich unter solchen Umständen entweder gar nicht oder nur neben der DV entwickeln. Je "flexibler" die DV-Systeme gebaut sind, desto leichter können sie mit Arbeitsprozessen

verkoppelt werden und desto kompromißloser kann die Selbstorganisation unterbunden werden. DV läßt Selbstorganisation nur dann "ungeschoren", wenn sie mit diesen Anteilen nicht verkoppelt wird, wenn sie neutral bleibt wie Kugelschreiber, Stempelkissen oder Text-Verarbeitung und CAD-Systeme.

Software, die an Organisation "angepaßt" werden kann, ist wie flüssiger Beton. Sie wird in die Organisation "gegossen" und härtet dort aus. Danach kann sich die Struktur der Organisation nur noch dort bewegen, wo sie sich schon während der Aushärtung bewegt hat. Neutrale Software ist dagegen schon ausgehärtet, bevor sie in die Organisation eingeführt wird.

Werkzeug-Charakter: Ein Werkzeug nimmt man zur Hand, um ein Ziel zu erreichen. Wenn dies mißlingt, legt man es wieder beiseite und sucht nach einer anderen Lösung. Auch DV-Systeme für den Einsatz im komplexen Bereichen der Organisation müssen diese Eigenschaft haben, sonst ist auch in Ausnahmesituationen, im Umgang mit Überraschungen, kreatives Arbeiten nicht möglich.

Medium statt Isolation: Ein DV-System kann Kreativität und Fehler nicht unterscheiden. "Fehlervermeidende" IT-Systeme schließen die Kreativität der Benutzer aus. Sie vertragen deren Dummheit, aber nicht ihre Kreativität. Eine gute Axt muß scharf sein, darf aber nicht von selbst schneiden.

Datenhoheit statt Integration: Datenhoheit bedeutet, daß Daten immer einen verantwortlichen "Besitzer" haben. Mit ihm kann Inhalt und Qualität der Daten vereinbart werden.

Integrierte betriebswirtschaftliche IT-Systeme etwa vom Typus ERP zwingen dagegen zur Verwendung anonymer Daten ohne Besitzer, für die nur eine zentrale Macht die Verantwortung übernehmen kann. Für die Konfliktfälle ist eine operative Steuerung durch die Zentrale unvermeidlich. Integrierte betriebswirtschaftliche Standardsysteme sind nur in tayloristischen Organisationen verwendbar, das heißt bei niedrigem kreativem Anteil in der Wertschöpfung, weil Daten und Abläufe mit den Arbeitsprozessen engstens verkoppelt sind. Es handelt sich hier um eng gekoppelte Systeme im Sinne von Perrow (1989), die infolge unbewältigter Überraschungen leicht in Katastrophen führen können.

Diese Prüfkriterien betreffen indes nur das Einsatzfeld und den Einsatzzweck der IT-Systeme. Darüber hinaus müssen sich, wie wir gezeigt haben, die in der Organisation zusammenwirkenden Akteure darüber klar werden, welche Handlungsanforderungen der Gebrauch der Systeme an deren Benutzer stellt. Ihnen haben sich die Benutzer zu unterwerfen, wenn sie produktiven Gebrauch davon machen wollen. Und schließlich müssen sie die aufwendigen Lernprozesse der kollektiven Aneignung organisieren, in denen die Benutzer lernen, wie sie mittels der Systemfunktionen ihre Aufgaben im Organisationszusammenhang bewältigen, den Systemen mithin für ihre Arbeit Sinn verleihen können, und in denen sie die Regeln produktiven Umgangs mit den Systemen entwickeln und einüben. Produktivität erwächst aus der Qualität dieser Prozesse organisationaler Veränderung, nicht, wie so häufig angenommen, aus der Leistungsfähigkeit der IT-Systeme. Kurzum: Wer IT-Systeme produktiv nutzen will, muss Organisationsentwicklung betreiben.

Zum Schluss: Für eine Theorie der Informatik-Anwendungen

Aus den hier angestellten Überlegungen lassen sich im Hinblick auf Theorieentwicklung zumindest drei Schlussfolgerungen ziehen. Erstens verweisen der weithin unproduktive Einsatz von IT-Systemen, die hartnäckig wiederkehrenden Software-Havarien und die bis heute ausbleibende Remedur darauf, dass die Natur und Wirkungsweise der datenverarbeitenden Maschine noch immer nicht recht verstanden werden. Das deutet auf einen Mangel an Theorie hin.

Zweitens kann in einer solchen Situation nur eine Theorie weiterhelfen, die diese Schwierigkeiten zu erklären vermag. Erst dann können Ansätze getestet werden, wie sich die Schwierigkeiten überwinden lassen. Eine Theorie der Anwendung informationstechnischer Systeme ist daher nötiger denn je. Gebraucht wird eine einheitliche Theorie, die IT-Systeme als handlungsunterstützende technische Artefakte und Organisationen als soziale Systeme zielorientierten kollektiven Handelns zugleich zu beleuchten vermag, vor allem also das Zusammenspiel von Organisationsentwicklung und Systemgestaltung und -gebrauch. Mit anderen Worten: Zu verstehen ist die wirkungsvolle Einbettung dieser Systeme in das kollektive Handeln in Organisationen.

In diesem Sinne erweist sich drittens die hier skizzierte theoretische Perspektive, IT-Systeme als semiotische Maschinen zu begreifen, als nützlich und fruchtbar, weil sie die Schwierigkeiten ernst nimmt und sie zu erklären sucht. Insoweit das gelingt, lohnt es sich, sie weiter zu entwickeln.

Literatur

Andelfinger, U., 1997: Diskursive Anforderungsanalyse. Ein Beitrag zum Reduktionsproblem bei Systementwicklungen in der Informatik, Frankfurt/M: Peter Lang.

- Andersen, P. B., 1990: A Theory of Computer Semiotics. Semiotic approaches to construction and assessment of computer systems, Cambridge: Cambridge University Press.
- Blumer, H., 1969: Symbolic Interactionism. Perspective and Method, Berkeley (CA): University of California Press.
- Brödner, P., 2002: Der Held von Caputh steht nicht allein. Wie Wissenschaft die Nutzungsprobleme der Informationstechnik ignoriert, in: Moldaschl, M. (Hg.): Neue Arbeit – Neue Wissenschaft der Arbeit? Heidelberg: Asanger 2002, 339-364.
- Brödner, P., 1997: Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen, Berlin: edition sigma.
- Brynjolfsson, E., 1993: The Productivity Paradox of Information Technology, CACM 36, No. 12, 67 - 77
- Brynjolfsson, E.; Hitt, L. M., 2001: Beyond Computation: Information Technology, Organizational Transformation and Business Performance, in: Greenan, N.; Lhorty, Y.; Mairesse, J. (Eds.): The Puzzling Relations Between Computer and the Economy, MIT Press.
- Coy, W., 1995: Automat - Werkzeug - Medium, Informatik Spektrum 18/1, 31 – 38.
- Coy, W., 1989: Brauchen wir eine Theorie der Informatik? Informatik Spektrum 12, 256-266.
- Dahlbom, B.; Mathiassen, L., 1993: Computers in Context. The Philosophy and Practice of Systems Design, Oxford: Blackwell.
- Davenport, T. H., 1998: Putting the Enterprise into the Enterprise System, Harvard Business Review July-August, 121 – 131.
- Eco, U., 1976: A Theory of Semiotics, Bloomington (IN): Indiana University Press.
- Floyd, C., 1992: Software Development as Reality Construction, in: Floyd, C.; Züllighoven, H.; Budde, R.; Keil-Slawik, R. (Eds.): Software Development and Reality Construction, Berlin Heidelberg New York: Springer, 86 – 100.
- Giddens, A., 1988: Die Konstitution der Gesellschaft. Grundzüge einer Theorie der Strukturierung, Frankfurt/M: Campus.
- Krämer, S., 1988: Symbolische Maschinen. Die Idee der Formalisierung in geschichtlichem Abriss, Darmstadt: Wissenschaftliche Buchgesellschaft.
- Landauer, T. K., 1995: The Trouble with Computers. Usefulness, Usability, and Productivity, Cambridge (MA): MIT Press.
- Maucher, I., 2001: Komplexitätsbewältigung durch Entwicklung und Gestaltung von Organisation, München: Hampp.
- Maucher, I. (Hg.), 1998: Wandel der Leitbilder zur Entwicklung und Nutzung von PPS-Systemen, München: Hampp.
- McKinsey Global Institute, 2001: Productivity in the United States, <http://www.mckinsey.com/knowledge/mgi/reports/productivity.asp>
- Mead, G. H., 1987: Wissenschaft und Lebenswelt, in: Gesammelte Aufsätze, hg. von Hans Joas, Bd. 2, Frankfurt/M: Suhrkamp 1987, 14-87.
- Mill, U., 1998: Technik und Zeichen. Über semiotische Aktivität im technischen Kontext, Baden-Baden: Nomos Verlagsgesellschaft.
- Nadin, M., 1988: Interface Design: A Semiotic Paradigm, Semiotica 69, 269-302.
- Nake, F., 1997: Der semiotische Charakter der informatischen Gegenstände, in: Bayer, U. (Hg.): Festschrift zu Ehren von Elisabeth Walter, Baden-Baden: agis.
- Nake, F., 2001: Das algorithmische Zeichen, in: Bauknecht, W.; Brauer, W.; Mück, T. (Hg.): Informatik 2001. Tagungsband der GI/OCG Jahrestagung, 736-742.
- Nake, F.; Grabowski, S., 2001: Human-Computer Interaction Viewed as Pseudo-Communication, Knowledge-Based Systems 14, 441-447.
- Nöth, W., 2000: Handbuch der Semiotik, 2. Aufl., Stuttgart: Metzler.
- OECD 2001: Knowledge, Work Organization and Economic Growth, DEELSA/ELSA (2001)2, Febr. 7.
- Oz, E., 1994: When Professional Standards Are Lax. The CONFIRM Failure and its Lessons, CACM 37 October, 29-36.
- Peirce, C. S., 1983: Phänomen und Logik der Zeichen, Frankfurt/M: Suhrkamp.
- Perrow, C., 1989: Normale Katastrophen. Die unvermeidbaren Risiken der Großtechnik, Frankfurt/M: Campus.
- Scott, E.J.; Vessey, I., 2002: Managing Risks in Enterprise Systems Implementations, CACM 45 April, 74-81
- Solow, R.M., 1987: "We'd Better Watch Out", New York Times Book Review, July 12: 36.
- Sommerville, I., 2001: Software Engineering, 3rd ed., Wokingham Reading: Addison-Wesley.
- Stewart, T. A., 1997: Intellectual Capital. The New Wealth of Organizations, New York: Doubleday.
- Wiendahl, H.-P., 1987: Belastungsorientierte Fertigungssteuerung - Grundlagen, Verfahrensaufbau, Realisierung. München Wien: Hanser.

Arbeitsgruppe
Informatik als Hybridwissenschaft

Informatik als Hybridwissenschaft.

Selbstverständnis, Ausbildung, Ethik, Theorie

Aufruf zur Arbeitsgruppe

Dirk Siefkes, Technische Universität Berlin

In einem Interdisziplinären Forschungsprojekt "Sozialgeschichte der Informatik" der TU Berlin haben wir unsere wissenschaftlich-technische Disziplin, insbesondere die Programmierung, als kulturelle Entwicklung untersucht (SGI97-99, Eul99, Sta01, Sie99). Von Anfang an ist das Problem beim Programmieren gewesen, die gegensätzlichen Welten von Mensch und Maschine zusammenzubringen (Brö97). Zunächst schienen beide kaum getrennt, da es nur um numerische Berechnungen ging. Mit den Formularen der Rechenbüros waren die mathematischen Verfahren leicht in Maschinenbefehle zu "kodieren", Computer wurden als direkte Nachbauten menschlicher Rechner an der Tafel oder an der Tischrechenmaschine gesehen (Zuse 1936, von Neumann 1945, Walther 1953). (Kritisch reflektiert wurde die Analogie zwischen Neuronen und Schaltelementen, also zwischen Gehirn und Maschine (Wiener 1948, von Neumann 1948). Turings theoretische Maschinen (1936) beeinflussten den Rechnerbau nicht, produzierten aber bei von Neumann (1948) und ihm selbst (1950) die ersten Phantasien von selbstreproduzierenden Automaten bzw. denkenden Maschinen.) Bald wurde mehr Formalismus nötig, um die länger werdenden Wege gehen zu können: Flußdiagramme und indirekte Adressierung erlaubten, komplexere Algorithmen effizienter auszuführen (Goldstine, von Neumann 1947/8). Kodierung wurde zur Geheimwissenschaft derer, die mit den Problemen mathematischer Optimierung und den Tücken der Maschine gleich gut umgehen konnten. Mit "automatischer Kodierung" wurde mehr und mehr von dieser Kunst der Maschine übertragen. Mit der Entwicklung der ersten "Programmiersprachen" wurde schließlich ein eigener Formalismus eingeführt, den die Mathematiker lesen und schreiben und die Maschinen selber in ausführbaren Code transformieren konnten (Fortran 1957). Schon Algol (1960) war als "rechnerunabhängig" konzipiert, es sollte vor allem die Kommunikation der Mathematiker über "rechnerausführbare" Verfahren erleichtern. Damit wurde die Sprach- von der Rechnerentwicklung getrennt, zunehmend wurden Formalismen aus Mathematik und Logik zur Programmierung und ihrer Unterstützung benutzt. Der Computer wird aus einer Maschine, die zu bedienen ist, zu einem Werkzeug, das man handhabt, um Anwendungen zu gestalten. Gleichzeitig treten Bilder neben Wörter als Mittel der "Mensch-Maschine-Kommunikation". So wie der Rechner (als Mensch *und* als Maschine) bei Turing Symbole (und nichts sonst) manipuliert, manipuliert er jetzt icons auf dem Bildschirm, um Aktionen auszulösen und zu steuern. Daß im Hintergrund eine Maschine läuft, ist kaum noch bewußt. Visuelle und motorische Schemata werden direkt angeregt, ohne Umweg über das Kognitive; und regen direkt zum Handeln an.

Auf der sprachlichen Seite treffen sich die beiden Entwicklungen in der objektorientierten Programmierung. Der Anwendungsbereich wird durch interagierende Objekte auf dem Bildschirm dargestellt, die mathematisch beschreibbar sind, aber wie reale Objekte erscheinen. Die Hybridisierung ist so weit getrieben, daß nicht nur menschliche und maschinelle Realität aus dem Blick verschwinden, sondern auch die sie verbindende der abstrakten Datenstrukturen. – Auf der bildlichen Seite kulminieren die beiden Entwicklungen im Internet. Der Cyberspace kennt nur eine Realität, die virtuelle, so wirkmächtig wie scheinbar. Der Computer wird dabei aus der Maschine zum Medium (Sch97); aber was gibt es in einer einheitlich hybridisierten Welt zu mediatisieren?

Aus Zeichen, die Menschen verstehen und die sie zum Handeln anregen, werden im Computer Signale, die Aktionen auslösen. Aus Kommunikation wird Datenübertragung, aus Denken Datenverarbeitung, aus Rechenautomaten werden Datenverarbeitungsanlagen oder gar Elektronengehirne. Und wie das menschliche Gehirn souverän den menschlichen Körper und seine Aktivitäten zu beherrschen scheint, können Computer beliebige Maschinen steuern. Im Computer übertragen wir also scheinbar beliebige menschliche Aktivitäten auf die Maschine, indem wir ihnen eine "Zeichenhaut" überziehen (Nake in Coy92, NRS01; Nak93). Wer Computer entwerfen, programmieren oder benutzen will, muß menschliche Aktivitäten, insbesondere geistige, in Maschinenläufe umzeichnen und umgekehrt. Im Umgang mit dem Computer *hybridisieren* wir die gegensätzlichen Welten von Menschen und Maschinen mit Hilfe formaler Ausdrücke (SGI97-99, Eul99, Sta01, Sie99,01).

Denken und arbeiten, lernen und Erfahrungen machen können wir nur in der menschlichen Welt. Beim Umgang mit Computern übertragen wir unsere Vorstellungen in Zustände und Abläufe von Maschinen und umgekehrt maschinelles Vorgehen in unsere Erfahrungswelt. Der Weg geht durch die Zwischenwelt der Zeichen: Algorithmen und gewisse allgemeinere formale Anweisungen sind von

Maschinen ausführbar; gleichzeitig sind formale Beschreibungen von Menschen nachvollziehbar. Wir benutzen also formale Ausdrücke, um Maschinen in Gang zu setzen und um Maschinenvorgänge zu verstehen; eins geht nicht ohne das andere. Als Informatiker beschäftigen wir uns meist mit der Symbolwelt, die Maschinen- und die Menschenwelt geraten in den Hintergrund (Flo92, Nau92). Es gibt aber auch andere Vorlieben und Blindheiten; wie das Bild ausfällt, hängt von unserer Beziehung zu den drei Welten ab. Sicher sind sie nicht gleichberechtigt: Wir produzieren Maschinen und Formalismen zu unserem Nutzen, nicht als Selbstzweck. Aber gelingen kann die Hybridisierung nur, wenn wir alle drei Welten im Blick und miteinander in Beziehung halten. Beim Umgang mit dem Computer müssen wir mit den Maschinen und Formalismen auch uns selber entwickeln.

Hybridisierung ist also kein Fluch, der auf der Informatik lastet, und keine Zauberkunst, die sie auszeichnet, sondern eine Aufgabe, der sie sich zu stellen hat. Daher ist die Informatik auf besondere Weise auf interdisziplinäre Arbeit angewiesen. Wie im Umgang mit dem Computer gegensätzliche Welten, müssen in der wissenschaftlichen Arbeit unterschiedliche Disziplinen hybridisiert werden, insbesondere geistes- und sozialwissenschaftliche mit ingenieur- und naturwissenschaftlichen. Nicht nur in der Softwaretechnik verlangt das gleichzeitig Offenheit, Bescheidenheit und kritische Distanz aller beteiligten Disziplinen. Das übliche Abschotten aus Arroganz oder Resignation oder Zeitmangel kann nicht einfach durch unbedenklichen Import und Export von Wissen oder durch das Einrichten von Brückenköpfen in sonst als feindlich behandelten Bereichen ersetzt werden. Die Hybridisierung gelingt nur in der gemeinsamen Arbeit. Wenn die Wissenschaftler dabei lernen, die Erfahrungen der anderen für sich nutzbar zu machen, können sich die unterschiedlichen Menschen und ihre Welten gemeinsam entwickeln. Sicher wird es dann weniger große Projekte geben, die die Welt ändern und unsere Karriere und die Disziplin voran bringen. Aber wir können kleinere durchführen, die wirklich helfen.

Wir werden die Ausbildung ändern müssen. Neben Technik und Mathematik brauchen wir Kenntnisse und Fähigkeiten aus den Geistes- und Sozialwissenschaften, nicht als Schmuck, sondern als drittes Standbein. Dafür müssen wir – Studenten wie Dozenten – keine Universalgenies werden. Wenn Dozenten herausfinden wollen, welche Rolle ihr Spezialgebiet im Prozeß der Hybridisierung spielt, müssen und können sie die engen Grenzen der Informatik überschreiten und lernen, fremde Wege zu gehen, und das den Studenten beibringen. Das ist nicht schwerer oder leichter, als im eigenen Gebiet immer tiefer zu bohren, erfordert aber ein Umdenken – auch derjenigen, die Wissenschaft weiter wie bisher betreiben. Unsere Curricula sind mit der Disziplin gewachsen und haben zu viel Material angesammelt, das nur historischen Wert hat. Es reicht aus, wenn Studenten Programmieren als Hybridisierungsaufgabe lernen; dabei lernen sie alles, was sie wissen müssen. Informatikgeschichte ist dabei nicht zusätzlicher lästiger Stoff, sondern liefert einen alternativen Zugang zur Informatik (Kor02, Sie02). Der Grund, den wir jetzt im Grundstudium legen, behindert eine wissenschaftliche Ausbildung. Ein Studiengang ist kein Tempel, der auf Betonfundamenten steht; Wissen wächst nur auf offenem Boden (Sie93).

Wir werden das Verhältnis zu unserer Arbeit ändern müssen. Wir sind gewohnt, Verantwortung für unser Tun abzulehnen, weil Programme für alles benutzt werden können: Die beste Idee kann zur schlimmsten Anwendung führen, weil wir ihre Folgen nicht absehen. Das ist aber kein Grund, sich verantwortungslos zu verhalten. Wenn wir jederzeit – mit Kollegen und "Betroffenen", mit Mitarbeitern und Freunden – über unsere Arbeit reden und wohin sie führen könnte, verstehen wir besser, was wir wirklich wollen, was wir durch Hybridisierung bezwecken und verstecken wollen. Wir lernen, Verantwortung als Herausforderung und nicht als Bürde zu betrachten (Rol98, Sie92, Siefkes in Coy 92).

Wir müssen eine breitere Theorie der Informatik entwickeln (Coy92, Siefkes in Fre97, NRS01). Die aus der Mathematik übernommene Theoretische Informatik hilft Programme und andere formale Werkzeuge zu verstehen. Um Hybridisierung zu verstehen, müssen wir nicht nur in der Lage sein, menschliche Aktivitäten in unterschiedlichen Bereichen zu formalisieren und zu maschinieren und die automatisierten Systeme so in menschliche Umgebungen zurück zu bringen, daß sie funktionieren. Beide Unterfangen – verharmlosend De- und Rekontextualisierung genannt – verändern die Umgebungen, insbesondere die beteiligten Menschen. Eine soziale Umgebung ist kein Text, den man mit "cut and paste" editieren kann. Auch die Theorien der Nachbardisziplinen genügen nicht für eine theoretische Grundlegung der Informatik; wir müssen auf beiden Seiten des Grabens stehen. Eine Theorie der Informatik muß die Disziplin mit anderen verknüpfen und so Informatik als Hybridwissenschaft erweisen.

Literatur

[Brö97] Brödner, Peter 1997: Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen. Berlin: edition sigma.

- [Coy92] Coy, Wolfgang et al. (Hrsg.) 1992: Sichtweisen der Informatik. Vieweg.
- [Eul99] Eulenhöfer, Peter 1999: Die formale Orientierung der Informatik. Zur mathematischen Tradition der Disziplin in der Bundesrepublik Deutschland. Diss., FB Informatik, TU Berlin.
- [Flo92] Floyd, Christiane et al. (eds.) 1992: Software Development and Reality Construction. Springer.
- [Fre97] Freksa, Christian et al. (eds.) 1997: Foundations of Computer Science – Potential, Theory, Cognition. Springer.
- [Kor02] Korb, Joachim et al. 2002: Geschichte als Zugang zur Informatik. Bericht zu einem Studienreformprojekt. TU Berlin, Fakultät Elektrotechnik und Informatik, Bericht 02-15.
- [Nak93] Nake, Frieder (Hrsg.) 1993: Die erträgliche Leichtigkeit der Zeichen. Ästhetik Semiotik Informatik. Agis.
- [NRS01] Nake, Frieder, Arno Rolf, Dirk Siefkes (Hrsg.) 2001: Informatik - Aufregung zu einer Disziplin. Tagung Heppenheim 2001. Universität Hamburg, FB Informatik, Bericht 235.
- [Nau92] Naur, Peter 1992: Computing - a Human Activity. ACM Press, Addison-Wesley.
- [Rol98] Rolf, Arno 1998: Grundlagen der Organisations- und Wirtschaftsinformatik. Springer.
- [Sch97] Schelhowe, Heidi 1997: Das Medium aus der Maschine. Zur Metamorphose des Computers. Campus.
- [SGI97] Eulenhöfer, Peter, et al. 1997: Die Konstruktion von Hybridobjekten als Orientierungsmuster in der Informatik. TU Berlin, FB Informatik, Bericht 97-23.
- [SGI98] Siefkes, Dirk, et al. (Hg.) 1998: Sozialgeschichte der Informatik. Kulturelle Praktiken und Orientierungen. Deutscher Universitätsverlag.
- [SGI99] Siefkes, Dirk et al. (Hg.) 1999: Pioniere der Informatik. Ihre Lebensgeschichte im Interview. Interviews mit Bauer, Floyd, Weizenbaum, Wirth, Zemanek. Springer.
- [Sie92] Siefkes, Dirk 1992: Formale Methoden und kleine Systeme. Lernen, leben und arbeiten in formalen Umgebungen. Vieweg.
- [Sie93] - " - 1993: Evolutionäre Modelle in der Informatik. TU Berlin, FB Informatik, Bericht 93-15.
- [Sie99] - " - 1999: Die Rolle von Schemata in der Informatik als kultureller Entwicklung. TU Berlin, FB Informatik, Bericht 99-6.
- [Sie01] - " - 2001: Informatikobjekte entstehen durch Hybridisierung. In Kurt Bauknecht et al. (Hg.): Informatik 2001. GI/OCG-Jahrestagung Wien. Österr. Comp. Ges., S. 798-803.
- [Sie02] - " - 2002: Sozialgeschichte und kulturelle Theorie der Informatik. TU Berlin, Fakultät Elektrotechnik und Informatik. Bericht 02-16.
- [Sta01] Stach, Heike 2001: Zwischen Organismus und Notation. Zur kulturellen Konstruktion des Computer-Programms. Deutscher Universitätsverlag.

Wissens-Verarbeitung in Informatik und Pädagogik

Johannes Busse, Universität Heidelberg

Informatik und Pädagogik sind dem Anschein nach sehr unterschiedliche Fächer. Auch wenn die Unterschiede überwiegen, soll im Folgenden der Blick auf einige Gemeinsamkeiten gelenkt werden.

Eine wichtige Schnittmenge zwischen Pädagogik und Informatik besteht in der medialen Konservierung von Wissen zum Zwecke seiner Vermittlung über die Grenzen des Raums und der Zeit hinweg – zwar jeweils in sehr unterschiedlicher Weise, aber doch mit einer wesentlichen Gemeinsamkeit: Beide Fächer bedienen sich der verschrifteten sprachlichen Darstellung.

Die Informatik kennt Formalismen, die zwar vom Rechner ausführbar sind, dabei aber dennoch explizit für die Mensch-Mensch-Kommunikation entworfen wurden. Dies gilt insbesondere für semiformale Spezifikationen, eine Darstellung, die explizit zwischen Experten und Anwendern Kommunikation herstellen soll.

- These: Programmiersprachen sind Sprachen in engerem Sinn: auch Menschen können damit kommunizieren, ohne dass eine Maschine je den Code ausführen müsste.

Mit Formalismen lassen sich Aspekte, wörtlich: Ansichten von Welt, z.T. durchaus adäquat beschreiben: nämlich diejenigen Teile, die den Formalismen strukturähnlich sind (oder die wir ihnen ähnlich machen wollen). Eine für Pädagogen wie Informatiker besonders wichtige Perspektive auf "Welt" ist der Umgang mit Wissen, bis hin zum derzeit so aktuellen Begriff des Wissensmanagements: Schöpfung von Wissen, seine Konservierung, seine Kommunikation, die Unterstützung seiner Aneignung.

Dafür nicht nur beschreibungsmächtige, sondern auch einfache, gering komplexe und damit hoffentlich verstehbare (semi)formale Strukturen bereit zu stellen, ist eine Kernaufgabe der Informatik. Abzuschätzen, inwiefern einzelne Beschreibungs-Frameworks einen Gegenstand *angemessen* beschreiben können, und damit Aussagen über Strukturähnlichkeiten von Formalismus und Anwendungsbereich zu treffen – dazu bedarf es freilich auf Seiten der Informatik noch einiger Lernprozesse: Es bedarf einiger erkenntnistheoretischer Offenheit und fachlicher Bescheidenheit, um mit den eigenen Tools nicht nur Welt zu verändern (und viele verschiedene lebensweltliche Erscheinungsformen möglicherweise auf das Bett des Prokrustes zu spannen), sondern auch in Kommunikation mit anderen Wissenschaftlern die Grenzbereiche (semi)formaler Beschreibungen auszuloten.

Lebenswelt aber zu beschreiben, um sie anderen Lernenden erfahrbar zu machen, sie als Kultur und Wissen zu tradieren, oder sie gar emanzipatorisch zu verändern: Dies ist eine ureigenste Aufgabe der Pädagogik. Und Gegenstandsbereiche so in einem Medium gerinnen zu lassen, dass ein Lernender in der Aneignung des Gegenstands unterstützt (aber nicht bevormundet) wird, hat sich die Didaktik vorgenommen. Pädagogik und Didaktik haben – solange sie es nicht verabsolutieren und über eine anständige Erkenntniskritik verfügen – keinen Grund, auf das kommunikative Potential zu verzichten, das (semi)formalen Darstellungen innewohnt.

Dazu bedarf es freilich auch auf Seiten der Pädagogen (der Lehrenden) und ihrer Adressaten (der Lernenden) Lernprozesse, die vor noch nicht allzulanger Zeit der Pädagogik fremd zu sein schienen: Es bedarf profunder Kenntnisse lebensweltlicher Gegenstandsbereiche *und* formaler Strukturen, um aus den (gar nicht so neuen) Potentialen der Neuen Medien Fruchtbare schöpfen zu können, ohne das Bestehende zu übersehen.

Als Anschlussfragen an diese Argumentation wären insbesondere curriculare Folgen zu diskutieren: Wie viel Pädagogik/Didaktik wäre für informatische Wissensarbeiter hilfreich, und wie viel Informatik für pädagogische Wissensarbeiter? Welches "Informatikwissen" ist damit gemeint? Und welchen Einfluss üben die beiden ungleichen Partner aufeinander aus?

Mut zum Nachdenken

Wie die Philosophie der Informatik helfen kann

Paul Drews, Universität Hamburg

Wenn ein kleines Kind in Schwierigkeiten ist, ruft es nach seiner Mutter. Ähnlich sollte sich die Informatik verhalten, wenn Sie erkannt hat, dass sie sich in Schwierigkeiten befindet. Sie sollte nach der "Mutter aller Wissenschaften" [Böhme, 1998], der Philosophie, rufen.

Die Informatik befindet sich in Schwierigkeiten, wie sich an einer Vielzahl an Quellen belegen ließe. Darauf soll jedoch verzichtet werden, da der Titel dieser Tagung bereits ausreichend auf bestehende Probleme hinweist.

Wie die Philosophie der Informatik helfen kann, wird nachfolgend an drei Themenbereichen aufgezeigt:

- Interdisziplinarität
- Ethik
- Philosophische Schulen als Orientierungsrahmen

Aus den ersten zaghaften Bemühungen, Inhalte aus anderen wissenschaftlichen Disziplinen für die Informatik zu entdecken, sind inzwischen vielfältige Publikationen und Forschungsprojekte [vgl. z.B. Valk, 2002] entstanden, die sich um eine erfolgreiche Integration von Wissen anderer Disziplinen in die Informatik verdient gemacht haben. Diese ersten Erfolge können jedoch nicht darüber hinweg täuschen, dass *Interdisziplinarität* ein schwieriges Geschäft ist [vgl. Hastedt in Scheffe et al, 1993]. Dies liegt zum einen daran, dass die Selbstauffassungen der einzelnen Wissenschaften alles andere als homogen sind und demzufolge in der Regel nur die Beziehung zu einer bestimmten Richtung einer Disziplin und nicht zu der jeweiligen Disziplin insgesamt untersucht werden kann. Zum anderen setzt es die Bereitschaft der an einem interdisziplinären Vorhaben Beteiligten voraus, sich auf die Grundannahmen, das Vokabular und andere Eigenheiten der anderen Disziplinen einzulassen.

Seit langer Zeit beschäftigt sich die Philosophie bereits mit dem Problem der Abgrenzung und der Integration der verschiedenen wissenschaftlichen Disziplinen, was nicht zuletzt aus ihrem Ursprung als "Mutter der Wissenschaften" [Böhme, 1998] resultiert. Zu untersuchen sind aus Sicht der Informatik insbesondere die Grenzgebiete zu den Disziplinen Sozialwissenschaft, Ökonomie und Arbeitswissenschaft, da diese einen bereits heute erkannten direkten Zusammenhang zur Informatik aufweisen. In Zukunft könnten jedoch auch weitere Wissenschaftsgebiete als relevant identifiziert werden.

Die Informatik sollte den Mut haben, sich von erfahrener Seite in Fragen der Interdisziplinarität beraten zu lassen.

Dass die *Ethik* bei der Informatik nicht außen vor bleiben kann, ist spätestens klar, seitdem Potentiale und Risiken der künstlichen Intelligenz diskutiert wurden. Heutzutage wird diese Diskussion um Themen wie dem Einsatz von Expertensystemen (beispielsweise in der Medizin) oder den Folgen von fehlerhafter Hard- und Software wesentlich gegenwarts- und praxisorientierter geführt als in den frühen Tagen. Unterrepräsentiert ist jedoch nach wie vor der Bereich, der sich mit den soziokulturellen Folgen der Modelle der Informatik und Kommunikationsforschung beschäftigt [Hubig, 1995]. Mit Schlagwörtern wie "Business-Process-(Re)-Engineering" werden die radikalen sozialen Dekonstruktionen in Unternehmen verniedlichend dargestellt. Gehandelt hat jedoch ein (wenn vielleicht auch inoffizieller) Vertreter der Informatik, da die IT im Zentrum des Interesses steht. Für dieses Handeln scheint die Frage "Was darf ich tun?" nicht gestellt worden zu sein.

Die Diskussion um Ethik und Informatik sollte nicht um das "ob" geführt werden, sondern sich um den Operationalisierungsgrad einer Informatik-Ethik sowie offensichtliche und verborgene Folgen des Einsatzes von IT-Systemen kümmern, ohne die "harten" Bedingungen der Praxis (insb. ökonomischer Zwänge) als Störfaktoren zu vergessen. Die Antipole "KI-Perspektive" und "humanistische Perspektive" können für einen Einstieg geeignet sein, eine spätere Betrachtung sollte jedoch wesentlich differenzierter erfolgen. Die Anforderungen in der ethischen Dimension sollten jedoch nicht nur auf die "traditionelle Ethik" beschränkt bleiben, es sollte auch die moderne angewandte Philosophie einbezogen werden, die sich als eine Art allgemeiner Lebenshilfe begreift. In den Anfängen der Philosophie im alten Griechenland war diese Art des Philosophierens selbstverständlich, heute wird sie wieder entdeckt und unter anderem in "philosophischen Praxen" praktiziert. Sie kann eine Vielzahl an Hilfestellungen anbieten, ohne dass der Nutzer tief in die Philosophie einsteigen muss.

Die Informatik sollte aus ihrem technokratischen Traum erwachen und die Augen nicht vor einer Technikfolgenabschätzung verschließen.

Der dritte Themenbereich weist einen starken Bezug zur Lehre der Informatik auf. Unter einem *Orientierungsrahmen* soll ein Bestandteil einer Theorie der Informatik verstanden werden, der metaphorisch als "Landkarte für die Informatik-Welt" beschrieben werden kann. Dieser Rahmen soll die Orientierung auf dem "offenen Boden" [Siefkes, 2002] erleichtern. Zur Benutzung der "Landkarte" ist jedoch ein Verständnis der "Legende" erforderlich. Diese kann durch die verschiedenen philosophischen Schulen repräsentiert werden, die sich immer wieder mehr oder weniger direkt in den verschiedenen Gebieten, Publikationen und Modellen der Informatik wieder finden. Dabei sollte von der wahren Tiefe einer Schule abstrahiert werden, ohne die Kernaussagen zu verwischen. Der Einstieg in die Philosophie oder in ihre verschiedenen Schulen muss keinesfalls "trocken" erfolgen, wie Rosenberg [Rosenberg, 1997] und Nagel [Nagel, 2000] zeigen.

Die einzelnen Schulen der Philosophie können in der Form von Antipolen wie z.B. den erkenntnistheoretischen Grundpositionen Realismus/Idealismus und Empirismus/Rationalismus zur Verortung von Vorgehensweisen, Modellen und Methoden genutzt werden.

Eine Einführung in den beschriebenen Orientierungsrahmen sollte möglichst zu Beginn des Studiums erfolgen. Die Bedeutung mag einigen Studenten auf den ersten Blick nicht offensichtlich sein (dies gilt jedoch sicherlich auch für andere Veranstaltungen); für das weitere Studium – und vielleicht auch das weitere Leben – kann dieser Ansatz jedoch eine technokratische Engstirnigkeit verhindern und zum Nachdenken anregen.

Literatur

[Becker et al., 1999]

Becker/König/Schütte/Wendt/Zelewski: Wirtschaftsinformatik und Wissenschaftstheorie. Wiesbaden 1999.

[Böhme, 1998]

Böhme, Gernot: Einführung in die Philosophie: Weltweisheit, Lebensform, Wissenschaft. 3. Aufl. Frankfurt 1998.

[Hubig, 1995]

Hubig, Christoph: Technik- und Wissenschaftsethik: Ein Leitfaden. 2. Aufl. Heidelberg 1995.

[Nagel, 2000]

Nagel, Th.: Was bedeutet das alles? Eine ganz kurze Einführung in die Philosophie. Stuttgart 1990.

[Rosenberg, 1997]

Rosenberg, Jay F.: Philosophieren - Ein Handbuch für Anfänger. 4. Aufl. Frankfurt a.M. 1997.

[Scheffé et al., 1993]

Scheffé, Peter et al: Informatik und Philosophie. Mannheim 1993.

[Siefkes, 2002]

Siefkes: Informatik als Hybridwissenschaft - Selbstverständnis, Ausbildung, Ethik, Theorie. Aufruf zur AG, in diesem Band.

[Valk, 2002]

Valk: Informatik als Methodendisziplin - am Beispiel interdisziplinärer Arbeit mit der Soziologie. Positionspapier, in diesem Band.

Was sind Anwendungen der Informatik?

Was könnte eine Theorie dieser Anwendungen sein?

Jochen Ludewig, Universität Stuttgart

Anwendungen der Informatik sind alle Systeme, die Informatik-Komponenten enthalten, also die Verarbeitung oder Speicherung von Informationen durch technische, traditionell nicht vorhandene Mittel leisten. (Ein Buchhandel im Web, eine moderne Kamera und ein typischer Fahrkartenautomat sind Anwendungen der Informatik, ein konventionelles Radio und ein Briefkasten sind keine, auch wenn sie mit Information umgehen.)

Von anderen Systemen der Informatik sind die Anwendungen dadurch unterschieden, dass sie nicht eine Informatik-interne Infrastruktur schaffen, wie es Betriebssysteme, Compiler, Editoren usw. tun, sondern *direkt* einem Zweck dienen (sollen), der nicht in der Informatik selbst liegt. Praktisch bedeutet das: Der Kunde ist typisch kein Informatiker.

Die eingesetzten Informatik-Komponenten bestehen aus Hard- und Software. In vielen Fällen (nämlich in massenhaft produzierten technischen Systemen) ist die Hardware sehr speziell, die Software relativ klein, so dass wir die Systeme kaum als Anwendungen der Informatik wahrnehmen. Der Airbag im Auto ist ein typisches Beispiel. In anderen Systemen ist die Hardware standardisiert, die speziellen Leistungen sind praktisch ausschließlich durch die Software definiert. Das ist typisch dort der Fall, wo auf eigentlichen Rechnern (mit Bildschirm und Tastatur) gearbeitet wird.

Nachfolgend werden nur die Systeme zweiter Art betrachtet, also Systeme, die der Benutzer als Informatik-Anwendung wahrnimmt und bei denen er (meist, wenn auch keineswegs immer) zwischen Hard- und Software unterscheiden kann. Der Vorteil dieser Einschränkung besteht einfach darin, dass "Informatik" mit "Software" gleichgesetzt werden kann. Ich spreche also von Anwendungen der Software.

Sehr viel schwieriger als zum Wort "Informatik" ist der Zugang zum Begriff der Theorie. Über diesen Begriff kann man sehr verschiedene Meinungen haben. Ich vertrete einen radikalen Standpunkt, für mich ist alles eine Theorie, was nicht unmittelbar wahrnehmbar ist, sondern aus den Wahrnehmungen als allgemein gültiges Prinzip, als Modell zur Erklärung und Prognose, destilliert wird. Dass es mal wärmer, mal kälter ist, nehmen wir wahr. Dass es einen Rhythmus der Jahreszeiten gibt, ist eine Theorie. Diese Deutung unterscheidet sich drastisch von einer Definition, die auf die Formalisierung der Modelle abhebt. Eine Differentialgleichung für natürliche Phänomene ist natürlich eine besonders schöne Theorie, weil man damit rechnen kann, aber Brook's Law ("Adding manpower to a late project makes it even later.") ist gleichwohl auch eine Theorie (mit der man leider *nicht* rechnen kann.)

Eine Theorie der Informatik-Anwendungen müsste also allgemeine Prinzipien identifizieren, die uns Erklärung und Prognose erlauben. Je nachdem, was wir erklären und vorhersagen wollen, sind sehr unterschiedliche Theorien denkbar. Ein Bereich, der mich aus berufsfachlichen Gründen interessiert, ist der Komplex Kosten und Qualität. Dazu will ich nachfolgend noch ein paar Gedanken zusammentragen.

In einer Spielzeugwelt, einfach und übersichtlich, wie sie in Lehrbüchern des Software Engineerings typisch zugrundegelegt wird, verhalten sich die handelnden Personen rational. Je nach Weltbild der Verfasser arbeiten die Leute am Wohl der Menschheit (Vortrag von Mary Shaw auf der ICSE 2000) oder am Profit ihrer Unternehmen (meine Vorlesungen über Software Engineering). In jedem Falle haben sie ein (konzeptionell) einfaches Ziel, das sie verfolgen.

Die Beobachtung in der Praxis lehrt, dass wir so nie zu einer brauchbaren Theorie kommen. Die Menschen verfolgen

- individuell sehr unterschiedliche Ziele,
- vielfach Ziele, die ihnen selbst nicht klar sind,
- vielfach mehrere Ziele, die nicht vereinbar sind,
- vielfach Ziele, die mit ihren Interessen, wie man sie aus der Distanz einschätzt, nicht vereinbar sind.
- Und natürlich sind sie weit davon entfernt zu wissen, wie sie die Ziele verfolgen sollten.

In der Entwicklung von Informatik-Anwendungen stoßen nun Ziele nicht nur verschiedener Individuen, sondern auch verschiedener Kulturen zusammen: Beteiligt sind mindestens Kunden, Anwender, Entwickler und Manager (auf Herstellerseite). Wir können leicht beobachten, dass die Resultate oft, wenn nicht in der Regel, weder eine Seite ganz befriedigen noch einen sinnvollen Kompromiss

darstellen, sondern eher wie eine Kriegsfolge aussehen. (Der Krieg ist ja auch nicht ein Kompromiss, sondern das Resultat einer voll ausgeschöpften lose-lose-Situation.)

Was man daran diskutieren, vielleicht klären könnte:

- Welche Ziele werden verfolgt? Welche Interessen haben die Beteiligten wirklich? Ist es möglich, die Ziele mit den Interessen zur Deckung zu bringen, oder bedeutet das eine Bevormundung? (So wie es eine Bevormundung ist, den Alkoholiker am Trinken zu hindern.)
- Wo liegen Kompromissmöglichkeiten? Wie kann man sie ermitteln?
- Gibt es im Zustandsraum der Informatik einen Übergang vom heute bestehenden Zustand, der oben skizziert ist, und der Utopie, die sich vielleicht konkretisieren ließe? Wäre die Utopie stabil, d.h. könnte sie sich gegen den Rückfall ins heute bestehende Chaos behaupten? Welche Leute spielen dabei eine herausragende Rolle, d.h. von wem hängt die ganze Sache ab?

Alles das (und weit mehr) würde ich gern diskutieren.

Informatik als Hybrid-Wissenschaft?

Am Beispiel der Entwicklung wissensbasierter Systeme

Frank Puppe, Universität Würzburg

Zusammenfassung: In seinem Aufruf zur AG "Informatik als Hybridwissenschaft" fordert Siefkes eine neue Standortbestimmung der Informatik. Wir untersuchen seine Thesen (Kapitel 4-5) vor dem Hintergrund der Entwicklung wissensbasierter Systeme (Kapitel 1-3), die in besonderem Maße auf die Interaktion zwischen Fachexperten und Informatikern angewiesen ist.

1. Wissensbasierte Systeme

Wissensbasierte Systeme sollen in eingegrenzten Anwendungsbereichen schwierige, d.h. wissensintensive Probleme lösen. Ihre Entwicklung erfordert die Formalisierung von Problemlösungsmethoden und Wissen. Während in der Anfangszeit der Künstlichen Intelligenz (bis ca. 1970) das Interesse fast ausschließlich allgemeinen Problemlösungsmethoden galt, hat sich seitdem die überragende Bedeutung von detailliertem Expertenwissen für die Entwicklung leistungsfähiger Programme herausgestellt. Das häufig auch als "Knowledge Acquisition Bottleneck" bezeichnete Kernproblem ist daher, wie man sehr viel Wissen in das Programm überträgt. Die wichtigsten primären Wissensquellen sind Experten, Fachbücher und Fallsammlungen. Das Wissen in den Köpfen der Experten ist jedoch schwer zugänglich, da Experten meist wenig Zeit, Motivation aber überwiegend auch große Schwierigkeiten haben, es weiterzugeben. Gute Fachbücher sind in vielen Anwendungsbereichen nicht vorhanden. Selbst wenn sie verfügbar sind, erfordert ihr Verständnis meist Hintergrundwissen und/oder Praxiserfahrung, so dass das Wissen der Fachbücher für sich allein selten ausreicht, erfolgreiche Systeme zu entwickeln. Ähnliches gilt auch für Fallsammlungen: Wenn überhaupt vorhanden, sind die Fälle meist nur unvollständig dokumentiert und enthalten häufig Fehler. Allerdings kann durch bessere Dokumentation der Wert von Fallsammlungen beträchtlich gesteigert werden. Erschwerend kommt in allen Situationen hinzu, dass das Wissen sich oft in kurzer Zeit ändert.

2. Phasen und Rollen bei der Entwicklung wissensbasierte Systeme¹

Wegen der Schwierigkeiten beim Wissenserwerb wurden viele Methoden erprobt, wobei hybride Vorgehensweisen mit wechselnden Rollen von Experten, Programmierern, unterstützenden Programmen (z.B. Wissensakquisitionswerkzeuge) und Mediatoren (Wissensingenieuren) im Vordergrund stehen. Die Phasen bei der Entwicklung von (Problemlöse-)Programmen entlang der zwei Dimensionen Abstraktionsgrad und Vagheit der Aufgabenstellung bzw. der Lösung veranschaulicht Abbildung 1.

Die sprachliche Freiheit (Vagheit) repräsentiert, wie präzise die Aufgabenstellung und das Verfahren zur Problemlösung charakterisiert ist. Der Abstraktionsgrad gibt an, ob die Beschreibung eher in Begriffen des Problemgebietes oder einer Programmiersprache erfolgt. Die Entwicklung eines Programms beginnt mit der Identifizierung eines Problems, das zunächst vage und abstrakt formuliert ist, und sollte mit einem präzisen Problemlösungsverfahren enden, das aus einem Zusammenspiel verschiedener Komponenten wie Menschen, Sensoren, Stellglieder, alte Programme, neues Programm besteht, von denen uns hier nur das neu entwickelte, auf einer Maschine ausführbare Programm interessiert.

Der empfohlene Programmierweg besteht darin, zunächst das Problem in einer problemnahen Sprache zu präzisieren bzw. zu spezifizieren und dann die präzisierte Aufgabenstellung in eine Maschinensprache umzukodieren. Ein wichtiger Fortschritt in der Informatik besteht in der Entwicklung immer problemnäherer Programmiersprachen verschiedener "Generationen" von der Maschinensprache über Assembler zu höheren Programmiersprachen, wobei die notwendigen Transformationen von einem Compiler übernommen werden. Es gibt in der Informatik vielfältige Bemühungen, noch abstraktere Programmiersprachen ("Spezifikationssprachen") zu entwickeln, in die sich die Entwicklungswerkzeuge für wissensbasierte Systeme (allgemeine Werkzeuge, die Basiswissensrepräsentationen wie Regeln unterstützen; problemspezifische Werkzeuge für Problemklassen wie Diagnostik oder Scheduling; oder bereichsspezifische Werkzeuge, die zusätzlich zu problemspezifischen Werkzeugen bereits allgemeines Basiswissen einer Branche enthalten) nahtlos einreihen lassen. Wichtig ist, dass die Spezifikationen möglichst vollautomatisch in Programm-Code

¹ Das Kapitel basiert auf dem Kapitel 1.2 in [Puppe 90].

einer Maschinensprache übersetzt werden können. Die größere Problemnähe bei den problem- und bereichsspezifischen Werkzeugen wird durch Spezialisierung erreicht und bedingt daher eine Vielfalt verschiedener Werkzeuge mit jeweils begrenztem Einsatzspektrum, was in Abb. 1 nicht veranschaulicht ist.

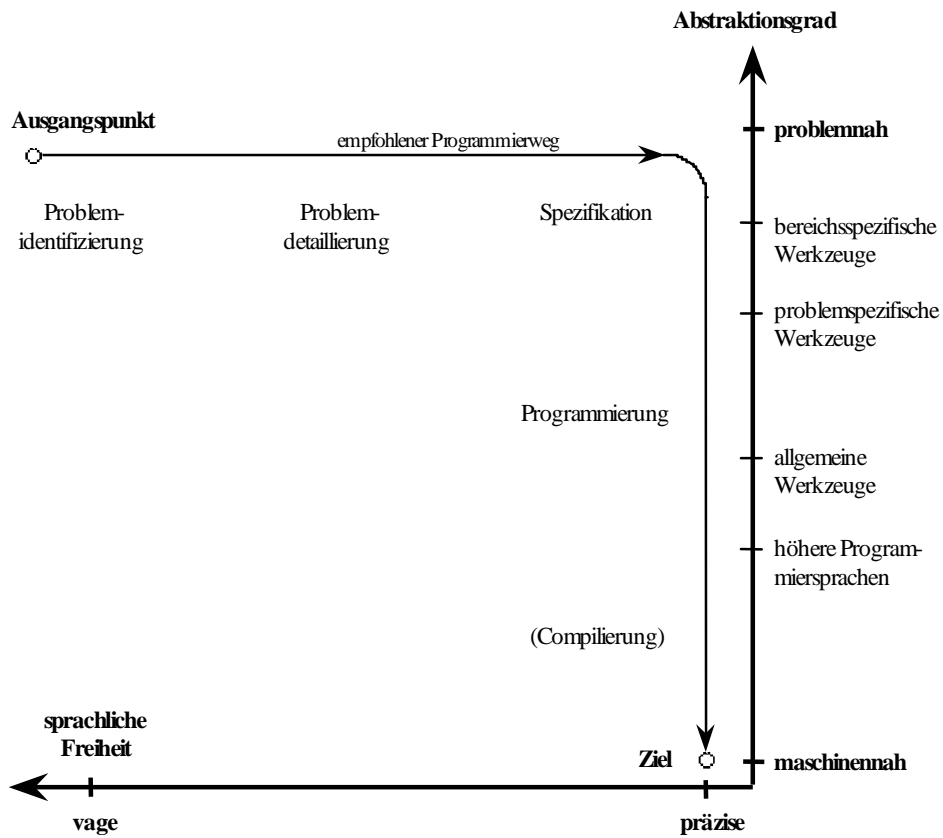
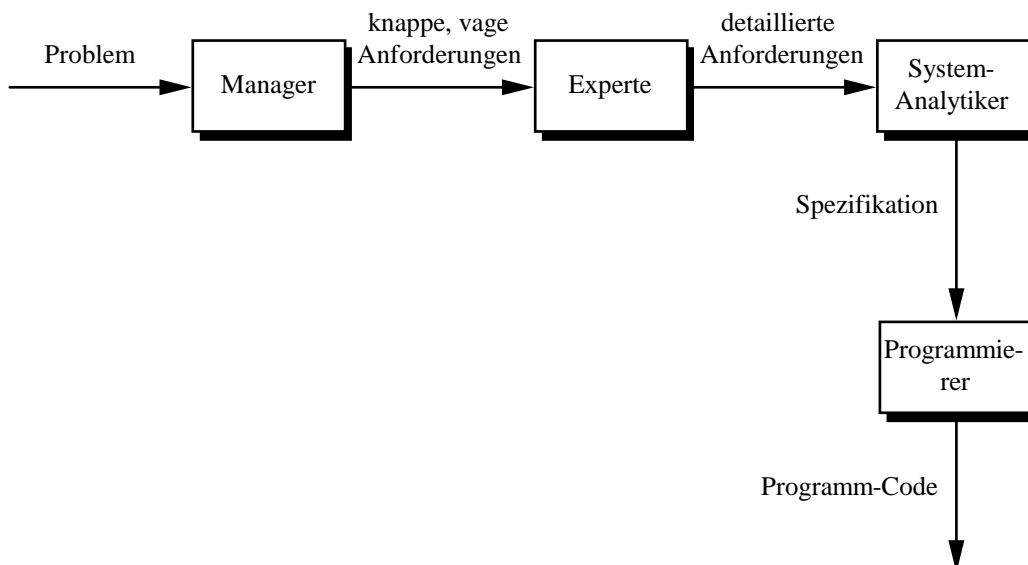


Abb. 1: Abstraktionsgrad und sprachliche Freiheit bei der Programmentwicklung

Der empfohlene Programmierweg kann nach den beteiligten Kompetenzprofilen in verschiedene Stufen unterteilt werden (Abb. 2)². Natürlich müssen nicht alle Kompetenzprofile von verschiedenen Personen ausgefüllt werden, aber die folgenden plakativen Berufsbezeichnungen erleichtern das Verständnis: Zunächst muss ein Problem identifiziert und als wichtig bewertet werden, wofür der Manager zuständig ist.



² Dieses Schema verdeutlicht mehr die logischen Stufen der Programmentwicklung als die tatsächliche Ausführung, bei der meist viele Iterationen zwischen allen Stufen unvermeidlich sind.

Abb. 2: Rollen bei der Programmentwicklung nach Abb. 1

Der Manager delegiert das Problem dann an einen Fachexperten, der für die präzise Ausformulierung des Problems und den abstrakten Entwurf eines Problemlösungsverfahrens zuständig ist. In der dritten Stufe werden die Aufgabenstellung und die Angaben über das Problemlösungsverfahren von einem Systemanalytiker oder Wissensingenieur in eine Programmspezifikation umgesetzt, die in der vierten Stufe von einem Programmierer in ein ausführbares Programm überführt wird.

3. Umsetzung der Rollen in Vorgehensweisen der Systementwicklung

In diesem Abschnitt werden die wichtigsten Vorgehensweisen zur Entwicklung wissensbasierte Systeme hinsichtlich der Qualifikationen der beteiligten Personen und der benutzten Entwicklungswerkzeuge diskutiert. Dabei wird eine Problemstellung vorausgesetzt und daher nicht auf die Sicht des Managers in Abb. 2 eingegangen. Typische Vorgehensweisen sind:

- Doppelqualifikationen: Experten, die sich sowohl im Fachgebiet als auch in der Informatik gut auskennen (häufig auch formal durch ein Doppelstudium qualifiziert), entwickeln das System weitgehend alleine.³ Der Doppelexperte übernimmt dann die Rollen Experte, Systemanalytiker und teilweise auch Programmierer in Abb. 2. Dieser Weg dominierte insbesondere in der Anfangszeit der wissensbasierten Systeme in den siebziger Jahren. So wurde z. B. das meistzitierte Expertensystem MYCIN [Shortliffe 76] von E. Shortliffe entwickelt, der sowohl eine medizinische als auch eine informatische Ausbildung hatte.
- Arbeitsteilung (indirekter Wissenserwerb): Da Doppelexperten selten sind und eine lange Ausbildung benötigen, hat sich sehr schnell eine Arbeitsteilung zwischen Fachexperten und Informatikern herausgebildet, die gemeinsam das System entwickeln und die Rolle des Systemanalytikers übernehmen. Diese Arbeitsteilung ermöglichte auch die Entwicklung wesentlich größerer Systeme. Zwei klassische Beispiele sind INTERNIST I [Miller et al. 82], das die ganze Innere Medizin umfasst (MYCIN dagegen nur die Antibiotika-Therapie bakterieller Infektionskrankheiten des Blutes und von Meningitis) und im Kern in Zusammenarbeit eines Internisten (Myers) und eines Informatikers (Pople) entwickelt wurde, sowie R1/XCON [Mc Dermott 82], das die Konfiguration der häufigsten verkauften Computertypen der Firma DEC übernahm und von zahlreichen Informatikern und Fachleuten zusammen entwickelt wurde. Dazu verständigten sich beide auf eine Schnittstelle, d.h. eine Wissensrepräsentation, gemäß der die Experten ihr Wissen formalisierten und für die die Informatiker Inferenzmaschinen entwickelten. Bei INTERNIST (Symptom-Diagnose-Tabellen) und bei R1/XCON (Regelsprache OPS5) wurden eher einfache Repräsentationen gewählt. Im Allgemeinen ist die Festlegung der Wissensrepräsentation aber ein schwieriger und dynamischer Prozess. Daher hat sich die Arbeitsteilung immer mehr dahin verschoben, dass der zum Wissensingenieur mutierte Informatiker die Wissensrepräsentation gemäß seinem Verständnis der Domäne festlegt und erprobt. Dazu liest der Wissensingenieur Fachliteratur und befragt Fachexperten. Mit dem sich angeeigneten Wissen entwickelt er dann das System selbständig, eventuell mit Rücksprache und Überprüfung durch Fachexperten. Das setzt voraus, dass der Wissensingenieur neben seinen Rollen als Systemanalytiker und Programmierer sich auch effektiv in die Domäne einarbeiten muss und entsprechende Techniken beherrscht. Dazu gehören Interviewtechniken, kontrollierte Studien, in denen Experten simulierte oder echte Fällen lösen, und psychologisch motivierte Erhebungstechniken wie das Strukturgitterverfahren. Insbesondere die Erhebungstechniken deuten darauf hin, dass sich die Zusammenarbeit immer stärker weg von einem gleichberechtigten Dialog hin zu "Tricks" entwickelte, wie man vielbeschäftigten Experten ihre verborgenen Wissensschätze im möglichst kurzer Zeit entlockt. Allerdings muss dieser Weg als gescheitert betrachtet werden, da er letztlich zu ineffektiv und fehleranfällig ist, und den Experten wird wieder eine aktivere Rolle zugeordnet. Begünstigt wird dieser Trend auch dadurch, dass es heute kaum noch anspruchsvolle Berufe gibt, in denen der Computer keine Rolle spielt, so dass die meisten Experten realistischere Vorstellungen über die Arbeitsweise von Computern und weniger Berührungspunkte als Mitte der achtziger Jahre haben. Dieser Trend zusammen mit der ständigen Weiterentwicklung von komfortablen Entwicklungswerkzeugen führte zu dem Ansatz der Selbstakquisition, bei der der Wissensingenieur in den Hintergrund tritt.
- Selbstakquisition (direkter Wissenserwerb): Der Experte gibt sein Wissen direkt in den Computer ein und wird von einem Wissensingenieur nur nach Bedarf unterstützt. Im Vergleich zur Doppelqualifikation ist der Experte aber auf dem Bereich des Systemanalytikers nur teilweise und

³ Das kann natürlich Konsultationen anderer Fachexperten oder die Beschäftigung von Programmierern zur Erledigung von Routineaufgaben einschließen, wichtig ist, dass die gesamte konzeptionelle Arbeit von einer Person ausgeht.

im Bereich des (klassischen) Programmierens kaum qualifiziert. Diese Arbeiten werden ihm von komfortablen Entwicklungswerkzeugen abgenommen, wobei er insbesondere bei deren Auswahl und Einsatz von einem Wissensingenieur beraten wird. Die Entwicklungswerkzeuge erzeugen aus dem formalisierten Wissen weitgehend automatisch Ablaufumgebungen, mit denen der Experte sein eingegebenes Wissen anhand von Fallbeispielen selbständig testen, eventuelle Fehler mit der Erklärungskomponente in der Wissensbasis lokalisieren und dann korrigieren kann. Die Selbstakquisition setzt einen erhöhten Zeitaufwand beim Experten voraus, reduziert aber die Gesamtarbeitszeit im Vergleich zum indirekten Wissenserwerb deutlich und ist damit wesentlich kostengünstiger. Außerdem identifiziert sich der Experte stärker mit dem System, so dass sich die latenten Motivationsprobleme des indirekten Wissenserwerbs und auch der Aufwand zur Qualitätskontrolle wegen Kommunikationsfehlern verringern. Allerdings ist der Experte mit der formalen Schnittstelle eines Entwicklungswerkzeuges konfrontiert, die sein Denken stark beeinflussen kann, wenn das Werkzeug sein mentales Modell prägt und in ungünstigen Fällen zu verzerrten, ineffektiven bzw. schwer wartbaren Wissensstrukturen und Vorgehensweisen führt. Die Beteiligung eines erfahrenen Wissensingenieurs reduziert diese Risiken und ist daher immer noch notwendig, auch wenn seine Rolle jetzt eine ganz andere ist als beim direkten Wissenserwerb.

- (Halb)automatische Lernverfahren (automatischer Wissenserwerb): Das Wissenssystem leitet die Wissensbasis selbstständig her. Da Lernen aus Texten (Fachliteratur) derzeit noch ein ungelöstes Problem darstellt, dienen als Basis meist Fallsammlungen. Der Prozess muss gesteuert werden, was das nur begrenztes Fachwissen erfordert. Umfangreicheres Fachwissen ist jedoch für eventuelles Hintergrundwissen und die Qualitätskontrolle nötig, da die Qualität des automatisch erworbenen Wissens meist schwer abschätzbar ist (bloße Tests mit Fallbeispielen reichen häufig nicht, da ein System auch in seltenen Fällen keine groben Fehler machen darf, aber diese wegen ihrer Seltenheit schwer gelernt und getestet werden können). Die Entlastung der Fachexperten ist trotzdem beträchtlich und betrifft insbesondere die kostspielige kontinuierliche Wartung der Wissensbasis.

4. Hybridwissenschaftliche Diskussion

Was bedeuten diese Beobachtungen für die in dem Positionspapier von Siefkes "Informatik als Hybridwissenschaft" [Siefkes 2002] angesprochenen Thesen? Den Begriff der Hybridisierung erläutert Siefkes am Beispiel der objektorientierten Programmierung: "Der Anwendungsbereich wird durch interagierende Objekte auf dem Bildschirm dargestellt, die mathematisch beschreibbar sind, aber wie reale Objekte erscheinen. Die Hybridisierung ist so weit getrieben, dass nicht nur menschliche und maschinelle Realität aus dem Blick verschwinden, sondern auch die sie verbindende der abstrakten Datenstrukturen." Wir betrachten folgende Thesen:

1. "Die Informatik ist in besonderer Weise auf interdisziplinäre Arbeit angewiesen."
2. "Das übliche Abschotten aus Arroganz oder Resignation oder Zeitmangel kann nicht einfach durch unbedenklichen Import und Export von Wissen oder durch das Einrichten von Brückenköpfen in sonst als feindlich behandelten Bereichen ersetzt werden. Die Hybridisierung gelingt nur durch gemeinsame Arbeit."
3. "Wir müssen die Ausbildung ändern. Neben Technik und Mathematik brauchen wir Kenntnisse und Fähigkeiten aus den Geistes- und Sozialwissenschaften, nicht als Schmuck, sondern als drittes Standbein. [...] Es reicht aus, wenn Studenten Programmieren als Hybridisierungsaufgabe lernen; dabei lernen sie alles, was sie wissen müssen."
4. "Um Hybridisierung zu verstehen, müssen wir nicht nur in der Lage sein, menschliche Aktivitäten in unterschiedlichen Bereichen zu formalisieren und zu maschinisieren und die automatisierten Systeme so in menschliche Umgebungen zurück zu bringen, dass sie funktionieren. Beide Unterfangen – verharmlosend De- und Rekontextualisierung genannt – verändern die Umgebungen, insbesondere die beteiligten Menschen."

Zunächst erscheinen wissensbasierte Systeme als ein prototypisches Beispiel für Hybridisierung, da hier Probleme der Realität detailliert im Computer beschrieben und manipuliert werden. Die Vermischung von Realität und Computerprogramm wird dadurch besonders deutlich, dass das Arbeitsmedium vieler Experten ohnehin der Computer ist, der zahlreiche arbeitserleichternde (oder unverzichtbare?) Dienste bereitstellt. Allerdings ist der Fokus hier weniger die Nutzung der Programme, sondern ihre Entwicklung und welche Rolle Informatiker dabei spielen. Um sich wissensbasierte Systeme besser vorstellen zu können, seien drei Problemklassen mit Anwendungsbeispielen genannt:

- Diagnoseprogramme, z.B. in der technischen oder medizinischen Diagnostik und Therapie.
- Schedulingprogramme wie das Ressourcen-Scheduling in Fertigungsunternehmen oder die Dienstplanung in Krankenhäusern.

- Simulationsprogramme z.B. bei der Konzeption von Fertigungsanlagen oder Produkten wie Autos oder bei der Durchführung von Experimenten in der Biologie.

Das zugrundeliegende Fachwissen kann im allgemeinen nicht innerhalb der Informatik vermittelt werden, da es zu speziell ist. Daher ist der ersten These von Siefkes zuzustimmen, dass derartige Projekte nur in interdisziplinärer Weise angegangen werden.

Allerdings kann man den Trend erkennen, dass die Fachexperten die zentrale Aufgabe übernehmen, die Systeme zu entwickeln und die Informatiker diesen Prozess durch Entwicklung und Bereitstellung von Werkzeugen und durch Beratung begünstigen, aber nicht wirklich gleichberechtigt sind, da sie sich durch die Aufwärtsbewegung entlang der Abstraktionsachse in Abb. 1 mehr und mehr überflüssig machen. Es erscheint wahrscheinlich, dass Fachexperten sich die notwendigen systemtechnischen Fähigkeiten zum Umgang mit Entwicklungswerkzeugen zunehmend selbst aneignen können und dies sich auch ökonomisch lohnt, so dass Informatiker mehr für die Bereitstellung der Infrastruktur zuständig sind. Dazu gehört in besonderem Masse die Interoperabilität verschiedener Teilsysteme, die vermutlich für einige Zeit im Zentrum der informatischen Forschung stehen wird (z.B. Komponentenware, Agentifizierung von Systemen, Kommunikationsstandards usw.). Daher ist fraglich, ob man Informatik als Hybridwissenschaft im Sinne der zweiten obengenannte These von Siefkes begreifen muss, sondern in diesem Zusammenhang eher als Wissenschaft von mächtigen Werkzeugen und ihrer Interoperabilität, aber weniger von deren konkreten Gebrauch.

Für die dritte These, dass die Ausbildung hin zu den Geistes- und Sozialwissenschaften geändert werden muss, spricht zunächst die erforderliche Kernfähigkeit, eng mit verschiedensten Fachexperten zusammenarbeiten zu können und diese in ihrer Arbeit und der Umgebung zu verstehen. Allerdings steht in unserer Vision nicht die gemeinsame Arbeit im Vordergrund, sondern das Ziel, die Fachexperten dazu zu befähigen, die Arbeit zunehmend selbständig übernehmen zu können (das heißt nicht, dass ein individueller Informatiker sich nicht selbst als Fachexperte begreifen kann und sich entsprechend spezialisiert, aber es trifft nicht für die Informatik als Disziplin zu). Insbesondere benötigen Informatiker weiterhin die Kernkompetenz zur Erstellung komplexer Software, da Werkzeuge wegen ihrer erforderlichen Allgemeinheit deutlich schwieriger zu erstellen sind als dedizierte Anwendungssysteme. Natürlich gibt es auch viele (Teil)Projekte, wie z.B. die Erstellung von GUIs, bei der Informatiker und Fachexperten eng zusammenarbeiten müssen, d.h. hier ist eine verzahnte Programmentwicklung nötig. Ein guter Ansatz, gerade auch für die Ausbildung von Informatikern, scheint mir hier das "Extreme Programming" [Kent 2000] zu sein, was auf informatischer Seite die Flexibilität und kontinuierliche Kommunikation stark erhöht und so Interdisziplinarität begünstigt.

Schließlich bestätigen die Erfahrungen mit wissensbasierten Systemen die vierte These, dass deren Einsatz die Umgebung verändert. Die Berücksichtigung dieses Umstandes bzw. die realistische Einschätzung eventueller Widerstände sind sogar einer der zentralen Erfolgsfaktoren wissensbasierter Systeme. Allerdings ist deren Einschätzung und die Reaktion darauf eine typische Management-Aufgabe. Daher wäre es unrealistisch, das dafür jeder Mitarbeiter eines Projektes die Verantwortung übernehmen muss. Obwohl im Zuge der steigenden Verantwortung vieler Mitarbeiter in Firmen der Trend besteht, dass diese auch mit Managementaufgaben konfrontiert sind, glaube ich nicht, dass dies ein spezifisch informatisches Problem ist.

5. Zusammenfassung

Informatiker neigen aufgrund der massiven Auswirkungen, die informationsverarbeitende Systeme in unserer Gesellschaft haben, sich für viele der damit verbundenen Probleme zuständig zu erklären. Das unterschätzt jedoch die Rolle der "Anwender" dieser Programme, für die der Computer ein immer selbstverständlicheres Arbeitsinstrument wird. Daher stimme ich zwar der Beobachtung zu, dass Informatik interdisziplinär angelegt ist und auch als Hybridwissenschaft aufgefasst werden kann, nicht jedoch den Schlussfolgerungen aus [Siefkes 2002] bezüglich Änderung der Ausbildung, der Art der Arbeitsteilung und der Berücksichtigung der Folgen des Einsatzes von Programmen. Anstatt überall mitzumischen, könnten Informatiker sich auch auf ihre genuinen Stärken bei der Softwarekonzeption und Entwicklung konzentrieren und diese auszubauen: Sie überlassen den Fachexperten die führende Rolle bei der Erstellung und dem Einsatz von Software in ihren jeweiligen Domänen und unterstützen sie dadurch, dass sie flexible, wiederverwendbare und adaptierbare Entwicklungswerkzeuge und Kooperationsformen anbieten. Interdisziplinarität und Arbeitsteilung müssen kein Widerspruch sein. Da die Fachexperten solide informatische Grundkenntnisse benötigen, sollte die Hybridisierungsdiskussion der Ausbildung in deren entsprechenden Fach- und insbesondere die zunehmenden Kombinationsstudiengänge hineingetragen werden.

Literatur

Kent, K: Extreme Programming Explained. Addison-Wesley, 2000.

Mc Dermott, J.: A Rule Based Configurer of Computer Systems. Artificial Intelligence 19, 39-88, 1982.

Miller, R., Pople, H., and Myers, J.: INTERNIST I, an Experimental Computer-based Diagnostic Consultant for General Internal Medicine. New England Journal of Medicine 307, 468-476, 1982.

Puppe, F.: Problemlösungsmethoden in Expertensystemen. Springer, 1990.

Shortliffe, E.: Computer-based Medical Consultations: MYCIN. American Elsevier, 1976.

Siefkes, D.: Informatik als Hybridwissenschaft. Aufruf zur AG, in diesem Band.

Ökumenische Studien der Informatik

Jochen Schneider, Universität Magdeburg

Die Vermittlung "weicher" Fertigkeiten im Informatikstudium wird seit längerem nicht nur von Vertretern des Faches, sondern auch solchen der Industrie gefordert. Es hat sich jedoch als schwierig erwiesen, geeignete Inhalte und Formen zu finden. Siefkes weist im Aufruf zur Arbeitsgruppe darauf hin, dass Geschichte einen entsprechenden Zugang zur Informatik bieten kann. Er bezieht sich dabei auf frühe Entwicklungen von Technik und Disziplin. Gleichzeitig nennt er als Fazit eines Arbeitskreises zur Informatikgeschichte in der Informatiklehre, dass Studierende zu geschichtlichen Betrachtungen nur selten zu motivieren sind (Siefkes 2001).

Pflüger (2001) und Coy (2001) betonen die Wichtigkeit der Aufnahme aktueller Entwicklungen in die Lehre von Informatik und Gesellschaft. Diese ergibt sich aus der Verbreitung entsprechender Inhalte in fachlichen und populärwissenschaftlichen Medien in gedruckter und Onlineform (Pflüger 2001). Weiterhin stellt die rasante Verbreitung Digitaler Medien in und durch aktuelle Publikationen eine Möglichkeit dar, die "unmittelbare Kopplung von Informatik und Gesellschaft, Informatik und Kultur" (Coy 2001) zu studieren.

Es geht also um mehr als nur technische Untersuchungen. Wie sollten sie praktisch durchgeführt werden? Unter vielen Informatikern steht die fachliche Auseinandersetzung auf anderer als technischer Ebene nicht hoch im Kurs. Es haben sich daher keine mit künstlerischen Disziplinen vergleichbare Schulen als Richtungen herausgebildet (Rolf 1992). Andererseits wird innerhalb und außerhalb der Informatik die Anhänglichkeit zu gewissen von ihr hervorgebrachten Artefakten mit Religiosität verglichen, von Eco das Verhältnis zwischen Anhängern verschiedener Betriebssysteme überspitzt als "Ein neuer Heiliger Krieg" bezeichnet.¹

Auch wenn die Entwicklung der Informatik häufig als eindimensional beschrieben wird, so zeigen sozialgeschichtliche Untersuchungen, dass es immer Entscheidungsmöglichkeiten der beteiligten Akteure gab. Und auch wenn sich, vermutlich durch die Orientierung der Informatikentwicklung an Bedürfnissen des Marktes und der Militärs, immer vorherrschende Ansätze ausgeprägt haben (wie der sogenannte objektorientierte Ansatz zur Zeit), so haben sich doch immer Alternativansätze gehalten, die auch in andere Kontexte als die ursprünglichen übertragen wurden (beispielsweise der Einsatz formaler Methoden für die Spezifikation von Benutzungsschnittstellen oder der Einsatz einer interaktiven Programmiersprache wie Smalltalk für die Echtzeitverarbeitung).

Pflüger bringt neben der Beurteilung historischer Prozesse Gedichtinterpretationen als Lehrformen für das Informatikstudium ins Gespräch, damit Studierende eine Sprache lernen, um Differenzen, Konflikte und Irrtümer ausdrücken zu können, was für ihre Berufspraxis wichtig ist (1994). Inhaltlich näher am Fach selbst wäre es, unterschiedliche Methoden (z.B. eher formale gegenüber eher sozialwissenschaftlichen) und Werkzeuge (Betriebssysteme, Programmiersprachen und Entwicklungsumgebungen) der Informatikpraxis gegenüberstellend und streitend zu vergleichen. Derzeit werden Mittel und Produkte der Informatik häufig in unterschiedlichen Vorlesungen oder dem zeitlich aufeinanderfolgenden Beschreiben von z.B. Programmiersprachen in einer einzigen Lehrveranstaltung behandelt. Je nach Wissensstand und Vorlieben werden dadurch Vorurteile eher zementiert denn aufgeweicht.

Durch den Wunsch nach sanften Übergängen zwischen Technologien und der technischen Möglichkeit der weitgehenden Anpassbarkeit vor allem von Software konnten historische und aktuelle Alternativen schon immer nebeneinander in verschiedenen Rechnersystemen oder sogar auf demselben Rechner existieren. Interessant für die Informatikausbildung könnte es daher sein, das Kontrastieren eigener Bedürfnisse und Meinungen schon im kleinen zu üben.

Motivierende zeitgeschichtliche Untersuchungen und streitendes Lernen können zu konfrontativem Vergleich von mehr oder weniger bekannten Methoden und Werkzeugen durch jeweilige Fürsprecher kombiniert werden. Coy benutzt diese Form für die Lehre des Faches "Informatik und Gesellschaft" und preist sie als "gutes Training zum Präsentieren, Zuhören und Urteilen" (2001). Er vergleicht sie mit gerichtlichen Verhandlungen.

¹ „The characteristics that distinguishes holy wars from normal technical disputes is that in a holy wars [sic.] most of the participants spend their time trying to pass off personal value choices and cultural attachments as objective technical evaluations“ (Raymond 1994, 230).

Valk hat in einem Artikel die Informatik mit Jura verglichen (1997). Beide vermitteln zwischen Formalem und Sozialem (bei Jura: Gesetz und Belange des Einzelnen). Er vermutet, dass in der Informatik häufig ein Vergleich zu erwarten ist, im Gegensatz zum abschließenden Urteil eines Rechtsprozesses durch einen Richter. Dies würde auch für den Streit zwischen Studierenden in den Rollen von Vertretern bestimmter Methoden oder Werkzeuge gelten: Nur in einem jeweils sehr genau festgelegten Kontext kann ein bestimmter Ansatz als überlegen angesehen werden.

Die zu vergleichenden Produkte der Informatik könnten dabei nach Interesse der Studierenden ausgewählt werden. Wenn die Produkte weder zu alt noch zu jung sind, könnten Studierende eigene und fremde Erfahrungen leicht zusammenbringen. Vorangehen müsste einem solchen Streitgespräch die Aneignung des jeweiligen Herkunftsgebietes und des Entstehungskontextes. Ergebnis könnte ein gemeinsames Dokument sein, in dem Vor- und Nachteile der jeweiligen Ansätze unter Berücksichtigung ihrer ursprünglichen und anderer Kontexte abgewogen werden.

Ziel soll dabei aber gerade nicht das Aufzeigen von Beliebigkeit sein, sondern das Fördern des Verständnisses für die Geschichtlichkeit von Entwicklungen. Ziel soll es weiterhin sein, den Diskurs nicht nur mit technischen, sondern auch mit konzeptionellen und sozialen Argumenten zu führen. Diese Argumente könnten sich dadurch als gerade nicht "unsachlich" erweisen. Dadurch könnte Verständnis für Formen geistes-, sozial- und humanwissenschaftlicher Diskurse entstehen.

Literatur

Coy, W. 2001: Weder vollständig noch widerspruchsfrei. *FIfF-Kommunikation*, 4/01, S. 45-48.

Pflüger, J. 1994: Auf der Mauer. *Informatik-Spektrum* 17, 4/01, S. 251-257.

Pflüger, J. 2001: Was machen wir, wenn wir gewonnen haben sollten? *FIfF-Kommunikation*, 4/01, S. 16-18.

Raymond, E.S. 1994: *The New Hacker's Dictionary*. 2. Aufl. MIT Press.

Rolf, A. 1992. Sichtwechsel: Informatik als Gestaltungswissenschaft. In: W. Coy et al. (Hg.), *Sichtweisen der Informatik*. Vieweg. S. 33-47.

Siefkes, D. 2001: Schreiben und Geschichte als Zugang zur Informatik. *FIfF-Kommunikation*, 4/01, S. 11-13.

Valk, R. 1997: Die Informatik zwischen Formal- und Humanwissenschaft. *Informatik-Spektrum* 20, 2. S. 95-100.

Informatik als Hybridwissenschaft.

Selbstverständnis, Ausbildung, Ethik, Theorie

Vorstellung der Arbeitsgruppe

Dirk Siefkes, Technische Universität Berlin

Im Aufruf zur AG habe ich die Informatik als eine Wissenschaft charakterisiert, die die gegensätzlichen Welten von Mensch und Maschine mit Hilfe von Formalismen *hybridisiert*, also zu etwas Neuem verschmilzt. Mensch und Maschine als Gegensätze hinzustellen, ist altmodisch: Der Computer ist zum Medium geworden, die Zeiten des Kampfes mit der Maschine sind vorbei. Eben deswegen wollen wir in der AG das Verhältnis von Mensch und Maschine näher ansehen. Vielleicht gewinnen wir damit einen Zugang zu dieser Wissenschaft Informatik, um deren Selbstverständnis es hier geht. Wie könnten wir vorgehen? In meinen Vorschlag beziehe ich Anregungen aus den Positionspapieren ein.

Wir könnten mit der Beziehung zwischen Mensch und Maschine anfangen: Wie gehen Entwickler und Benutzer mit Computern um? Welche Probleme haben sie dabei? Welche Annehmlichkeiten? Wie wirkt sich beides auf ihre Arbeit aus? Auf sie selber?

Dann können wir den Gegensatz herausarbeiten: Computer können vieles gut, was Menschen schlecht oder gar nicht können, z.B. Algorithmen ausführen. Umgekehrt können Menschen vieles gut, was Computer nicht können, z.B. Algorithmen kritisieren. Gibt es überhaupt etwas, was beide können?

Auf der Basis können wir die Beziehung betrachten: Wie gehen Informatiker mit der Gegensätzlichkeit um? Der erwähnte Befund lautet: Sie vereinbaren die unvereinbaren Welten mit Formalismen – "Sprachen", die sie auf zwei Weisen lesen müssen. Programme haben zwei Gesichter, sie sind janusköpfig: Menschen erscheinen sie als Texte, die sie lesen und verstehen können. Für Computer sind es Symbolfolgen, die ausgeführt werden müssen; so müssen die Menschen sie also auch lesen können.

Damit ist der Dualismus verschwunden, er ist zur Dreierbeziehung geworden. Also können wir die untersuchen, z.B. historisch: Wie und wozu nutzen Informatiker diese Methode der Hybridisierung? Welchen Teilen des Dreiecks wenden sie ihre Aufmerksamkeit zu? Was unterstützen sie?

Die unterschiedlichen Formen der Dreierbeziehung, die dabei zu Tage treten, können wir vergleichen: Welche Sichtweisen auf die Informatik oder aus ihr heraus bringen sie hervor? Welchen Abgrenzungen und Strukturierungen der Disziplin entsprechen sie? Vertreten die Fachgebiete bestimmte Sichtweisen?

Nun können wir die Sichtweisen bewerten: Welche Folgen haben sie für unsere Arbeit, für deren Ergebnisse, für deren Nutzer? Welche davon halten wir für wünschenswert? Mit welchen Sichtweisen können wir uns identifizieren? Was wollen wir mit der Informatik?

Das bringt uns auf "Informatik und Gesellschaft" als Fachgebiet: Sollte es sich nicht mit solchen Fragen befassen, z.B. mit Mensch-Maschine-Kommunikation, Humanisierung vs. Automatisierung der Arbeit, Verantwortung (Gesetze vs. Leitlinien vs. individuelle Moral)? Teilweise geschieht das, aber meist auf der phänomenologischen Ebene. Die spezifisch informatischen Hintergründe kommen selten in den Blick. Und wenn sich Vertreter auf eine kritische Analyse informatischen Vorgehens einlassen, werden sie leicht selbst zu "normalen" Informatikern. Wissenschaftliche Arbeit "auf der Mauer" (Jörg Pflüger) zwischen den Disziplinen hockend, janusköpfig, scheint schwierig.

Insbesondere wird in "Informatik und Gesellschaft" die formale Seite der Informatik nicht untersucht. Formale Ausdrücke sind mathematisch gesehen abstrakte Beschreibungen, die für den Computer in konkret ausführbare Anweisungen umgeformt werden müssen. Wir Menschen lesen sie aber als Organisationsformen, vielleicht juristisch oder ethisch untermauert, die immer interpretiert werden müssen, also Spielräume für Handlungen liefern, aber keine Handlungen selbst.

Als eine Folge wird "Technik" gern unbesehen auf den Menschen übertragen, von Geistes- und Sozialwissenschaftlern ebenso wie von Informatikern. "Techniken" des Handelns und Denkens werden mit Maschinentechniken gleichgesetzt und lassen sich dann problemlos durch diese ersetzen. Technische Agenten im Netz lösen Probleme menschlicher Akteure; was sind die Folgen solcher Verbindung?

Das bringt uns zurück zur Informatik: Wie gehen Informatiker mit solchen Fragen um, z.B. in Softwaretechnik oder bei Datenbanken? Sind sie sich der Hybridisierung, die sie betreiben, bewußt?

Gehen sie sie als Herausforderung an? Oder spiegeln die Probleme des Fachs "Informatik und Gesellschaft" die der Disziplin Informatik? Ist das eine so theorielos wie die andere?

Informatik als Methodendisziplin - am Beispiel interdisziplinärer Arbeit mit der Soziologie

Rüdiger Valk, Universität Hamburg

"Eine soziale Umgebung ist kein Text, den man mit 'cut and paste' editieren kann. Auch die Theorien der Nachbardisziplinen genügen nicht für eine theoretische Grundlegung der Informatik; wir müssen auf beiden Seiten des Grabens stehen. Eine Theorie der Informatik muss die Disziplin mit anderen verknüpfen und so Informatik als Hybridwissenschaft erweisen."

Diese schließenden Sätze im Einladungsbeitrag von D. Siefkes' [Siefkes 02] laden dazu ein, aus einem DFG-Projekt zur Sozionik¹ zu berichten. Ziel des Projektes war, (ich zitiere aus dem Antrag) "typische Ineffizienzen öffentlicher Verwaltungen aufgrund problematischer Entscheidungsstrukturen sozionisch, d.h. hier auf Basis soziologischer Organisationstheorien und softwaretechnisch sowie formal fundierter Petrinetzagentensysteme, zu analysieren, Entscheidungsversagen aufzudecken und auf der Basis von soziologisch begründeten Handlungsanweisungen in Petrinetzagentenmodellen Alternativen zu entwerfen oder durch Entwicklung hybrider Gemeinschaften Effizienzsteigerungen vorzubereiten".

Petrinetze wurden von Beginn an dazu konzipiert, Handlungen und Handlungskomplexe sowie Beziehungen von Sachverhalten zu modellieren. C.A. Petri wollte dies nie ausschließlich nur auf die Informatik bezogen wissen, wenngleich die weitgehendsten Modellierungserfahrungen und Theoriefortschritte in der Informatik gemacht wurden. Es war daher aussichtsreich, dies in den Dienst soziologischer Theoriebildung zu stellen, zumindest dort wo der Begriff des sozialen Handelns im Vordergrund stand. Konkret konnten Szenarien² aus dem Bereich der soziologischen Organisationstheorien erfolgreich mit höheren Petrinetzen (d.h. Objektnetzen [Valk 98] und insbesondere Referenznetzen³ [Kummer 98,02]) modelliert werden.

Dabei erwies es sich als notwendig, einschlägige Theorien der Soziologie heranzuziehen und insbesondere auf ihre operationalisierbaren Inhalte zu prüfen. Behandelt wurden die Theorien von Crozier/Friedberg, Bourdieu, Elias, Popitz und Schimank [HKLMR00,01, KLMR00, Langer03, KLR03]. Auch hier – und dies ist in Hinblick auf das Thema dieses Beitrages von Interesse – erwies sich die Petrinetz-basierte Modellierungstechnik als praktikabel und als Wegbereiter neuer Erkenntnisse. In dem Projekt saßen Informatiker und Soziologen beisammen, gingen miteinander die jeweilige Theorie Schritt für Schritt durch und stellten die Quintessenz des gerade betrachteten Aspektes durch ein Petrinetzkonstrukt dar. Dazu war natürlich jeweils ein mehr oder weniger großer Abstraktionsschritt notwendig, da eine solche Operationalisierung nicht unbedingt schon vorgegeben war. Zuweilen gab es lange Diskussionen über die Adäquatheit eines Schrittes. Dies erinnert stark an ein Auftraggeber/Modellierer-Gespräch bei der Anwendungsprogrammierung. Das Besondere ist jedoch, dass das Ziel nicht ein Stück Software sondern ein neuer Theorieteil (oder zumindest die Interpretation eines solchen) war. Beide Partner zogen ihren Vorteil aus diesem Vorgehen. Die Informatiker verstanden besser, was die jeweiligen soziologischen Theorien aussagen (zur Vorbereitung einer realen Ablaufmodellierung), während die Soziologen neue Struktureinsichten gewannen. Teilweise waren sie gezwungen, über bestimmte Aspekte nachzudenken, die vorher übersehen wurden oder die in der Theorie nicht explizit ausgeführt wurden. Ein intuitives Verstehen oder eine schließende Deutung ist jedoch für die Petrinetzmodellierung nicht ausreichend. Soziologische Theorien spezialisieren sich meist auf bestimmte Aspekte. Dadurch werden andere Bereiche nicht oder nur kursorisch behandelt. Durch die strukturellen Vergleiche konnte nun damit begonnen werden, solche "Lücken" durch Bezug auf andere Theorien zu schließen. Umgekehrt lernten die Informatiker in soziologischen Kategorien zu denken und zu modellieren. Dies führte zu einer Modellierung von Petrinetzagenten, die entsprechend der allgemeinen soziologischen Theorie ein spezifisches und individuelles Verhalten haben. Ihre Struktur entspricht der von BDI-Agenten der verteilten Künstlichen Intelligenz in der Darstellung durch Petrinetze.

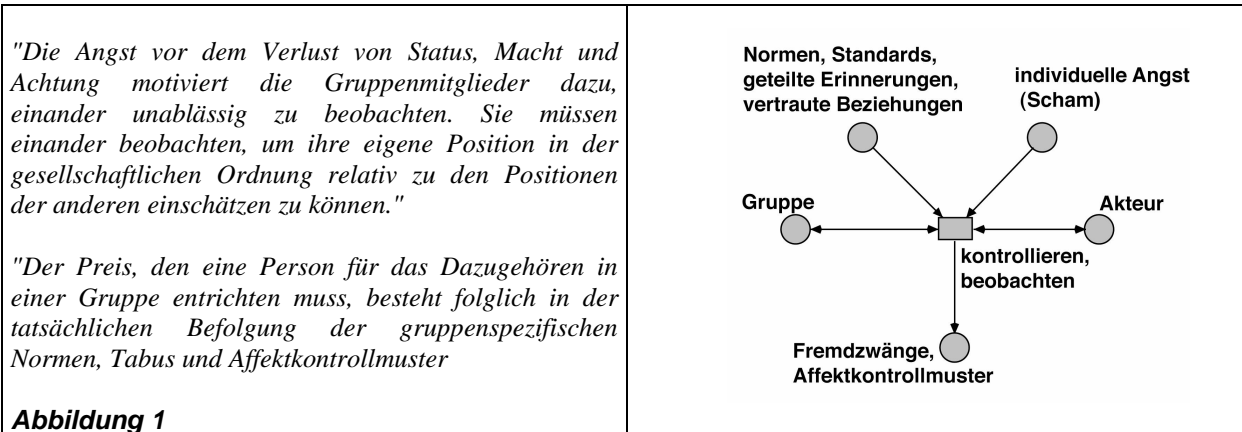
Da dieser Vorgang ziemlich ungewohnt und innovativ ist, möchte ich ihn an einem kleinen Beispiel

¹ ASKO: Agieren in sozialen Kontexten - ein sozionischer Ansatz zur Modellerstellung und Theoriebewertung.
<http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/>

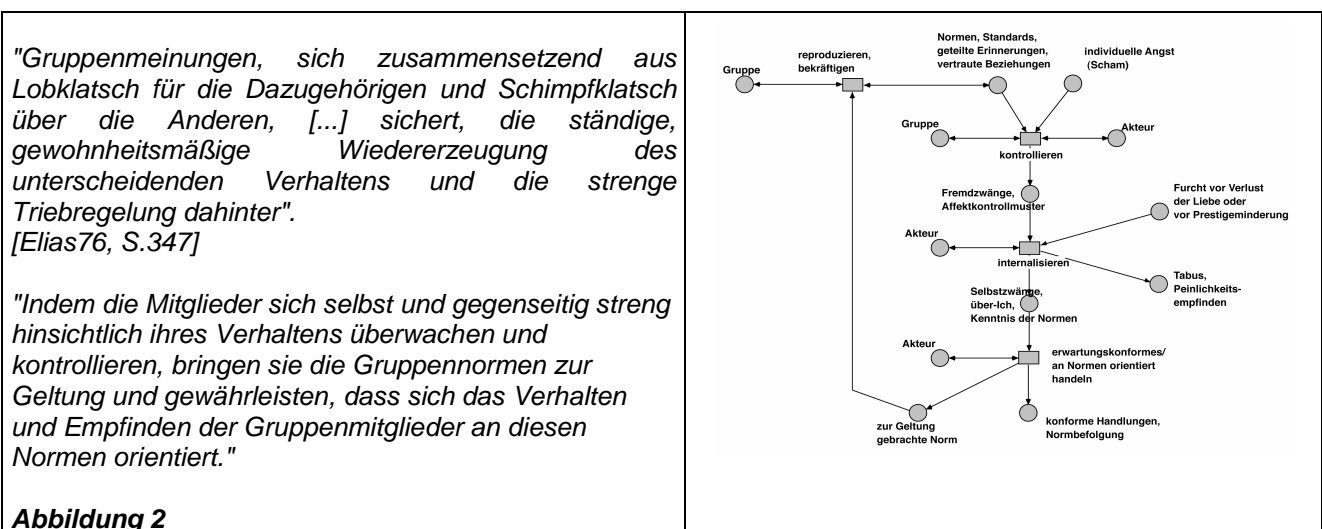
² insbesondere das „garbage can“-Szenario [HHM 00]

³ für Referenznetze gibt es das Werkzeug „renew“ unter <http://www.renew.de>

illustrieren. Der erste Kasten (Abb.1) enthält links einen Text oder die Paraphrase einer Aussage von Elias [Elias 76] und rechts daneben, eine entsprechende Petrinetz-"Transition". Die runden Symbole ("Stellen") beschreiben Teilaussagen bzw. Sachverhalte, während das Rechteck ("Transition") eine Verknüpfung darstellt, die hier in Handlungen bestehen kann. Ein Pfeil von einer Stelle zu der Transition kennzeichnet die der Stelle zugeordnete Teilaussage als Vorbedingung oder Ausgangspunkt, während in einer Stelle mündende Pfeile Nachbedingungen oder Folgen beschreiben. Doppelpfeile führen zu Nebenbedingungen, d.h. Sachverhalten, die für den Übergang heranzuziehen sind. Im Beispiel von Abbildung 1 wird also durch eine Transition ein Vorgang dargestellt, dessen Ausgangspunkt individuelle Angst und kollektive Normen sind. Das Ergebnis sind Fremdzwänge und Affektkontrollmuster. Beteiligt an dem Vorgang ist natürlich das Individuum (der Akteur) und seine Wahrnehmung der Gruppe, beides als "Nebenbedingung" darstellt.



Der zweite Kasten in Abb. 2 zeigt die Kombination mehrerer so gebildeter Transitionen, wobei ein Rückbezug zu beachten ist. Er rührt von einer Rekurrenz ("Wiedererzeugung") in der Theorie von Elias. Der soziologisch beschriebene Reproduktionskreislauf der Internalisierung von Fremdzwängen spiegelt sich in der zyklischen Struktur des Petrinetzgraphen. Der dritte Kasten (Abb. 3) zeigt schließlich das Ergebnis der Zusammenfügung mehrerer solcher Teilergebnisse. Damit soll die Komplexität der durchgeführten Modellierung illustriert werden. Der grau unterlegte Teil ist das in Abb. 2 dargestellte Teilergebnis. Eine genauere Beschreibung ist in [HKLMR 01] nachzulesen.



Obwohl Petrinetze konzeptionell einfach sind und keine speziellen Kenntnisse (wie zum Beispiel Programmierung) erfordern, war zu beobachten, dass ihre Modellierung immer die Domäne der Informatiker blieb. Die Soziologen machten große Fortschritte im Verständnis der Modelle, so dass es nach einiger Zeit möglich war, qualitativ anspruchsvoll mit ihnen über deren Adäquatheit in Hinblick auf die originären soziologischen Theorien zu diskutieren und sogar zu streiten. Offenbar erfordert der Entwurf der Modelle eine langjährige Erfahrung, wie sie zum Beispiel normalerweise Informatiker erwerben. Das Beispiel zeigt weiter, wie die Informatik als Methodendisziplin in anderen Bereichen in

einer Weise wirksam werden kann, die weit über das gängige softwaremäßige Umsetzen von gegebenen Spezifikationen hinausgeht. Ich fühle mich dabei in meinen früher geäußerten Thesen [Valk 97,01] bestätigt, die die eigenständige Kompetenz des Informatikers gegenüber einer "anwendenden" Disziplin hervorheben. Diese Kompetenz ist dabei in gewissen Grenzen wertneutral. Ähnlich wie ein Jurist einen Fall auf der Basis von formalen Gesetzestexten behandelt und dabei von anderen Gesichtspunkten weitgehend abstrahiert. Damit die Kommunikation mit der anderen Disziplin auf anspruchsvollem Niveau zu führen ist, muss der Informatiker andererseits auch hinreichend "auf der anderen Seite des Grabens" agieren können, wie D. Siefkes in dem Eingangszitat formuliert. Eine "Theorie der Informatik" sollte auf Methodenkompetenz abheben, da sie ja im Verhältnis zu vielen anderen Partnerdisziplinen wirksam werden soll.

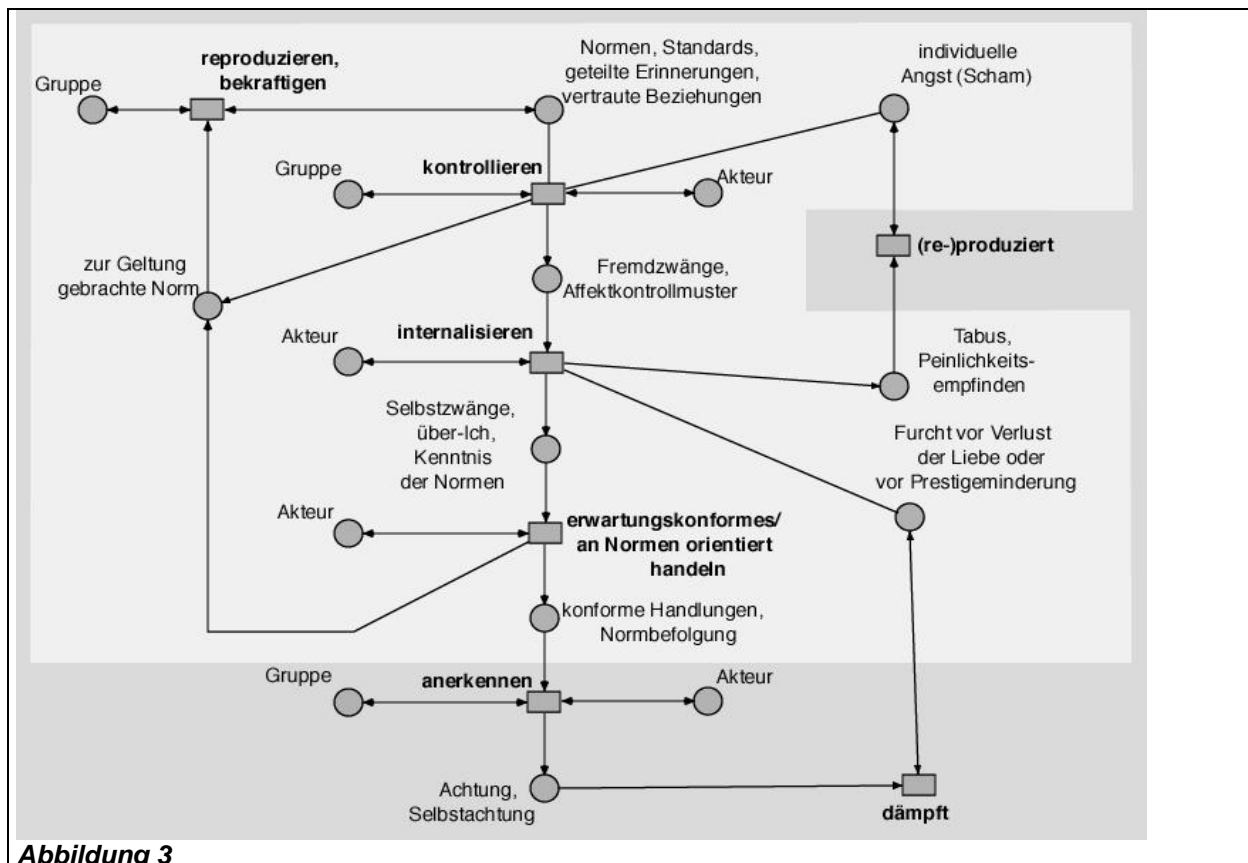


Abbildung 3

Literatur

- [Elias 76] Elias N.: Der Prozess der Zivilisation. Suhrkamp, Frankfurt 1976 (1939)
- [HHM 00] Heitsch S., Hinck D., and Martens M.: A New Look into Garbage Cans - Petri Nets and Organisational Choice. In: Proceedings of AISB 2000. Time for AI and Society. April 17-20, Birmingham, UK, 2000.
- [HKLMR 00] Hinck D., Köhler K., Langer R., Moldt D., Rölke H.: Bourdieus Habitus-Konzept als prägendes Strukturelement für Multiagentensysteme. Mitteilung 298/00 Univ. Hamburg, Fachbereich Informatik, 2000.
- [HKLMR 01] Hinck, D., Köhler M., Langer R., Moldt D., Rölke H.: Organisation etablierter Machtzentren: Modellierungen und Reanalysen zu Norbert Elias. Mitteilung 306/01, Univ. Hamburg, Fachbereich Informatik, 2001.
- [KLR 03] Köhler M., Langer R., Rölke H.: Sozionisches Modellieren mit Petrinetzen - ein kommentiertes Protokoll. In: [LMV 03].
- [KLMR 00] Köhler M., Langer R., Moldt D., Rölke H.: Combining the Sociological Theory of Bourdieu with Multi Agent Systems. In C. Jonker, A. Letia, G. Lindemann, T. Uthmann (eds): Modelling Artificial Societies and Hybrid Organisations. MASHO'00, Workshop at the ECAI 2000.
- [Kummer 98] Kummer O.: Introduction to Petri nets and reference nets. Sozionik-aktuell 1(2001). <http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/journal/1/index.html>
- [Kummer 02] Kummer O.: Referenznetze, Dissertation, Fachbereich Informatik, Univ. Hamburg, 2002.

- [Langer 03] Langer R.: Heinrich Popitz' Anatomie der Selbstorganisationsprozesse sozialer Machtordnungen - ein methodologischer Kommentar zur soziologischen Theorierekonstruktion. In: [LMV 03].
- [LMV 03] Lüde R. v., Moldt D., Valk R.: Sozionik - Modellierung soziologischer Theorie. LIT Verlag, Münster, 2003.
- [Siefkes 02] Siefkes D.: Informatik als Hybridwissenschaft - Selbstverständnis, Theorie, Ausbildung, Ethik. Aufruf zur Arbeitsgruppe, in diesem Band.
- [Valk 97] Valk R.: Die Informatik zwischen Formal- und Humanwissenschaften. Informatik-Spektrum 2(1997), S. 95-100.
- [Valk 98] Valk R.: Petri nets as token objects: An Introduction to elementary object nets. In J. Desel and M. Silva (eds), Application and Theory of Petri Nets. Lecture Notes in Computer Science, Vol. 32, Springer, Berlin, 1998.
- [Valk 01] Valk R.: Von der Mikro- zur Makrotriade. Informatik im Zwiespalt der Wahrnehmungen. Positionspapier in Nake/Rolf/Siefkes: Informatik - Aufregung zu einer Disziplin, Arbeitstagung Heppenheim 2001. Bericht 235/01, Univ. Hamburg, Fachbereich Informatik, 2001, S. 73-75.

Informatik ist Kulturwissenschaft! (ja und?)

Ulrike Wilkens, Hochschule Bremen

"Im Umgang mit dem Computer *hybridisieren* wir die gegensätzlichen Welten von Menschen und Maschinen mit Hilfe formaler Ausdrücke. (...) Wie im Umgang mit dem Computer gegensätzliche Welten, müssen in der wissenschaftlichen Arbeit unterschiedliche Disziplinen hybridisiert werden, insbesondere geistes- und sozialwissenschaftliche mit ingenieur- und naturwissenschaftlichen." So umreißt Dirk Siefkes in seinem Aufruf zur AG den Problembereich, der im Zentrum der Diskussion stehen soll.

Für mich stellt sich seit zwei Jahren die Frage nach dem Verhältnis von Informatik und "anderen Wissenschaften" im pädagogischen Alltag – im Weiterbildungsgeschäft, in der Ausbildung an der Hochschule, in der bildungspolitischen Entscheidungspraxis,¹ wo ich Aussagen zum Lernen und Lehren mit dem Computer machen muss, die sich in Empfehlungen oder Handlungsanweisungen in der Aus- und Weiterbildung niederschlagen sollen. Von einer theoretischen Position, die Informatik als technische Semiotik und damit im weitesten Sinne als Kulturwissenschaft ausweist,² blicke ich also auf das, wofür man mir als Informatikerin Kompetenz zuschreibt – auf das Lernen und Lehren mit dem und über den Computer, und frage mich:

Was (oder wem) nützt eine Theorie, die Informatik als Kulturwissenschaft ausweist?

Wie? Was?

Komische Frage, falls sie von einer Informatikerin gestellt wird, *die Wissenschaft betreibt*.

Die mögliche (und ernst gemeinte) fragmentarische Antwort: "...Erkenntnisgewinn..." könnte als dezenter Hinweis auf die Unangemessenheit der Frage sein, weil sie fälschlicherweise impliziert, dass es angebracht sein könnte, in irgendeiner Form den "Nutzwert" einer Theorie bestimmen zu wollen.

Ja genau! Sag doch mal!

Gute Frage, wenn sie an eine Informatikerin gestellt wird, die *pädagogische, bildungsökonomische und bildungspolitische Praxis betreibt* in einem Umfeld, in dem große Hoffnungen in das Rationalisierungspotential von Informatiksystemen gesetzt werden, einhergehend mit der Erwartung an eine gleichzeitige Verbesserung der Qualität von Arbeitsprozessen und -ergebnissen.

Diese Antwort würde mir gefallen:

Eine (diese!) Theorie kann der Praxis z.B. zeigen, welche Zielvorstellungen sich tatsächlich aus dem Einsatz informatischer Systeme ableiten lassen, welche Behauptungen nur Zuschreibungen an informatische Systeme sind und welche (informatischen *und* nicht-informatischen) Voraussetzungen für Prozesse gelten müssen, um in ihnen durch den Einsatz informatischer Systeme die gewünschten Ziele zu erreichen.

Der Praxis die Theorie auf den Begriff bringen

Eine mögliche Methode, Theorie und Praxis informatischen Handelns miteinander in Beziehung zu setzen, ist die Analyse der – in der IT-geprägten pädagogischen und bildungspolitischen Praxis und Diskussion – verwendeten Begriffe, d.h. eine Reflexion des Gebrauchs populärer Begriffe vor dem Hintergrund ihrer Bedeutung im Rahmen der Theorie.

Das ist für die Wissenschaftlerin nützlich, weil sie überprüfen kann, ob das festgestellte Verhältnis des Anwendungsgebiets zur Informatik durch die vertretene Theorie zu erklären ist. Die Praktikerin erweist der Theorie einen Dienst, wenn sie zeigen kann, dass mit Bezug auf die Theorie für die Praxis etwas gewonnen wird – und sei es "nur" Erkenntnisgewinn, der aber unerwünschte Begleiterscheinungen vorhersehen, die Konzentration auf wesentliche Entwicklungsaspekte lenken und vor Fehlinvestitionen bewahren kann.

Sie betreibt also so etwas wie "Wissenschaftsmarketing". Das könnte z.B. so aussehen: Man schaffe Anlässe, zu begründen, warum "Instrumentale Medien als Basis einer neuen Lernkultur" eine geeignete (weil theoriegeleitete) Betitelung des Geschehens rund um "IuK-Technologien und Bildung" ist,³ zu

¹ Die Autorin war seit Anfang 2000 pädagogisch-technische Leiterin einer Weiterbildungseinrichtung mit IT-Schwerpunkt; seit November 2001 ist sie als Projektkoordinatorin des Multimedia-Kompetenzzentrums mit der Integration virtueller Studienelemente in den Lehr- und Lernbetrieb der Hochschule Bremen befasst.

² (...und die ich, wie am Ausrufungszeichen im Titel zu sehen ist, auch noch nicht wieder in Frage stelle ;-)

³ Z.B. die Bundesfachtagung des Internationalen Netzwerks Weiterbildung e.V., Hamburg, 24.-26.9.00.

begründen, warum gerade diese Begriffe so wichtig sind, um zu verstehen, wie und warum die Informations- und Kommunikationstechniken unsere Kultur verändern und warum wir Aussagen über die Zukunft des Lehrens und Lernens mit informations- und kommunikationstechnischen Mitteln gewinnen wollen.

Die Kurzfassung der Argumentation lautet ja bekanntermaßen so:⁴ (Warum bloß gehe ich davon aus, dass in der AG den meisten Teilnehmerinnen dies bekannt sein wird? Die können jetzt ruhig 1,5 Seiten überspringen.)

Kultur?

Warum reden wir von *Lernkultur*? Der Begriff schließt erstens ein, dass es sich nicht nur um das lernende Individuum dreht, sondern dass Lernen auch immer ein gesellschaftlich (oder kulturell) vermittelter Prozess ist, der durch den Einsatz von IuK-Techniken beeinflusst wird.

Zweitens betonen wir, wenn wir von Lernkultur (und deren Veränderung durch die IuK-Techniken) reden, dass wir es mit einer symbolischen Dimension zu tun haben, mit der Vermitteltheit unseres Umgangs untereinander und unserer Begegnung mit der Welt, die bestimmter Formen bedarf, durch die sich unsere Kultur wiederum konstituiert – dass es um unseren Umgang mit Zeichen geht, der ein anderer wird. Der Zusammenhang von "Kultur" und "Symbol" bzw. "Zeichen" wird von allen Ansätzen, den Kulturbegriff zu definieren, konstatiert. Die Semiotik ist die Wissenschaft, die den geeigneten Begriffsapparat bereithält, kulturelle Phänomene zu erfassen.

Zeichen?

Im Zeichen liegt die Verbindung zwischen Informatik und Kultur. Um diese Verbindung wissenschaftlich behandelbar zu machen, bedarf es eines Zeichenbegriffs, der beiden Disziplinen gerecht wird und alle darin relevanten Aspekte berücksichtigt. (Schon ein Aspekt von Hybridisierung?)

Speziell dort, wo in der Informatik die Benutzung des Computers im Zentrum der Forschung steht, d.h. wo es um Analyse und Konstruktion von Mensch-Maschine-Kommunikation geht, stützen die WissenschaftlerInnen sich auf den Zeichenbegriff von C. S. PEIRCE. Das besondere an seinem Zeichenbegriff ist, dass durch ihn der Mensch als Element eines Zeichenprozesses theoretisch behandelbar wird.

Er beschreibt Zeichen als dreistellige Relation: Ein Zeichen ist "etwas, das für jemanden in gewisser Hinsicht für etwas steht." (Eco 1977, S. 37) Die Zeichenrelation erklärt sich aus der Beziehung zwischen dem wahrnehmbaren *Repräsentamen* (oder dem darstellenden Mittel), dem bezeichneten *Objekt*, allgemein vorzustellen als Gegenstand oder Sachverhalt, und dem *Interpretanten*, der die Beziehung zwischen dem Repräsentamen und dem Objekt erst realisiert. Er nennt die drei Dimensionen auch "*Firstness, Secondness, Thirdness*". Durch diese Universalität der Begriffe wird der Gegenstandsbereich der Zeichentheorie ausgedehnt: Alles kann "Zeichen" sein, was zum Gegenstand dieser Relation wird, was in den Prozess der Semiose eintritt.

Berechenbare Zeichen?

Die Anwendung der PEIRCESchen Zeichentheorie zur Beschreibung informatischer Prozesse setzt voraus, dass wir es in der Informatik, die die "Maschinisierung von Kopfarbeit" betreibt, mit besonderen Zeichenprozessen zu tun haben.⁵ Wir denken bei der Maschine, auf die die Arbeit übertragen wird, an den Computer. Um jedoch dieser Maschine anheimfallen zu können, muss die Kopfarbeit – wie alle Dinge oder Sachverhalte, die einer maschinellen Bearbeitung durch den Computer zugänglich gemacht werden sollen – drei Transformationen durchlaufen:

- Die *Semiotisierung* beschreibt den Vorgang, der die Dinge und Prozesse unseres unmittelbaren Tuns in Zeichen transformiert. Im Falle von Kopfarbeit haben wir es z.B. mit Verlaufsplänen, Texten, Tabellen, Zeichnungen, Listen etc. zu tun, in denen sich Menschen, Materialien, Arbeitsprozesse und ihre Beziehungen untereinander in Form von Zeichen niederschlagen.
- Die *Formalisierung* nimmt den Zeichen ihre Bedeutung tragende Dimension; sie werden standardisiert. Sie werden in eine Form gebracht, die von jeglicher Subjektbeziehung abstrahiert und sich nur auf den Aspekt des Zeichens, Mittel (representamen) zu sein, konzentriert.
- Die *Algorithmisierung* schließlich übersetzt diese von ihren Kontexten maximal befreiten Zeichen und Zeichenprozesse in berechenbare Funktionen.

⁴ Langfassung nachzulesen u.a. in (Wilkens 2000, 2001).

⁵ In der Diskussion um eine Theorie der Informatik wird diese Auffassung zuerst von Frieder Nake (1977) vertreten. In verschiedenen Forschungsprojekten wurde dieser Ansatz auf seine Tragfähigkeit hin untersucht, vgl. Schelhowe 1997, Wilkens 2000. Er mündet in einer Theorie der Informatik als „Technische Semiotik“.

Das Ergebnis solcher Prozesse nennen wir gemeinhin "Software". Mit diesem Konstrukt stellt die Informatik den Aspekten von Welt, die sich als formalisierbar und algorithmisierbar erweisen, ein dem jeweiligen Zwecke entsprechendes Doppel zur Seite, das zunächst als maximal dekontextualisiertes Zeichen seiner eigentlichen Zeichenhaftigkeit beraubt ist.

Informatisches Zeichen und instrumentales Medium?

Zeichen, mit denen die Informatik zu tun hat, sind zugleich Teil berechenbarer Prozesse und menschlicher Interpretation. Mittels Computer, genauer: mittels Software, gehen wir mit Welt um, die einer semiotischen, einer syntaktischen und einer algorithmischen Transformation unterzogen worden ist. Von ihren Kontexten befreit, reduziert auf das reine Datum, werden Dinge und Prozesse als Zeichen der maschinellen Berechnung zugänglich. Hieraus erwächst die besondere Nützlichkeit des Computers für uns, da wir jetzt Welt als Zeichen instrumental bearbeiten können.

Durch den Computer, als technisches Medium der Mitteilung aufgefasst, haben wir es aber nun in Kommunikationssituationen mit einer neuen Klasse von Zeichen zu tun, mit Software, die als quasi-autonomer Sender von Zeichen und Empfänger unserer Zeicheninterpretationen erscheint, weil – anders als bei den herkömmlichen Medien – die Zeichen selbst sich interpretieren, weil in der Maschine Zeichenprozesse stattfinden können, die wir in der Kommunikationssituation nicht nachvollziehen können, deren Ergebnisse wir aber in unseren menschlichen Kontext einbetten.

Das Ergebnis eines berechenbaren, also reduzierten Zeichenprozesses wiederum wird zum Gegenstand vollständiger Semiosen, sobald es durch Menschen interpretiert wird, auf diese Weise in die Kultur hineinwirkt und wiederum auf die Maschine zurückkommt. Hier liegt die mediale Eigenschaft des Computers, die sich erst in der jüngsten Geschichte der Informationstechnik entpuppt.

Also ein Zweig der Semiotik und damit Kulturwissenschaft

Das Zeichen ist also die zentrale Kategorie, die der Wechselwirkung von instrumentaler und medialer Funktion des Computers zugrunde liegt. Der Computer erscheint als Medium, weil er an zwei Arten von Zeichenprozessen teilhat. Neben den maschinellen Zeichenprozessen finden vollständige Semiosen statt, Zeichenprozesse, in denen die "Computerzeichen" in unserem Lebenskontext interpretiert werden und für uns Menschen Bedeutung erhalten. Darin liegt das Besondere des informatischen Zeichens – in seiner Eigenschaft, gleichzeitig Signal für die Maschine und Zeichen für den Menschen sein zu können.

Die Praxis auf den Begriff bringen

Um den Einzug von berechenbaren Prozessen in die herkömmliche Auseinandersetzung mit Zeichen und ihrer Bedeutung im Rahmen von Wissenserwerb geht es in allen Diskussionen über Computer und Internet im Unterricht – als Lerngegenstand und als Lernmittel.

Mit dem theoretischen "Begriffsinstrumentarium" gehe man also auf aktuelle Schlagworte aus der IT-Bildungsdiskussion los und setze sie vor dem Hintergrund einer Theorie der Informatik als Kulturwissenschaft zueinander in Beziehung, z.B. an erster Stelle "Vernetzung" (seit Mitte der 90er Jahre das Universalversprechen an alle Problemlösungssuchenden), dann "Lernumgebung", "Medienkompetenz", "Handlungsfähigkeit", Sie benennen bestimmte Aspekte des Lehrens und Lernens, die wiederum für Veränderungen durch die Computertechnik stehen bzw. diese widerspiegeln, in denen es zu neuen Formen des Zeichenhandelns kommt, wo den Zeichen eine neue Qualität (im Sinne einer neuen Eigenschaft) zukommt und wo die Rollen und Aufgaben von Lehrenden und Lernenden neu definiert werden. Nicht zuletzt soll ja ein Denken darüber angeregt werden, welche Konsequenzen sich daraus für die Organisation von Bildungsprozessen ergeben können bzw. in welcher Richtung nach neuen Antworten zu suchen ist.

Lernumgebung – Daten – technische Vernetzung

Lernumgebungen zeichnen sich zunächst durch nichts anderes als eine Ansammlung von *Daten* aus, auf die nach Bedarf zugegriffen wird. Der Begriff Lernumgebung impliziert ein Bild vom Lernenden als aktiv Handelndem, der unter den angebotenen Daten eine Auswahl treffen kann und muss und seinen Lernweg zum Teil selbst bestimmt. Lernumgebung im weitesten Sinne ist alles das, was uns von Geburt bis zum Tod umgibt und von uns als Zeichen aufgefasst und mit Bedeutung versehen wird. Also *alles* – der Möglichkeit nach. In einer engen Auslegung des Begriffs haben wir es zunächst nur mit Daten zu tun, mit einem Angebot von Möglichkeiten, einem Setting von Signalen, z.T. technisch vermittelt, die – eingebunden in einen Lernprozess – als Information aufgefasst und mit Bedeutung versehen werden können.

Mit dem Attribut "Vernetzung" wird auf eine besondere Qualität von Lernumgebungen verwiesen, auch wenn sie sich zunächst allein in der *technischen* Vernetzung beliebiger Daten – wie z.B. im Internet – zeigt.

Es scheint kein Zufall zu sein, dass die Stärkung der konstruktivistischen Sicht auf das Lernen zeitlich und inhaltlich eng verknüpft ist mit der Entwicklung von Hypermedien für den Unterricht, mit der didaktischen Nutzung der technischen Vernetzung. Der Begriff Lernumgebungen erlangte seine besondere Bedeutung erst im Zusammenhang mit einer Auffassung, die Lernen als "aktiven, selbstgesteuerten, konstruktiven, situativen und sozialen Prozess" ausweist (Tulodziecki 1997, S. 130). Nicht mehr Unterricht als Veranstaltung, in der ein Lehrer die Daten für die Schüler aufbereitet hat und ihnen den Stoff vermittelt, d.h. Unterricht als gemeinsam von Lehrenden und Lernenden zu leistender Prozess der Bedeutungszuweisung gesehen wird, prägt die Vorstellung vom Lernen in der Informationsgesellschaft, sondern die Modularisierung im Sinne einer Loslösung des Materials von Quelle und AdressatIn, auch von einer bestimmten Institution, in der das Lehren und Lernen stattfindet und die für das Gelehrte verantwortlich ist.

Mit der Entwicklung von Speichermedien, die immer größere Mengen von Daten auf kleinstem Raum vorhalten und in kürzester Zeit zugreifbar und darstellbar machen und mit dem Entstehen und Wachsen des Internet als technischer Basis für die Übermittlung von Daten, die weltweit für dieses Medium aufbereitet und an jedem beliebigen Ort verfügbar werden, hat sich fälschlicherweise das Schlagwort von der "Informationsflut" etabliert. Aber wenn wir uns an das Zeichenmodell erinnern, haben wir es zunächst nur mit Daten zu tun, wenn auch in großer Menge, in schneller Verfügbarkeit, technisch verknüpft, fast beliebig veränderbar und indiziert. Diesen Daten unterstellen wir zwar, Information zu sein, d.h. für etwas zu stehen. Diese Information ist aber zunächst eine virtuelle, eine Information "der Möglichkeit nach", die jedoch, um realisiert zu werden, einer Instanz bedarf, die die Daten als Information interpretiert. Daten gehören zur Mittel-Dimension des Zeichens, sie sind zunächst nichts als reine Stofflichkeit, sie können beliebig gedeutet werden. Um Information zu werden, wird die Aktivität von Menschen vorausgesetzt, die die Daten als Zeichen interpretieren und ihnen Bedeutung zuweisen.

Die *Vernetzung* instrumentaler Medien macht die allgegenwärtige Verfügbarkeit von Daten möglich. Ihre Interpretation wird dadurch nicht leichter, im Gegenteil: Wir wissen nicht in jedem Falle, wer der Sender der Daten ist, und wir können nicht mehr sicher sein, dass das, was bei uns ankommt, auch das ist, was der Sender abgeschickt hat. Das Verschwinden des Senders hinter den Daten und das Wissen, dass die Daten im Computer auf dem Weg von Sender zu Empfänger beliebig verändert werden könnten, erinnert an die Eigenschaft von Daten, die sie immer schon hatten, die sich aber erst durch die Entpuppung des Computers als Medium erneut ins Zentrum der Aufmerksamkeit rücken: Daten müssen interpretiert werden, um als Information erkannt und mit Bedeutung versehen zu werden.

Medienkompetenz – Information – inhaltliche Vernetzung

Wenn wir von Lernumgebungen sprechen und dabei an den Einsatz der Neuen Medien denken, impliziert das oft die Vorstellung, dass den verfügbaren Daten bereits eine didaktische Qualität anhaftet, die sich im Interpretationsprozess entfalten kann und soll, dass im selbstgesteuerten Lernprozess also ihre Wirkung erschlossen werden kann. Wir können die Lernumgebung als Teil eines Lernmodells nur akzeptieren, wenn wir in dem Modell auch die methodische und didaktische Aufbereitung der Daten wiederfinden. D.h. es bedarf einer Aufbereitung der Daten, so dass jemand entscheiden kann, ob Daten für etwas stehen, das für sie Information ist, ob diese Information für ihr Ziel relevant ist, ob sie diese Information in ihren Wissenshintergrund einbetten kann, ihr also auch Bedeutung zukommt etc.

Auswahl und Vernetzung kennzeichnen diesen Prozess auf inhaltlicher Ebene. Lernumgebungen mit didaktischer Qualität verlangen also nach besonderer Kompetenz. Es geht darum, das Verhältnis Daten zu Information zu gestalten und zu rezipieren, ihm Bedeutung zuzuweisen, es geht um vollständige Zeichenprozesse. In diesem Sinne ist also Medienkompetenz gefordert.

Medienkompetenz ist mit Bezug zur Auffassung des Lernens als Zeichenprozess also dort gefordert, wo es um die Information geht, um die Beziehung zwischen den "reinen" Daten und den Objekten, Gegenständen, Sachverhalten, für die diese Daten stehen. Medienkompetenz setzt auf Seiten der Lernenden – neben dem technischen Know-How im Umgang mit den Geräten, das zunehmend selbstverständlich wird – voraus, dass sie aus dem gesamten Angebot für sie relevante Daten auswählen und auf ihre Qualität hin bewerten können, z.B. auf Richtigkeit, Wahrhaftigkeit, Bedeutsamkeit für das angestrebte Lernziel etc. Sie müssen Zusammenhänge konstruieren, verstehen, welche Daten auf welche Objekte verweisen, ob sie für sie interessant und wichtig sind und ihnen bei der Erreichung ihres aktuellen Ziels nützen, wie die Qualität, Zuverlässigkeit, Aktualität und Vollständigkeit überprüft werden kann u.v.m.

In Lernumgebungen, in denen die Qualität nicht mehr durch bestimmte Gremien garantiert wird, müssen diese Kenntnisse vorhanden sein bzw. durch "Lernberater" entwickelt oder zur Verfügung gestellt werden. Medienkompetenz auf Seiten der Lehrenden heißt also, die Daten, die in Lernumgebungen zur Verfügung stehen, so auszuwählen und zusammenzustellen, dass die Beziehung zum Bezeichneten vom Lernenden entsprechend der verfolgten Lernziele erschlossen werden kann.

Dies ist eigentlich nichts Neues. Die inhaltliche (oder Sach-)Analyse, die didaktische und die methodische Analyse gehören zum Handwerkszeug jeder Lehrerin. Aber: Die Möglichkeit, *inhaltliche* Verknüpfungen anlegen zu können, die im Lernprozess dann *technisch* realisiert werden können (aber nicht müssen, deswegen "virtuell"), *unabhängig* von einer Lehrperson, verlangt nach der Fähigkeit, die Netze nach didaktischen und methodischen Kriterien zu strukturieren und zu gestalten.

Mittels instrumentaler Medien werden mögliche Lernwege antizipiert und technisch unterstützt. Die Balance, die bei Konstruktion und Einsatz computergestützter Lernumgebungen zu finden ist, ist die zwischen Berechenbarkeit und Beliebigkeit. Für alle institutionalisierten Bildungsprozesse, auch für die Weiterbildung, muss also die Frage entschieden werden, welche Sicht auf das Lernen eingenommen werden soll: Verfolgen wir z.B. den Instruktionsansatz, oder wollen wir das Lernen gemäß einer konstruktivistischen Sichtweise gestalten? Mir dieser Entscheidung wird eine Entscheidung über die Rolle der Lehrenden gefällt – und damit auch, welche Aspekte des Lehrens an die Maschine abgegeben werden können und wollen und für welche Aspekte gerade die Offenheit der Interpretation gefordert oder gewünscht ist.

Handlungsfähigkeit – Wissen – personale Vernetzung

Neben der technischen und inhaltlichen Vernetzung schöpft Weiterbildung mit neuen Medien aus der personalen Vernetzung. Wenn die technische Basis die Kultur verändert, unseren Umgang mit Zeichen, unsere Kommunikation und damit unser Lernen, so sprechen wir immer von sozialen Prozessen. Instrumentale Medien als Basis einer neuen Lernkultur wirken auch in der dritten Zeichendimension: Zur technischen Vernetzung der Daten und zur inhaltlichen Vernetzung der Informationen tritt die personelle Vernetzung der "Träger der Interpretanten", d.h. die soziale Gemeinschaft, vor deren Hintergrund Informationen Bedeutung erlangen und Wissen zur Anwendung kommt.

Instrumentale Medien schaffen technisch vermittelte Netzwerke. Auch wenn das Lernen (und Arbeiten) an einem beliebigen physischen Ort stattfinden kann, setzt Wissen und Bedeutung doch immer auch ein personale Vernetzung voraus. Wissen erlangt seine pragmatische Dimension erst in kommunikativen Zusammenhängen, wenn ich mich mitteilen kann, mein Wissen anwenden kann, neue Zeichen schaffen kann, mich über die Bedeutung des Gelernten in der Gruppe rückversichern kann, wenn sich das Gelernte auf einen gesellschaftlichen oder kulturellen Zusammenhang beziehen lässt.

Die personale Vernetzung wird durch die Möglichkeit der technisch vernetzten Kommunikation ja nicht aufgehoben, sondern erweitert. Die soziale Gemeinschaft (oder Kultur) bleibt nach wie vor der Hintergrund, vor dem die Daten für den einzelnen Bedeutung erlangen, der ihm eine Einordnung der Information ermöglicht, auf dem er sich über die Relevanz des Gelernten rückversichern kann.

Allerdings ist die personale Vernetzung nicht mehr an die physische Gegenwärtigkeit gebunden, nicht einmal mehr an räumliche Nähe. Weltumspannende Interessengemeinschaften intensivieren ihre Kommunikation durch die Möglichkeiten des schnellen Datenaustauschs per Email und FTP. Telelernen und Telearbeit als zeit- und wegsparende Alternative zur physischen Präsenz an einem bestimmten Ort wird als Perspektive für Arbeitsgruppen entwickelt und erprobt. Die Stärkung von Netzwerken durch die Bündelung von Informationen und die konzentrierte Einflussnahme durch die weltweite Vernetzung und Präsenz im Internet sind inzwischen fast klassische Beispiele für die Bildung personaler Vernetzungen, die ohne die neuen Medien in dieser Intensität oder überhaupt nicht möglich gewesen wären.

Gibt es Widersprüche zwischen technisch vermittelter und unmittelbarer Kommunikation? Offensichtlich gibt es viele Bereiche, in denen das eine möglich, das andere aber nach wie vor gewünscht oder notwendig ist. Drei Beispiele mögen die Vielfalt möglicher Entwicklungen nur andeuten:

Die unmittelbare (technisch unvermittelte!) Begegnung beim Surfen im Internet (z.B. in Internet-Cafes) hat einen hohen Stellenwert, um die Informationen, die im Netz verfügbar sind, mit anderen vor Ort teilen und nutzen zu können.

Weltweit verteilte Kooperationspartner wickeln ihre Aufträge online ab und produzieren Medien für das Internet - aber der technisch nicht vermittelten, unmittelbaren Kommunikation vor Ort im Betrieb wird ein so großer Stellenwert bei der Weitergabe von Information und für die Erzeugung und Anwendung von Wissen zugeschrieben, dass Modelle von Telearbeit nur schwer mit dieser Bedeutung der unvermittelten Kommunikation zu vereinbaren sind, selbst alternierende Modelle von Präsenz- und

Telearbeit schwer einzuführen sind. Gerade die Medienbranche scheint sich hier besonders schwer zu tun.

Technisch vermittelte, nur über dieses Medium realisierte personale Vernetzungen wirken zurück in die Region und schaffen neue Anregungen. Globale, technisch vermittelte Netzwerke geben regionalen Arbeits- und Kommunikationsnetzen neue Impulse, Wissenszuwachs und Stärkung der Handlungsfähigkeit. Möglichkeiten der regionalen, personalen Vernetzung werden erst durch die Auseinandersetzung mit den Möglichkeiten des Internet gesehen und umgesetzt, sozusagen als Rückbesinnung auf die Bedeutung des Ortes, so z.B. bei der Nutzung der Arbeit Internationaler Netzwerke für die regionale Entwicklung.

Die Koppelung technisch vermittelter (virtueller) Netzwerke und lokaler bzw. regionaler personaler Vernetzungsstrukturen muss mit dem Ziel eines Qualitätszuwachses in der regionalen (Bildungs-)Arbeit weiterentwickelt werden. Medienkompetenz ist nicht zuletzt auch soziale Kompetenz, die es zu entwickeln gilt, um der Devise des Informationszeitalters in gesellschaftlich verantwortlicher Weise gerecht zu werden: Global denken, lokal handeln.

Braucht Bildung eine Theorie der Informatik?

Bildung muss für ihre Entwicklung die Möglichkeiten der Neuen Medien, deren Kern das instrumentale Medium ist, erschließen. Die Gestaltung und Nutzung von Lernumgebungen mit Computern setzt Medienkompetenz voraus, um die gewünschte didaktische und methodische Qualität von Unterricht zu sichern und weiter zu entwickeln. Die allseits ausgerufene Forderung nach Medienkompetenz stellt an die Praxis der Aus- und Weiterbildung neue Aufgaben und konfrontiert sie mit wichtigen Fragen. Die Fragen sind gar nicht so neu, z.T. mussten sie schon immer für einen verantwortlichen Medieneinsatz beantwortet werden, z.B.: Wie gewährleisten wir die Qualität der Daten und der Lernmedien, die wir in der Lehre verwenden? Wie definieren wir die Rollen von Lehrenden und Lernenden angesichts der Möglichkeiten der Neuen Medien? Wie müssen wir die Inhalte unserer Curricula ergänzen, im Hinblick auf die Rezeption und Konstruktion von Daten und Informationen? Wie entwickeln wir soziale Kompetenz?

Oder: Wie gestalten wir Prozesse, in denen die den Neuen Medien einbeschriebenen (oft auch nur zugeschriebenen) Virtualitäten wie Interaktivität, Mobilität, Konnektivität... sich als Interaktion, Bewegung und Begegnung entfalten – mit und zwischen Ideen und Personen.

Diese Fragen, die die Neuen Medien an die allgemeine Bildung und an institutionalisierte Lernprozesse stellen, müssen vor allem und zuallererst von PädagogInnen (i.w.S.) reflektiert und beantwortet werden! Dabei sollten Sie sich auf den Rat von InformatikerInnen berufen und verlassen können.

In dieser – theoretischen und praktischen – Begegnung der beteiligten Disziplinen scheint eine große Herausforderung für Bildung *und* Informatik zu liegen.

PädagogInnen nützt eine Theorie, die Informatik als Kulturwissenschaft ausweist. Vielleicht... InformatikerInnen auch? Vielleicht besonders diejenigen InformatikerInnen, die aus ihrer Disziplin heraus an Bildungsgeschäft und pädagogische Praxis mit Vorschlägen herantreten – mit Vorschlägen, die nicht als Zumutung, sondern von allen Beteiligten als Bereicherung verstanden werden sollen und als Verbesserung von Lehren und Lernen wirksam werden können.

Literatur

Eco, Umberto: Zeichen. Einführung in einen Begriff und seine Geschichte. Frankfurt a. M.: Suhrkamp 1977.

Nake, Frieder: Informationssysteme als Mittel zur Maschinisierung von Kopfarbeit. Mitteilungen des Instituts für Informatik in der Universität Hamburg Nr. 46, 4.3.1-11, 1997.

Schelhowe, Heidi: Das Medium aus der Maschine. Zur Metamorphose des Computers. Frankfurt a. M.; New York: Campus 1997.

Tulodziecki, Gerhard; Blömeke, Sigrid (Hrsg.): Neue Medien – neue Aufgaben für die Lehrerbildung. Tagungsdokumentation. Gütersloh: Verlag Bertelsmann Stiftung 1997.

Wilkens, Ulrike: Das allmähliche Verschwinden der informationstechnischen Grundbildung. Zum Verhältnis von Informatik und Allgemeinbildung. Aachen: Shaker 2000.

Wilkens, Ulrike: Instrumentale Medien als Basis einer neuen Lernkultur. In: Monika Waldmüller-Blaum; Annette Bromma (Hrsg.): Neue Chancen durch Weiterbildung. 15 Jahre FrauenTechnikZentren. Dreiskau-Muckern: Förderverein für soziale Projekte für Frauen und Jugendliche e.V. 2001, S. 94-108.

Informatik im interdisziplinären Kontext

Erhalt und Entwicklung durch Kooperation

Bericht zur AG Informatik als Hybridwissenschaft

**Dirk Siefkes, unter Mitarbeit von Johannes Busse, Jochen Ludewig,
Rüdiger Valk, Ulrike Wilkens**

Einleitung

Ziel der Arbeit in unserer Gruppe war, unser Selbstverständnis als Informatiker zu klären und damit den Zugang zu einer breiteren Theorie der Informatik zu gewinnen. In der Diskussion gingen die Meinungen über Informatik und Theorie weit auseinander. Trotzdem führte der Versuch, eine Übersicht über Inhalte und Umfang der Informatik zu erstellen, zu einer Vorstellung, wie die Disziplin im wissenschaftlichen Gefüge zu erhalten und zu entwickeln sei. Unter der Sichtweise konnten alle ihre Meinungen einordnen, und wir haben beschlossen, unsere Arbeit auf der Grundlage weiterzuführen.

In diesem Bericht versuche ich, die Hauptpunkte der Diskussion darzustellen und dabei die Linie herauszuarbeiten, die zu dem gemeinsamen Abschluß führte. Das ist nicht ohne Widersprüche möglich. Ich gebe einerseits die unterschiedlichen Positionen möglichst getreu wieder. Manchmal muß ich dazu weiter ausholen, als die Zeit auf der Tagung zugelassen hat; teilweise greife ich auf Positionspapiere und Vorbereitungstext der AG zurück, ohne das immer explizit anzugeben. Beim Lesen ist daher auf Textbrüche zu achten; "wir", gelegentlich "ich", sind wechselnde Teilnehmer der AG, manchmal stellvertretend für alle Informatiker. Andererseits versuche ich, Konvergenzen zu zeigen, wo es sie gab. Ich hoffe, dass meine eigene Sicht die Darstellung nicht zu sehr bestimmt und dass die verschiedenen Stimmen zu hören und zu ordnen sind.

Der Gedankengang lässt sich so zusammenfassen: Inhalte und Abgrenzung der Informatik sind so unklar wie die Ziele unserer Arbeit (Abschn. 1+2). Andererseits haben wir so großen Einfluß auf alle Lebensbereiche, dass wir die Rolle der Informatik klären müssen (Abschn. 3). Die Informatik ist vielfältig gefährdet; sie hat aber kein klar abgrenzbares Territorium, das wir verteidigen könnten. Deswegen betrachten wir sie als kulturelle Entwicklung: Wir können sie nur in Kooperation mit Nachbardisziplinen erhalten und verändern. Fachgebiete sind keine Besitztümer, sondern historisch gewachsenen Einheiten; Inhalte und Methoden dienen der Reproduktion und ändern sich dabei; Grenzbereiche können regional unterschiedlich zugeordnet werden (Abschn. 4). Beim Programmieren hybridisieren wir Mensch- und Maschinenwelt (Abschn. 5). Deswegen ist Informatik keine Wissenschaft vom klassischen oder moderneren Typ, sondern lässt sich nur im interdisziplinären Zusammenspiel charakterisieren. Ihre Aufgabe ist die Entwicklung von Computersystemen mit dem Ziel, die Bedingungen im Einsatzbereich zu verbessern; das kann sie nicht im Alleingang (Abschn. 6). Die Kooperation kann ganz unterschiedlich aussehen, wie die Beispiele Pädagogik, Soziologie und Geschichte sowie Expertensysteme zeigen (Abschn. 7+8). Obwohl unsere Vorstellungen über Informatik und Theorie ganz unterschiedlich sind, haben wir damit auf eine Theorie der Informatik hingearbeitet. Das Ergebnis ist keine Theorie über Informatik, auch keine informatische Theorie, sondern eine aus unserer Arbeit heraus entwickelte Antwort auf die Frage nach ihrem Sinn (Abschn. 9).

Ich danke allen Teilnehmern der AG für ihre Mitarbeit, besonderes denen, die zu diesem Bericht beigetragen haben, und zusätzlich Corinna Bath für hilfreiche Kritik.

1. Was ist Informatik?

Auch 60 Jahre nach der Erfindung des Computers und 30 Jahre nach der Etablierung der Disziplin in der Bundesrepublik Deutschland ist immer noch strittig, was für eine Wissenschaft die Informatik sei. In den Jahren nach ihrer Gründung gab es eine Reihe von Publikationen zur Frage "Was ist Informatik?" (vgl. dazu den Beitrag von Eulenhöfer in SGI98). Der Name "Informatik" – an der TU Berlin "Kybernetik" – war gewählt worden, um die zu bildende Disziplin von der US-amerikanischen "Computer Science" abzusetzen. Im Gegensatz zu anderen Ingenieurdisziplinen wie der Elektrotechnik oder dem Maschinenbau sollte die Computerwissenschaft mehr als die Wissenschaft von ihrem Gegenstand sein. Die meisten Beteiligten kamen aus der Mathematik oder der Nachrichtentechnik. Sie wollten sich einerseits von ihrer Mutterdisziplin abgrenzen, z.B. weil sie dort nicht genügend Unterstützung für ihre Projekte fanden, andererseits die Verbindung zu ihr betonen, um die Wissenschaftlichkeit ihrer neuen Leidenschaft außer Frage zu stellen. F.L. Bauer (Bau74) verknüpft die beiden Herkünfte zu der

Bindestrich-Definition "Ingenieur-Geisteswissenschaft oder, wenn Sie lieber wollen, Geistes-Ingenieurwissenschaft", wobei er die Mathematik kühl zu einer Geisteswissenschaft überhöht. Die Informatiker konstruieren Maschinen, die numerische Verfahren ausführen können, und brauchen dazu wie andere Ingenieure klassische Mathematik. Für die Programmierung legen sie dagegen die logischen Kalküle der mathematischen Grundlagenforschung zugrunde, die den Geisteswissenschaften nahesteht. Auch Wilfried Brauer definiert die Informatik im Studien- und Forschungsführer (Bra73) ganz allgemein durch "(vor allem maschinelle) Verarbeitung von Informationen", charakterisiert sie dann aber mit Bezug auf R. v. Weizsäcker als "Strukturwissenschaft", wobei der Maschinenaspekt verschwindet. Walter Zemanek schließlich knüpft stark an die Ingenieurtradition an, setzt aber die Informatik durch den Vergleich mit der Architektur davon ab: Wie die Architekten sind die Informatiker nicht nur für das technische Funktionieren, sondern ebenso für die gute Verwendbarkeit ihrer Produkte zuständig (Zem71). Damit meint Zemanek allerdings, wie die meisten Architekten, eher die fachliche als die soziale oder gar die moralische Verantwortung.

Die drei Autoren sind sich darin einig, dass das Spezifische der Informatik nicht in der Maschine und nicht in der Mathematik zu suchen ist, sondern im Programmieren: Fürs Programmieren muß man die mathematischen Formalismen und die maschinellen Bedingungen beherrschen. Für das jeweils Einzelne sind Mathematiker bzw. Nachrichtentechniker zuständig, die Zusammenführung beider macht das Eigentliche der Informatik aus. Damit folgen die Autoren doch wieder der US-amerikanischen Tradition: Nachdem John von Neumann im "First Draft" 1945 den "Code" für die Instruktionen an den Computer eher beiläufig am Ende erwähnte, diskutierte Herman Goldstine und er schon 1947/48 ausführlich das Problem, numerische Algorithmen in Maschinenkode zu transformieren, und führten Flussdiagramme als Zwischenstufe ein. Daraufhin entwickelte sich Kodieren als eine Kunst, noch ohne wissenschaftlichen Anspruch, aber mit allen Anzeichen eines Geheimkults. Dieses eigenständige Tätigkeitsfeld verschwand nach zehn Jahren wieder, als John Backus mit seiner Gruppe bei IBM Fortran entwickelte – einen Formalismus, der als quasi-mathematische "Sprache" ausdrücklich vom Maschinenkode abgegrenzt wurde. Gegen Ende der 50er Jahre führte in Europa eine Gruppe um F.L. Bauer Algol ein – eine Sprache, deren Programme zwar rechnerausführbar, aber ausdrücklich maschinenunabhängig sein sollten, und die hauptsächlich der wissenschaftlichen Kommunikation dienen sollte. Damit war der Anspruch etabliert, dass Programmieren eine wissenschaftliche Tätigkeit sei. (Vgl. dazu SGI97 ff, Eul99, Sta01.)

Bei der Einführung der Disziplin in der Bundesrepublik Deutschland zehn Jahre später wurde Programmierung mit anderen, davon abgespaltenen Fächern zum "Kern" der Informatik zusammengefaßt, zwischen den Pfeilern der Theoretischen und der Technischen Informatik. Als vierten Bereich gab es die Angewandte oder Praktische Informatik, in der es um Programmsysteme für spezielle Anwendungen geht. Nach den Wirkungen z.B. auf Arbeitssituationen, die solche Anwendungen haben, oder gar nach den außer-informatischen Kenntnissen, die sie voraussetzen, wurde selten gefragt. An einigen Universitäten wurden Professuren für "Informatik und Gesellschaft" eingerichtet; das Fachgebiet blieb aber umstritten, es paßte vielen nicht in das Bild von der Disziplin. Auch die Zusammenarbeit mit anderen geistes- und sozialwissenschaftlichen Fächern blieb eine Aktivität Einzelner.

Der Kampf um die "Disziplinierung" (Feyerabend) schien damit abgeschlossen. Die Disziplin "Informatik" ist jetzt an den meisten Hochschulen vertreten, an Technischen Hochschulen und Universitäten als Ingenieurwissenschaft, an klassischen Universitäten meist in Verbindung zu Mathematik und Naturwissenschaften. Das Grundstudium ist durch Bemühungen des Fakultätentags Informatik ziemlich einheitlich. Neben mathematischen und technischen Grundlagen werden vor allem Kenntnisse in Programmierung vermittelt; meist gibt es ein Nebenfach, am häufigsten wird Mathematik, Elektrotechnik oder Wirtschaftswissenschaften gewählt. Das Hauptstudium ist unterschiedlich strukturiert; meist steht es in engem Zusammenhang mit den jeweiligen Forschungsaktivitäten und ändert sich entsprechend schnell. Der Bezug zur Praxis wird dabei als ein doppelter gesehen: Studenten sollen für die Praxis taugen, sie in Zukunft tragen; die Praktiker stellen daher Forderungen an die Lehre. Die Forschung soll den Praktikern helfen, Probleme zu meistern; die Praxis stellt daher eine Herausforderung für die Forschung dar, deren Auswirkungen sich wieder in der Lehre niederschlagen. In diesem Idealbild entwickeln sich Wissenschaft und Praxis in Wechselwirkung: Auf beiden Seiten werden Probleme und Lösungen entwickelt und ausgetauscht, die sich in der praktischen und der wissenschaftlichen Arbeit festigen und bewähren und so einen Fundus von neuen Techniken und Methoden sowie Wissen darüber anwachsen lassen, der in der Ausbildung, durch Publikationen und in Projekten weitergegeben wird. Daß es Fähigkeiten und Arbeitsgewohnheiten, also Menschen sind, die sich dabei entwickeln und diesen Fundus produzieren, wird seltener gesehen.

Die Menschen, also die Informatiker, stellen aber in den letzten Jahren die Frage "Was ist Informatik?" wieder dringlicher (z B. Wil96, Rec00, Des01). Informatiker sind mit schuld daran, dass sich in einer unkritischen Öffentlichkeit, insbesondere in Politik und Verwaltung, die Vorstellung festgesetzt hat, alle

Probleme seien mit Computern lösbar. Dadurch wachsen die Probleme, mit denen sich Informatiker konfrontiert sehen, schneller als der Fundus von Möglichkeiten, damit umzugehen. Außerdem wird immer deutlicher, dass Erfolg und Misserfolg des Einsatzes von Informatiksystemen stark von der intimen Kenntnis des Einsatzbereichs abhängen. Da Informatiker solche Kenntnis in der Regel weder mitbringen noch erarbeiten können, fordern Praktiker von der Wissenschaft zunehmend Ausbildung auch in "sozialer Kompetenz", nicht nur in technischer. Das soll Informatiker befähigen, Informatiksysteme zusammen mit den "Betroffenen" des Einsatzbereichs zu entwickeln. Das macht aber den Einfluß des Nichttechnischen auf die Technikentwicklung erst recht deutlich. Informatiksysteme sind genauso stark durch soziale Anforderungen geprägt, wie sie die gesellschaftliche Entwicklung bestimmen. Man kann "das Soziale" und "das Technische" nicht trennen, also kann man es auch nicht getrennt lehren. Beides können aber die wenigsten Wissenschaftler vermitteln. Informatiker sind zu stark technisch sozialisiert, und Sozialwissenschaftler können sich nicht so in Technik einarbeiten. Soweit Wissenschaft traditionell separiert betrieben wird, können Informatiker daher soziale Kompetenz von Geistes- oder Sozialwissenschaftlern so wenig lernen wie im eigenen Fach; das gilt für Wissenschaftler wie für Studenten. Dasselbe gilt dann aber für technische Kompetenz, auch wenn Informatiker das genauso wenig akzeptieren wie andere Ingenieure. So können sie die traurige Erfahrung machen, dass "Betroffene" ihnen beim Umgang mit Computern ebenbürtig oder überlegen sind, aber keine Ahnung von den einfachsten Prinzipien des Systementwurfs haben, während ihnen selber trotz aller Brillanz in Kommunikation und Management die Kultur des Anwendungsbereichs fremd bleibt.

Auch in der eigenen Wissenschaft fühlen Informatiker keinen festen Boden unter den Füßen. Die beiden ältesten Teile, Theoretische und Technische Informatik, sind ihren Ursprungsdisziplinen Mathematik und Nachrichtentechnik methodisch enger verbunden geblieben als der Informatik selbst; zudem ist die diskrete Mathematik der Theoretischen Informatik getrennt von der kontinuierlichen der Ingenieurwissenschaften. Frühere Kernbereiche wie Betriebssysteme und Compilerbau sind längst in einer Vielzahl anderer für spezielle Systemtypen untergegangen, und Softwaretechnik hat sich als zentraler Bereich der Informatik etabliert. Sie ist ohne Bezug zu formalen Methoden und technischen Gegebenheiten nicht zu betreiben; der Wert rein mathematischer oder ingenieurwissenschaftlicher Methoden in der Softwaretechnik ist aber beschränkt, und beide wollen sich nicht zu einem informatischen Bereich zusammenfügen. Ein eigener Kanon von Inhalten und Methoden hat sich nicht herausgebildet.

Noch schlechter sieht es mit der Abgrenzung nach außen aus. Anwendungen nutzen den Computer als Medium, nicht als spezifisches Gerät; sie werden daher methodisch stark vom Anwendungsgebiet bestimmt. Mit jedem Bereich, in den Computer Einzug halten, erhebt sich daher neu die Frage, ob die zuständigen wissenschaftlichen Disziplinen einen entsprechenden Anteil an die Informatik abzugeben oder umgekehrt die Informatik die neu entstehenden technischen Gebiete abzuspalten habe. Auf die Frage "Was ist Informatik?" zu antworten "Was Informatiker tun" bringt uns also nicht weiter. Ob das, was Informatiker (und andere) tun, Informatik sei, ist zunehmend unklar. Erst recht gibt es kein Einvernehmen über die "eigentlichen" Fragen: Wozu betreiben wir Informatik? Welchen Einfluß haben wir? Was richten wir damit an? Wohin entwickelt sich die Disziplin? Wie können wir die Entwicklung beeinflussen?

2. Wozu betreiben wir Informatik?

Informatik ist unser Beruf; wir alle müssen damit Geld verdienen. Für die wenigsten ist das aber der einzige Grund, Informatik zu betreiben. Und wir alle haben Ziele, die darüber hinaus gehen. Können wir lange ohne Schaden unser Leben mit etwas füllen, in dem wir keinen Sinn sehen? Worin diese Ziele liegen könnten, darüber gingen die Meinungen in unserer AG weit auseinander. Aber für alle schien Theoriebildung mit Sinngebung verknüpft zu sein.

Teilweise waren die Antworten ganz allgemein und entsprechend zögerlich. Wir Informatiker wollen mit KI-Systemen den Menschen helfen; die Aussichten sind aber bedrohlich: Vielleicht tragen wir dazu bei, dass Cyborgs die Herrschaft auf der Erde übernehmen. Wir wollen mit unserer Arbeit die Gesellschaft verändern; aber in welche Richtung?

Teilweise bezogen sich die Antworten spezifischer auf die eigene Tätigkeit und waren dementsprechend persönlicher und überzeugter. Unsere Gesellschaft ist ohne wissensbasierte Systeme nicht mehr denkbar; ich muß als Informatiker kompetent sein, um in Zusammenarbeit mit Experten anderer Gebiete solche Systeme zu entwickeln. Mit Hilfe gewisser Informatikmethoden, z.B. der Modellierung mit Petrinetzen, können Sozialwissenschaftler ihre Theorien besser verstehen; ich will also solche Methodenkompetenz exportieren.

Zur Arbeit von Informatikern im Bildungsbereich gab es zwei ganz verschiedene Antworten. Zum einen: Meine Erfahrung als Informatiker im formalen und technischen Umgang mit Wissen und Sprachen (z.B. Expertensysteme, Programmiersprachen) kann Erziehungswissenschaftlern helfen, ihre Vorstellungen bezüglich dieser Grundkategorien zu klären und den neuen Gegebenheiten anzupassen. Umgekehrt sind pädagogische Konzepte und Theorien für die Arbeit von Informatikern in den genannten Bereichen hilfreich. Zum anderen: Als Informatikerin strebe ich mit Hilfe geeigneter Anwendungen in Bildungszusammenhängen Qualitätssteigerung in vielerlei Hinsicht an, zum Beispiel (Wil00, Wilkens Positionspapier)

- Erleichterung der Arbeit von Lehrenden durch Flexibilisierung von Arbeitsprozessen, Erweiterung der Gestaltungsmöglichkeiten von Materialien, Befreiung von lästiger Routine: Autorensysteme, Office-Programme, Grafikanwendungen, Planungswerkzeuge...
- besserer und schnellerer Zugriff auf berufsrelevante Informationen: Suchmaschinen, Rechercheunterstützung, Agenten im Internet...
- Verbesserung des Kenntniserwerbs durch multimediale Anreicherung, Simulationssysteme usw...
- Senkung der Kosten von Bildung durch die Erweiterung der Präsenzlehre um telemediale Lernszenarien (nach Unterscheidung in betreuungsintensive und selbstlerngeeignete Unterrichtskomponenten)...

Die im 1. Abschnitt geschilderte Unklarheit der Situation in der Informatik kann auch die Frage "Wozu?" dringlich machen: Früher habe ich nur das positive Potential gesehen, das in Informatikanwendungen steckt; heute sehe ich mehr und mehr negative Auswirkungen. Menschen sind nicht rational; aber heute verschwenden auch Firmen ihr Geld mit nutzloser Software, statt ihren Profit zu maximieren. Kann ich Informatik überhaupt mit gutem Gewissen betreiben?

Die Antworten lassen sich so zusammenfassen: Entwicklung und Anwendung von Informatiksystemen geschieht mit dem Ziel, bestimmte Bedingungen des Einsatzbereichs zu verändern: zu effektivieren, zu beschleunigen, zu ermöglichen, zu erleichtern. In jedem Falle lässt sich ein Gesichtspunkt finden, unter dem sich die Veränderung als *Verbesserung* auffassen lässt. Mit diesem Ziel wird in der Informatik geforscht und gelehrt. Aber die Vorstellungen und Urteile darüber, was eine Verbesserung sei, können unterschiedlich bis gegensätzlich sein – insbesondere zwischen Entwicklern und Anwendern.

Ein Ziel einer Theorie der Informatik könnte also sein, dass wir uns klar werden, was wir als Informatiker tun und was wir damit bewirken. Dann können wir eher entscheiden, ob unsere Arbeit sinnvoll oder sinnlos ist, ob sie positive oder negative Folgen hat, und können versuchen, nicht nur einzelne Anwendungen, sondern die Profession insgesamt zu verbessern.

3. Welchen Einfluß haben wir? Was richten wir an?

Auch hier lagen die Meinungen weit auseinander. Die Lehrenden waren sich einig, dass sie, wenn überhaupt jemanden, dann die Studenten beeinflussen. Wir wirken als Multiplikatoren, sowohl fachlich wie persönlich: Wir verbreiten informatisches Wissen, beeinflussen dabei das Entstehen von Fähigkeiten – oder Unfähigkeiten. Dabei verbreiten wir aber auch unsere Sicht- und Herangehensweisen, unsere fachlichen, politischen, persönlichen Standpunkte und Bewertungen, die die Studenten übernehmen oder ablehnen; so beeinflussen wir das Entstehen von Persönlichkeiten – oder Unpersönlichkeiten.

Welchen Einfluß wir in der Forschung oder praktischen Arbeit ausüben, war umstritten. Als Informatiker verbreiten wir Techniken, wirken wieder als Multiplikatoren. Das gilt für "Denktechniken", die wir durch unsere Art zu formalisieren in Umlauf bringen, ebenso wie für materiale Technik. Aber, wurde dagegen gesetzt, die Technik entwickelt sich selbständig – wie, das liegt außerhalb unserer Reichweite. Unsere ganze Kultur – technisch oder nicht – hat sich durch die Nutzung von Computern grundlegend geändert; aber die Entwicklung ist nicht von Informatikern gesteuert worden. Heute gehen die wesentlichen Anstöße von anderen Gebieten aus; wir Informatiker sind nur die nützlichen Idioten, die die technischen Möglichkeiten liefern. Früher hat man Einflussmöglichkeiten vielfach an der Idee der solidarischen Aktion aufgehängt; heute scheint diese Vorstellung völlig unreal. Solidarität ist ein aussterbender Begriff. Es ist nicht auszuschließen, dass wieder einmal eine Zeitenwende kommt, die zu einer anderen Bewertung führt; aber wir sehen sie nicht. So haben wir Informatiker wenig Effekt nach außen. Wir kreisen um uns selbst, daher kreist die Disziplin in sich selbst.

Das Dilemma lässt sich auflösen, wenn wir betrachten, wie Änderungen sich ausbreiten. Wenn wir etwas ausrichten wollen, müssen wir zunächst unsere Umgebung überzeugen – aber so, dass die derart Überzeugten die Veränderung mitnehmen und in ihren Umgebungen weitergeben. Mit Energie, Taktik, Glück und Geduld können wir so unsere Ideen und Techniken um den Erdball wandern sehen. Wir müssen nicht nur große Vorlesungen halten und dicke Bücher schreiben; in Sprechstunde oder

Tutorium haben wir oft mehr Einfluß. Auf den Fahnen der Professoren richtet "Informatik und Gesellschaft" mehr aus denn als Pflichtfach im Studium. (Sie92 ff.)

Dasselbe gilt für die praktische Arbeit mit Informationstechnik. Wir schaffen durch unsere Systeme *Möglichkeiten* der Veränderung, bringen Leute auf Ideen, erzeugen Bedürfnisse, regen Vorstellungen über Veränderungen an. Unsere Systeme erzeugen Zustimmung und Ablehnung. Wo positive Begründungen für Veränderungen durch Informationstechnik gefunden und vertreten werden, kommen die Systeme zum Einsatz. Einen "verantwortlichen Umgang" können wir nicht garantieren, da setzen wir auf die Professionalität der Anwender. Aber wir können vermitteln: den Einsatz unserer Systeme mit Informationen begleiten, durch sachliche Aufklärung versuchen, verschiedenen Formen von Zuschreibungen, Überhöhungen, Mythen, Missverständnissen, Halbwissen zu begegnen. Oft haben wir auch wenig Einfluss; andere gesellschaftliche Kräfte benutzen die Informationstechnik, um durch Zuschreibungen an diese Technik Veränderungen voranzutreiben, von denen sie sich Vorteile versprechen und die sie nur deswegen als Verbesserungen hinstellen.

Wir streben immer Verbesserungen an; darum ging es schon bei den Zielen. Wir haben aber keinen allgemeinen Maßstab, was besser oder schlechter sei; es kommt darauf an, dass wir das Ziel vor anderen und uns selbst vertreten können. Wir müssen nicht Politiker werden oder umwerben; als Mitentwickler einer "defining technology" sind wir immer politisch, auch wenn wir uns unpolitisch geben. Ethische Leitlinien für die Disziplin zu entwickeln, ist wichtig, um anderen klar zu machen, dass uns Verantwortung für unser Tun am Herzen liegt. Ob sie verpflichtend sind, ist nicht entscheidend; sie wirken mehr dadurch, dass wir sie beachten und weitergeben.

Diese Einstellung bringt allerdings mit sich, dass wir nicht einmal vage ermessen können, welche Wirkungen unser Tun tatsächlich hat. Es ist völlig illusorisch, anzunehmen, unser (positiver) Einfluss könne es mit den vielen anderen, aus unserer Sicht negativen Einflüssen an Stärke aufnehmen. In diesem Sinne sind wir ohnmächtig. Die moralische Anleitung zum Handeln kann sich nicht am Erfolg orientieren, sondern an einem moralischen Konjunktiv: Welches Verhalten wäre sinnvoll, wenn unser Verhalten insgesamt eine Chance auf Erfolg hätte?

Den stärksten Einfluss haben wir auf uns selbst. Nicht, dass wir uns gestalten oder gar konstruieren könnten; wir sind keine IT-Systeme. Aber wir überzeugen nur, wenn wir selber überzeugt sind. Daher prägen wir uns selbst. Und wenn wir gegen unsere Überzeugungen handeln oder keine haben wollen, prägt uns das erst recht.

4. Informatik als kulturelle Entwicklung

Eine Organisation, die nicht untergehen will, muss Selbsterhaltungskräfte mobilisieren. Denn es gibt selten (oder nur sehr kurz) eine Umgebung, die ihr die Existenz gönnt. Folglich stellt sich auch für die Informatik täglich die Frage, wie sie sich als Institution verteidigen kann. Uns allen liegt die Erhaltung der Informatik am Herzen – nicht nur, weil wir uns sonst einen neuen Beruf suchen müssten. Wir stehen unserer wissenschaftlichen Heimat teilweise sehr kritisch gegenüber; aber diese Kritik gilt anderen Disziplinen ebenso oder mehr. Manche halten zum Beispiel die Informatik für zu formalistisch oder technikzentriert. Wir können aber dem Schicksal keine Bedingungen stellen, etwa in der Art, dass wir eine Fortdauer der Informatik wünschen, falls sie diese oder jene Bedingungen erfüllt.

Man kann Disziplinen mit Staaten vergleichen. Solange sich alle einig sind, brauchen sie keine Waffen. Aber gegen Bedrohungen von außen oder innen müssen sie sich verteidigen. Und die Informatik ist vielfältig bedroht, heute in der Zeit der Sparzwänge mehr denn je. Der Kampf mit den Nachbardisziplinen um Ressourcen ist härter geworden, teilweise existenzgefährdend. In der Außensicht wird Informatik mit Programmieren gleichgesetzt; es genügt also, wenn die anderen Disziplinen sich ihre Hausinformatiker halten, die Informatik als Disziplin ist damit überflüssig. Mit dem Hinweis auf Globalisierung soll das Diplom durch Abschlüsse wie Bachelor und Master zunächst ergänzt, später sicherlich ersetzt werden. Aus Angst, den Anschluß zu verlieren, beteiligt sich die Informatik am Rennen und setzt damit ihre Diplomstudiengänge aufs Spiel. Die Konsequenzen kann man drastisch ausmalen: Die Unterschiede zwischen Universitäten, Technischen Hochschulen und Fachhochschulen werden nivelliert; massenhafte Ausbildung für die Praxis wird gefördert, die wissenschaftliche Ausbildung und damit die Forschung werden bald an einige (Privat-)Universitäten für die Elite verdrängt. Von der anderen Seite droht die Wirtschaft in Akademien und Corporate Universities spezifische Studiengänge für den eigenen Bedarf aufzubauen und damit die staatliche Berufsausbildung in Informatik überflüssig zu machen. Die Lage wäre weniger bedrohlich, wenn die Informatik in sich geschlossen wäre. Im 2. Abschnitt haben wir aber gesehen, dass die Informatik weder inhaltlich noch methodisch zu charakterisieren ist. Sie hat keine klaren Grenzen, die sie verteidigen könnte. Die Metapher vom Staat Informatik ist für die Mobilisierung der Selbsterhaltungskräfte also ungeeignet.

Wir haben stattdessen die Informatik als Kultur betrachtet. Kulturen brauchen einen Lebensbereich, aber nicht notwendig ein eigenes Gebiet; immer leben sie zwischen und mit anderen Kulturen. Wenn sie sich nicht von ihren Nachbarn abgrenzen, gehen sie in ihnen auf; wenn sie sich gegen Angriffe nicht verteidigen, werden sie geschluckt. Aber wenn sie sich abschotten, um ihre Identität zu wahren, erstarren sie; wenn sie andere angreifen, um sich zu bereichern, verlieren sie ihre eigentlichen Ziele. Sie ziehen ihre Stärke und Entwicklungsfähigkeit nicht aus Aggression, sondern aus Eigenständigkeit und Kooperation. Sie gewinnen und erhalten ihre Identität aus Traditionen, die sich in Riten und Mythen, in Sitten und Festen äußern. Diese Traditionen werden mündlich und schriftlich weitergegeben, aber reproduzieren sich vor allem im Gebrauch; ohne praktizierende Menschen werden sie zu Museumsstücken. Daher sind Kulturen regional unterschiedlich ausgeprägt, abhängig von Umgebung und Geschichte; Teilgebiete können sich voneinander stärker unterscheiden als von denen benachbarter Kulturen. So entwickeln sich Kulturen, manchmal schnell, oft überraschend; so bleiben sie überlebensfähig.

In dieser Vorstellung braucht die Informatik einen festen Bestand von Methoden, Inhalten und Lehrgegenständen, um überleben zu können. Diese Kanons sind aber nicht ewig, sondern reproduzieren sich im Gebrauch durch die beteiligten Menschen und verändern sich dabei. Es gibt wie in Kulturen keinen hochrangigen Kern, umgeben von Randgebieten, die weniger wichtig wären. Theoretische und Technische Informatik sind Teilbereiche der Informatik, die ihre Methoden aus den Ursprungsdisziplinen beziehen, aber ihre Identität aus der Informatik. Löste man sie aus der Informatik heraus, verlören sie Antrieb und Richtung ihrer Entwicklung. Neue Teilgebiete wie wissensbasierte Systeme, Datenschutz, CSCW, Neue Medien in der Ausbildung, Internet oder Sozionik sind dagegen anderen Disziplinen wie Psychologie, Jura, Arbeitswissenschaften, Pädagogik, Organisationstheorie oder Soziologie ebenso nahe wie der Informatik. Sie leben aus der Verbindung zu jeweils mindestens zwei Disziplinen und können in verschiedenen Hochschulen der einen oder der anderen zugeordnet sein. Die Informatik entwickelt sich in solchen regionalen Kooperationen nachhaltiger als durch die globale Akquisition neuer Kolonien; sie ist damit zukunftsfähiger. Auch Gebiete wie Informatikgeschichte, Mensch-Maschine-Kommunikation oder Theorie der Formalisierung können von Informatikern allein genauso wenig betrieben werden wie von Historikern, Psychologen oder Philosophen.

Um uns ein Bild von der Informatik zu machen (1. Abschnitt), haben wir in der AG mit einer Landkarte von Themen und Fachgebieten begonnen. Wir konnten uns aber nicht einigen: Zuordnung und Abgrenzung, Nähe oder Abstand waren in vielen Fällen strittig, ebenso die wechselseitigen Bezüge. Die Einschätzung hing von der eigenen Positionierung und von der Blickrichtung – z.B. Vergangenheit, Gegenwart oder Zukunft – ab. Die Unstimmigkeiten lösten sich erst, als wir von der statischen zu einer dynamischen und von der Staat- zur Kulturmetapher übergingen. Informatik als kulturelle Entwicklung zu betrachten, heißt dreierlei: Die Grenzen der Disziplin nicht als zu befestigende Gräben, sondern als Verkehrsflächen auffassen; die Karte der Gebiete als veränderlich betrachten; und die Grenzgebiete als regional unterschiedlich zugehörig ansehen. Fachgebiete sind keine Besitztümer, die den Wert der Informatik ausmachen. Sie sind generische Einheiten und damit so wichtig, wie sie bereit und in der Lage sind, die Entwicklung zu tragen und zu fördern. Grenzgebiete spielen eine besondere Rolle. Sie ziehen viele junge Leute an, die sich nicht mit "der Informatik" identifizieren würden. Und der intensive Kontakt mit anderen Disziplinen bringt neue Ideen, manchmal Revolutionen. Ohne Gebiete, die "schon immer" dazugehört haben, gäbe es "die Informatik" nicht; aber auch sie müssen sich ständig fragen (lassen), was sie zur Entwicklung der Informatik beitragen.

5. Programmieren als Hybridisieren

Unsere Kultur ist so mit Technik durchtränkt, daß wir Mensch und Maschine kaum als unterschiedlich erleben. Mehr und mehr Aufgaben werden auf Maschinen übertragen, und Menschen gehen mit Problemen immer maschinenartiger um, ob sie nun Maschinen benutzen oder nicht. Das gilt in besonderem Maße für Computer, mit denen wir geistige Aktivitäten maschinisieren, nicht mehr nur körperliche wie mit klassischen Maschinen. Computer sind so Bestandteil unseres täglichen Lebens geworden, dass wir sie als Partner, als Gegenüber empfinden. Bei genauerem Hinsehen sind Mensch und Computer aber durchaus gegensätzlich. Computer sind gut in Bereichen, in denen Menschen schwach sind – wie, sinnlose Anweisungen ausführen, ohne zu zögern oder zu ermüden oder zu protestieren oder abzuweichen oder abzuschweifen. Sie versagen dagegen, wenn Intuition eine Rolle spielt – wie in emotionalisierten oder sonstwie irrationalen Situationen.

Von Anfang an ist das Problem beim Programmieren gewesen, die gegensätzlichen Welten von Mensch und Maschine zusammenzubringen (Brö97). Zunächst schienen beide kaum getrennt, da es nur um numerische Berechnungen ging. Mit den Formularen der Rechenbüros waren die mathematischen Verfahren leicht in Maschinenbefehle zu "kodieren", Computer wurden als direkte Nachbauten menschlicher Rechner an der Tafel oder an der Tischrechenmaschine gesehen (Zuse

1936, von Neumann 1945, Walther 1953). (Kritisch reflektiert wurde die Analogie zwischen Neuronen und Schaltelementen, also zwischen Gehirn und Maschine (Wiener 1948, von Neumann 1948). Turings theoretische Maschinen (1936) beeinflussten den Rechnerbau nicht, produzierten aber bei von Neumann (1948) und ihm selbst (1950) die ersten Phantasien von selbstreproduzierenden Automaten bzw. denkenden Maschinen.) Bald wurde mehr Formalismus nötig, um die Wege zwischen Computern und Menschen zurückzulegen, die immer tiefer ins menschliche Territorium hineinführten. Diese Entwicklung ist im 1. Abschnitt skizziert.

Mit der Entwicklung der ersten Programmier"sprachen" Fortran (1957) und Algol (1960) wurde die Sprach- von der Rechnerentwicklung getrennt, zunehmend wurden Formalismen aus Mathematik und Logik zur Programmierung und ihrer Unterstützung benutzt. Der Computer wurde aus einer Maschine, die zu bedienen ist, zu einem Werkzeug, das man handhabt, um Anwendungen zu gestalten. Gleichzeitig traten Bilder neben Wörter als Mittel der "Mensch-Maschine-Kommunikation". So wie der Rechner (als Mensch *und* als Maschine) bei Turing Symbole (und nichts sonst) manipuliert, manipuliert er jetzt icons auf dem Bildschirm, um Aktionen auszulösen und zu steuern. Daß im Hintergrund eine Maschine läuft, ist kaum noch bewußt. Visuelle und motorische Schemata werden direkt angeregt, ohne Umweg über das Kognitive; und regen direkt zum Handeln an.

Auf der sprachlichen Seite treffen sich die beiden Entwicklungen in der objektorientierten Programmierung. Der Anwendungsbereich wird durch interagierende Objekte auf dem Bildschirm dargestellt, die mathematisch beschreibbar sind, aber wie reale Objekte erscheinen. Nicht nur menschliche und maschinelle Realität verschwinden aus dem Blick, sondern auch die sie verbindende der abstrakten Datenstrukturen. – Auf der bildlichen Seite kulminieren die beiden Entwicklungen im Internet. Der Cyberspace kennt nur eine Realität, die virtuelle, so wirkmächtig wie scheinbar. Der Computer wird dabei aus der Maschine zum Medium (Sch97).

Es ist kein Versehen, wenn wir Menschen und Computer gleichsetzen. Alle unsere Aktivitäten, körperliche oder geistige, kommen aus Routinen, die wir ererbt oder erworben haben; sie schaffen oder verändern oder überdauern die Phänomene, geistige wie körperliche, zwischen denen wir leben; und aktivieren damit andere Routinen. Auf diese Weise reproduzieren sich Routinen – auch Schemata, Gewohnheiten oder Programme(!) genannt – ständig und verändern sich langsam dabei. Unsere Routinen charakterisieren uns, als Individuen, Gruppen, Institutionen, Gesellschaften. Aber genauso charakterisiert uns die Fähigkeit, aus ihnen auszubrechen. Menschen führen Routinen kreativ aus. Wahrnehmen können wir in jedem Fall nur die Phänomene und Aktivitäten; die Routinen liegen darunter verborgen.

Wenn wir eine (reale oder gedachte) Aktivität auf die Maschine bringen wollen, formalisieren wir zuerst die zugrunde liegenden Routinen, d.h. wir beschreiben sie als unabhängig von den entsprechenden Phänomenen. Dann realisieren wir die Beschreibungen in mechanischer oder elektronischer Form. Maschinen führen Routinen wie vorgeschrieben aus. Wenn wir aber Maschinen konstruieren oder benutzen, nehmen wir nur ihre Aktionen wahr oder denken sie uns nach den Beschreibungen.

Aus Zeichen, die Menschen verstehen und die sie zum Handeln anregen, werden im Computer Signale, die Aktionen auslösen. Aus Kommunikation wird Datenübertragung, aus Denken Datenverarbeitung, aus Rechenautomaten werden Datenverarbeitungsanlagen oder gar Elektronengehirne (Nak93, Wil00, Nake in NRS01, Bau01). Und wie das menschliche Gehirn souverän den menschlichen Körper und seine Aktivitäten zu beherrschen scheint, können Computer beliebige Maschinen steuern. Im Computer übertragen wir also scheinbar beliebige menschliche Aktivitäten auf die Maschine, indem wir ihnen eine "Zeichenhaut" überziehen (Nake in Coy92). Wenn wir Computer entwerfen, programmieren oder benutzen, müssen wir menschliche Aktivitäten, insbesondere geistige, in Maschinenläufe umzeichnen und umgekehrt. Wir können die Aktionen der Maschine aber nur als Aktivitäten lebendiger Wesen auffassen; anderes lässt unsere Erfahrung nicht zu. Im Umgang mit dem Computer *hybridisieren* wir die gegensätzlichen Welten von Menschen und Maschinen mit Hilfe formaler Ausdrücke (SGI97b, Eul99, Sta01, Siefkes in Bau01).

Denken und arbeiten, lernen und Erfahrungen machen können wir nur in der menschlichen Welt. Beim Umgang mit Computern übertragen wir unsere Vorstellungen in Zustände und Abläufe von Maschinen und umgekehrt maschinelles Vorgehen in unsere Erfahrungswelt. Der Weg geht durch die Zwischenwelt der Zeichen: Algorithmen und gewisse allgemeinere formale Anweisungen sind von Maschinen ausführbar; gleichzeitig sind formale Beschreibungen von Menschen nachvollziehbar. Wir benutzen also formale Ausdrücke, um Maschinen in Gang zu setzen und um Maschinenvorgänge zu verstehen; eins geht nicht ohne das andere.

Auch in anderen Ingenieurwissenschaften benutzt man mathematische Formalismen, um Maschinen zu beschreiben und zu konstruieren. Dabei werden aber körperliche Vorgänge erfaßt (oder allgemeiner

physikalische Abläufe), nicht direkt geistige Prozesse. Nur in der Informatik werden geistige und körperliche Vorgänge in der Maschine (nämlich in Computersystemen) nachgebildet – der ganze Mensch kommt aufs Reißbrett, nicht ein Körper(teil). Hybridisierung mit Hilfe der Formalismen der mathematischen Logik und diskreten Mathematik ist spezifisch für die Informatik (Sie99 ff).

Wir müssen also Wesen und Wirksamkeit informatischer Zeichen (= Software) auf ihren Doppelcharakter zurückführen. Sie sind gleichzeitig Signal (für die Maschine) und Zeichen (für den Menschen), *janusköpfig*. Janus ist der altrömische Gott der Türen und Tore, der mit zwei Gesichtern dargestellt wurde. Wenn er über Türen und Tore, die die Verbindung zwischen der Informatik und anderen Disziplinen ermöglichten, in beide Richtungen blickte – her zum "Inneren" der Informatik, den berechenbaren Prozessen, und hin zum "Inneren des Anderen", das auf informatisch Behandelbares reduziert oder durch informatisch Konstruiertes bereichert wird –, gäbe er vielleicht ein freundliches Vorbild für die SoftwareentwicklerInnen und Theoriesuchende ab.

Aber Vorsicht mit einer semiotischen Überladung der Mythologie; Janus ist auch die Gottheit des Anfangs und des Endes. Als Informatiker beschäftigen wir uns meist mit der Symbolwelt, die Maschinen- und die Menschenwelt geraten in den Hintergrund (Flo92, Nau92). Es gibt aber auch andere Vorlieben und Blindheiten; wie das Bild ausfällt, hängt von unserer Beziehung zu den drei Welten ab. Sicher sind sie nicht gleichberechtigt: Wir produzieren Maschinen und Formalismen zu unserem Nutzen, nicht als Selbstzweck. Aber gelingen kann die Hybridisierung nur, wenn wir alle drei Welten im Blick und miteinander in Beziehung halten. Beim Umgang mit dem Computer müssen wir mit den Maschinen und Formalismen auch uns selber entwickeln.

6. Informatik im interdisziplinären Kontext

Schon auf der 1. Theoretagung 2001 in Heppenheim wurde bezweifelt, daß die Informatik eine klassische Wissenschaft mit klaren Grenzen sei. In der AG "Kulturelle Aufregung" wurden drei Thesen zum Selbstverständnis der Disziplin dagegen gestellt, in denen sich Hoffnungen, Befürchtungen und Überzeugungen mischen (Bericht in NRS01).

These 1: Informatik ist eine Wissenschaft der Moderne. Informatiker vertreten eine universell einsetzbare Technologie; damit gehen sie viel problemloser in andere Gebiete hinein als umgekehrt und helfen, sie zu modernisieren.

These 2: Informatik ist eine postmoderne Wissenschaft mit einer Flickenteppich-Identität. Sie ist nicht durch Inhalte und Methoden zu charakterisieren; man muß die Forschungsschwerpunkte untersuchen, um ein Bild zu bekommen.

These 3: Informatik ist eine postpostmoderne Wissenschaft. Sie will alle Lebensbereiche computerisieren, befördert die Globalisierung, bezieht daraus ihre wissenschaftliche Identität.

Der These 1 wurde schon in Heppenheim entgegengesetzt: Die Offenheit ist nur scheinbar. Es geht um den technischen Erfolg (das System muß laufen), nicht darum, das Fremde kennenzulernen. So kolonialisieren die Informatiker den Teil der anderen Disziplin, in dem sie Erfolg haben; der Rest ist in Gefahr, weil er kein Geld mehr bekommt. Informatiker reiten in die anderen Gebiete ein, soweit ihre Maschinen sie tragen.

Mit der Metapher von der Informatik als kultureller Entwicklung sind wir auch in Hersfeld davon ausgegangen, dass die Informatik keine klassische Wissenschaft ist (Abschnitt 1). Ebenso haben wir uns gegen These 3 als Zielvorstellung gewandt: Kulturen sind nicht global, sie leben von Kooperation mit *anderen* Kulturen, also von regionaler Verschiedenheit und Vielfalt. Träume von uniformen Weltkulturen sind faschistisch: *Eine* Kultur, Weltanschauung, Religion, Wissenschaft soll alle anderen ersetzen, notfalls mit Gewalt, weil sie als die beste angesehen wird.

Aber auch These 2 war umstritten. Die Postmoderne hat keine einheitliche Vision, ihre Positionen erscheinen daher vielen beliebig. "Kulturelle Entwicklung" heißt aber nicht Beliebigkeit. Eine Kultur lebt davon, dass es Menschen gibt, die ihr angehören, sich mit ihr identifizieren und diese Identität tragen und weitergeben. So bleibt auch die Informatik nur am Leben, wenn wir Informatiker eine Identität haben. Die speist sich nicht aus einem Kanon von Inhalten und Methoden und davon abgeleiteter Curricula. Wir brauchen eine gemeinsame Aufgabe, an der zu arbeiten uns sinnvoll erscheint und daher Freude macht und die sich von den Aufgaben anderer Disziplinen klar unterscheidet. Weil wir zum Arbeiten Methoden und andere Hilfe brauchen, ergeben sich Kanon und Curricula aus der disziplinären Arbeit, bestimmen ihre Richtung, aber ändern sich dabei, entwickeln sich weiter. Deswegen haben wir Informatik eine kulturelle Entwicklung genannt, nicht bloß eine Kultur.

Wenn wir Computer oder Programme oder computergestützte Systeme als Ziel unserer Arbeit betrachten, reduzieren wir Informatik auf Maschinenkonstruktionen oder Angewandte Mathematische Logik oder eine Kombination beider. Wenn wir den Computer als neutrales Medium betrachten und nur auf die Anwendungen schauen, machen wir Informatik zur Universalwissenschaft, die alle anderen verdrängt oder kolonialisiert. Die *Aufgabe der Informatik* liegt dazwischen: Entwicklung von Computersystemen mit dem Ziel, Bedingungen des Einsatzbereichs zu verbessern. So hieß es schon im 2. Abschnitt, für Entwicklung und Anwendung. Welche Verbesserungen sie wünschen, können nur die Anwender formulieren; und nur die Informatiker können die Systeme entwickeln. Aber nur die Informatiker können beurteilen, wie sich die gewünschten Abläufe verändern, wenn sie tatsächlich automatisch ablaufen; und nur die Anwender können sagen, ob sie mit diesen Veränderungen einverstanden sind. Konstruktion und Bewertung lassen sich nicht trennen, weder zeitlich noch personell; Informatiker und Anwender müssen bei beidem zusammenarbeiten. Andere Ingenieurwissenschaften greifen nicht so tief in menschliche Lebensbereiche ein, sie verändern direkt nur Äußerliches (s.o.); daher gilt das für sie nicht in demselben Maße. Man vergleiche dazu die Diskussion der Informatik als "Gestaltungswissenschaft" im Sichtweisenband (Rolf, Siefkes, Volpert in Coy92) und in (Rol98).

Aus der Metapher "Informatik als kulturelle Entwicklung" ist mit der oben formulierten Aufgabe eine Bestimmung der Disziplin geworden. Die Informatik entwickelt sich nicht nur *wie* eine Kultur. In die Entwicklung von Informatiksystemen fließen dauernd die Eigenheiten der Anwendungsbereiche ein, also über Hybridisierung unserer ganzen Kultur, ebenso wie wir mit unseren Systemen die ganze Kultur beeinflussen. Informatik entwickelt sich als Teil unserer Kultur und mit dieser, sie ist – gewollt oder ungewollt – prominenter Teil der kulturellen Entwicklung (SGI97a,b, Sie99, 02b; Siefkes in Fre97, NRS01, Bau01).

Das gilt nicht nur für Anwendungssysteme. Auch Standardsoftware und Hardware, mathematische Theorien und formale Methoden der Informatik dienen letztlich dazu, "bestimmte Bedingungen des Einsatzbereichs zu verbessern". In allen ihren Tätigkeiten hybridisieren Informatiker menschliche Welten (ihre eigene wie die realer und gedachter, direkter oder indirekter Benutzer) mit dem Maschinenbereich. Ihre Aufgabe ist, das weder zu verdrängen noch zu vertuschen noch auf andere abzuwälzen.

Hybridisierung ist kein Fluch, der auf der Informatik lastet, und keine Zauberkunst, die sie auszeichnet, sondern eine Herausforderung, die Informatik auf besondere Weise auf interdisziplinäre Arbeit angewiesen sein läßt. Da im Umgang mit dem Computer gegensätzliche Welten hybridisiert werden, müssen in der wissenschaftlichen Arbeit unterschiedliche Disziplinen zusammenkommen, insbesondere geistes- und sozialwissenschaftliche mit ingenieur- und naturwissenschaftlichen. Nicht nur in der Softwaretechnik verlangt das gleichzeitig Offenheit, Bescheidenheit und kritische Distanz aller beteiligten Disziplinen. Das übliche Abschotten aus Arroganz oder Resignation oder Zeitmangel kann nicht einfach durch unbedenklichen Import und Export von Wissen oder durch das Einrichten von Brückenköpfen in sonst als feindlich behandelten Bereichen ersetzt werden. Informatiker müssen menschliche Bereiche mit dem Computer hybridisieren, aber nicht andere Disziplinen mit der eigenen. Diese Balance gelingt nur in der gemeinsamen Arbeit. Wenn die Beteiligten lernen, die Erfahrungen der anderen für sich nutzbar zu machen, können sich die unterschiedlichen Menschen und ihre Welten gemeinsam entwickeln. Sicher wird es dann weniger große Projekte geben, die die Welt ändern und unsere Karriere und die Disziplin voran bringen. Aber wir können kleinere durchführen, die wirklich helfen (vgl. den Schluß des Beitrages von Siefkes in Coy92).

Wir werden die Ausbildung ändern müssen. Neben Technik und Mathematik brauchen wir Kenntnisse und Fähigkeiten aus den Geistes- und Sozialwissenschaften, nicht als Schmuck, sondern als drittes Standbein. Dafür müssen wir – Studenten wie Dozenten – keine Universalgenies werden. Wenn Dozenten herausfinden wollen, welche Rolle ihr Spezialgebiet im Prozeß der Hybridisierung spielt, müssen und können sie die engen Grenzen der Informatik überschreiten und lernen, fremde Wege zu gehen, und das den Studenten beibringen. Das ist nicht schwerer oder leichter, als im eigenen Gebiet immer tiefer zu bohren, erfordert aber ein Umdenken – auch derjenigen, die Wissenschaft weiter wie bisher betreiben. Unsere Curricula sind mit der Disziplin gewachsen und haben zu viel Material angesammelt, das nur historischen Wert hat.

Studenten müssen Programmieren lernen, grundständig, damit sie mit ihrer Arbeit zurechtkommen. In Architekt, der keine Schaufel führen kann, ist peinlich. In der Informatik gilt das umso mehr, als in der Praxis auf jeden Ausgebildeten ca. drei Leute ohne Ausbildung kommen. Wer soll denen Programmieren beibringen, wenn es die Ausgebildeten nicht können? Gerade deswegen dürfen aber in der Ausbildung Programmiersprachen nur ein Hilfsmittel sein, nicht Selbstzweck. Genauso sollen Studenten in den mathematischen Lehrveranstaltungen Formalisieren lernen, nicht Formalismen. Und

den Zugang zu den Geistes- und Sozialwissenschaften können sie nicht über lauter Zweitstudien gewinnen. Wir müssen ihnen Programmieren als Hybridisierungsaufgabe stellen. Dann lernen sie nicht nur, korrekte Programme zu schreiben, sondern auch deren Wert zu beurteilen. Damit können sie solide Fähigkeiten in der Informatik gewinnen, die durch Fähigkeiten in den anderen Bereichen gestärkt und nicht nur ergänzt werden (Sie 92 ff, Siefkes in Coy92, Fre97, Bau01).

Wir werden das Verhältnis zu unserer Arbeit ändern müssen. Wir sind gewohnt, Verantwortung für unser Tun abzulehnen, weil Programme für alles benutzt werden können: Die beste Idee kann zur schlimmsten Anwendung führen, weil wir ihre Folgen nicht absehen. Das ist aber kein Grund, sich verantwortungslos zu verhalten. Wenn wir jederzeit – mit Kollegen und "Betroffenen", mit Mitarbeitern und Freunden – über unsere Arbeit reden und wohin sie führen könnte, verstehen wir besser, was wir wirklich wollen, was wir durch Hybridisierung bezwecken und verstecken wollen. Wir müssen lernen, Verantwortung als Herausforderung und nicht als Bürde zu betrachten (Rol98, Sie92, Siefkes in Coy 92).

7. Informatik und Pädagogik

Informatik entwickelt sich im Austausch mit Nachbardisziplinen. Wir betrachten zunächst die Pädagogik, weil der Bereich in der AG am stärksten vertreten war; weitere Beispiele folgen im nächsten Abschnitt.

Behandelt man z.B. die Geschichte der Verwendung des Computers in der Schule, so sieht man anschaulich, wie zwei Disziplinen sich gegenseitig beeinflussen und ihre Identität sich (jedenfalls in Teilbereichen) dadurch wandelt. (Vgl. auch Abschnitt 9: Theorie aus der Betrachtung der Geschichte gewinnen.) Auch sonst haben sich im Bildungsbereich verfügbare Informatiksysteme und vorherrschende Lerntheorien und Lehrziele wechselseitig beeinflusst (Wil00): Lehrmaschinen – Behavioristische Lerntheorien; Werkzeuge, Intelligente tutorielle Systeme – Kognitivistische Lerntheorien; Hypermedien – Konstruktivistische Lerntheorien; Internet, Lernumgebungen – Theorien der Kommunikativen Kompetenz.

Dabei hat sich die Konzentration auf bestimmte Sichtweisen jeweils verschoben. Der Charakter des Computers wurde durch seinen Einsatz im Unterricht neu definiert, das Unterrichtsgeschehen wurde als in Teilen formalisierbarer und automatisierbarer Prozess aufgefasst. Die Stellung als eigenständige Fachdisziplinen wurde dadurch nicht in Frage gestellt, wohl aber die Sichtweise auf den Charakter der Gegenstände, mit denen es die jeweilige Disziplin zu tun hat. Sicherlich endet der Prozess nie: Beide Disziplinen begegnen sich immer wieder in der Frage: Wie können wir das Lehren und Lernen durch den Einsatz neuer technischer Möglichkeiten verbessern? Die einen verändern dabei die Technik und entwickeln sie weiter, den anderen ist die Gestaltung und Innovation von Lernprozessen zentrales Anliegen.

Informatik und Pädagogik werden hier als zwei getrennte Disziplinen gesehen, die sich beeinflussen, obwohl und weil sie so unterschiedlich sind. Pädagogen geht es um das Lernen von Menschen vermittelt durch Menschen mit Hilfe geeigneter Medien; Informatikern, die sie dabei unterstützen wollen, geht es um das Vermitteln von Informationen mit Hilfe von Computern. Damit die Faszination an und die Probleme mit der Maschine nicht die Liebe zum und die Probleme mit dem Menschen in den Hintergrund drängen, ist es notwendig, die Unterschiede zwischen beiden Disziplinen klar zu sehen und die Verbindung durch Menschen herzustellen. Diese Mittler müssen nicht in beiden Bereichen gleich kompetent sein, aber müssen sich über die Grenzen hinweg verständigen können und die ihnen fremden Vorgehensweisen und Vorlieben akzeptieren. Dann gewinnen beide Disziplinen.

In der AG wurden aber auch die gegensätzliche Ansicht vertreten, nach der Informatik und Pädagogik sachlich enger verknüpft sind als allgemein wahrgenommen. Sie verfolgen verwandte Ziele mit unterschiedlichen Mitteln; um die Zusammenarbeit zu fördern, muß man die Übereinstimmungen herausarbeiten. Diese Ansicht wurde wie folgt formuliert und diskutiert.

Eine wichtige Schnittmenge zwischen Pädagogik und Informatik besteht in der medialen Konservierung von Wissen zum Zwecke seiner Vermittlung über die Grenzen des Raums und der Zeit hinweg – zwar jeweils in sehr unterschiedlicher Weise, aber doch mit einer wesentlichen Gemeinsamkeit: Beide Fächer bedienen sich der verschrifteten sprachlichen Darstellung.

Die Informatik kennt Formalismen, die zwar vom Rechner ausführbar sind, dabei aber dennoch explizit für die Mensch-Mensch-Kommunikation entworfen wurden. Dies gilt insbesondere für semiformale Spezifikationen, eine Darstellung, die explizit zwischen Experten und Anwendern Kommunikation herstellen soll.

These: Programmiersprachen sind Sprachen in engerem Sinn: auch Menschen können damit kommunizieren, ohne dass eine Maschine je den Code ausführen müsste.

Mit Formalismen lassen sich Aspekte, wörtlich: Ansichten von Welt, z.T. durchaus adäquat beschreiben: nämlich diejenigen Teile, die den Formalismen strukturähnlich sind (oder die wir ihnen ähnlich machen wollen). Eine für Pädagogen wie Informatiker besonders wichtige Perspektive auf "Welt" ist der Umgang mit Wissen, bis hin zum derzeit so aktuellen Begriff des Wissensmanagements: Schöpfung von Wissen, seine Konservierung, seine Kommunikation, die Unterstützung seiner Aneignung.

Dafür nicht nur beschreibungsmächtige, sondern auch einfache, gering komplexe und damit hoffentlich verstehbare (semi)formale Strukturen bereit zu stellen, ist eine Kernaufgabe der Informatik. Abzuschätzen, inwiefern einzelne Beschreibungs-Frameworks einen Gegenstand *angemessen* beschreiben können, und damit Aussagen über Strukturähnlichkeiten von Formalismus und Anwendungsbereich zu treffen – dazu bedarf es freilich auf Seiten der Informatik noch einiger Lernprozesse: Es bedarf einiger erkenntnistheoretischer Offenheit und fachlicher Bescheidenheit, um mit den eigenen Tools nicht nur Welt zu verändern (und viele verschiedene lebensweltliche Erscheinungsformen möglicherweise auf das Bett des Prokrustes zu spannen), sondern auch in Kommunikation mit anderen Wissenschaftlern die Grenzbereiche (semi)formaler Beschreibungen auszuloten.

Lebenswelt aber zu beschreiben, um sie anderen Lernenden erfahrbar zu machen, sie als Kultur und Wissen zu tradieren, oder sie gar emanzipatorisch zu verändern: Dies ist eine ureigenste Aufgabe der Pädagogik. Und Gegenstandsbereiche so in einem Medium gerinnen zu lassen, dass ein Lernender in der Aneignung des Gegenstands unterstützt (aber nicht bevormundet) wird, hat sich die Didaktik vorgenommen. Pädagogik und Didaktik haben – solange sie es nicht verabsolutieren und über eine anständige Erkenntniskritik verfügen – keinen Grund, auf das kommunikative Potential zu verzichten, das (semi)formalen Darstellungen innewohnt.

Dazu bedarf es freilich auch auf Seiten der Pädagogen (der Lehrenden) und ihrer Adressaten (der Lernenden) Lernprozesse, die vor noch nicht allzulanger Zeit der Pädagogik fremd zu sein schienen: Es bedarf profunder Kenntnisse lebensweltlicher Gegenstandsbereiche *und* formaler Strukturen, um aus den (gar nicht so neuen) Potentialen der Neuen Medien Fruchtbare schöpfen zu können, ohne das Bestehende zu übersehen.

Bei der ersten Vorgehensweise versucht man also, die Zusammenarbeit durch Klärung der Unterschiede zu fördern. In beiden Disziplinen verwendete Begriffe werden auf ihren unterschiedlichen Gebrauch hin abgeklopft, um die unterschiedlichen Umgänge damit sinnvoll übertragen zu können. Bei der zweiten Vorgehensweise werden dagegen die Gemeinsamkeiten hervorgehoben, um beide Seiten zu ermutigen, sich der anderen Disziplin zu nähern.

8. Andere Beispiele für Interdisziplinarität

In anderen Disziplinen waren die Positionen geschlossener, weil sie jeweils von einer Person vertreten wurden; sie unterschieden sich voneinander aber nicht weniger krass.

Expertensysteme: Informatiker neigen aufgrund der massiven Auswirkungen, die informationsverarbeitende Systeme in unserer Gesellschaft haben, sich für viele der damit verbundenen Probleme zuständig zu erklären. Das unterschätzt jedoch die Rolle der "Anwender" dieser Programme, für die der Computer ein immer selbstverständlicheres Arbeitsinstrument wird (Pup90). Daher ist zwar der Beobachtung zuzustimmen, dass Informatik interdisziplinär angelegt ist, nicht jedoch den obigen Schlussfolgerungen bezüglich Änderung der Ausbildung, der Art der Arbeitsteilung und der Berücksichtigung der Folgen des Einsatzes von Programmen. Anstatt überall mitzumischen, könnten Informatiker sich auch auf ihre genuinen Stärken bei der Softwarekonzeption und Entwicklung konzentrieren und diese auszubauen: Sie überlassen den Fachexperten die führende Rolle bei der Erstellung und dem Einsatz von Software in ihren jeweiligen Domänen und unterstützen sie dadurch, dass sie flexible, wiederverwendbare und adaptierbare Entwicklungswerkzeuge und Kooperationsformen anbieten. Interdisziplinarität und Arbeitsteilung müssen kein Widerspruch sein. Da die Fachexperten solide informatische Grundkenntnisse benötigen, sollte die Hybridisierungsdiskussion der Ausbildung in die entsprechenden Fach- und insbesondere die zunehmenden Kombinationsstudiengänge hineingetragen werden (Puppe Positionspapier).

Informatik und Soziologie: Petrinetze wurden von Beginn an dazu konzipiert, Handlungen und Handlungskomplexe sowie Beziehungen von Sachverhalten zu modellieren. C.A. Petri wollte dies nie nur auf die Informatik bezogen wissen, wengleich die weitestgehenden Modellierungserfahrungen und Theoriefortschritte in der Informatik gemacht wurden. Es scheint daher aussichtsreich, dies in den Dienst soziologischer Theoriebildung zu stellen, zumindest dort wo der Begriff des sozialen Handelns im Vordergrund steht. Konkret können Szenarien aus dem Bereich der soziologischen

Organisationstheorien erfolgreich mit höheren Petrinetzen, d.h. Objektnetzen und insbesondere Referenznetzen, modelliert werden (Val98, Hin01, Kum02).

Dabei ist es notwendig, die Theorien der Soziologie insbesondere auf ihre operationalisierbaren Inhalte zu prüfen. Dafür müssen Informatiker und Soziologen miteinander die jeweilige Theorie Schritt für Schritt durchgehen und die Quintessenz des gerade betrachteten Aspektes durch ein Petrinetzkonstrukt darstellen. Dazu ist natürlich jeweils ein mehr oder weniger großer Abstraktionsschritt notwendig, da eine solche Operationalisierung nicht unbedingt schon vorgegeben ist. Es kann lange Diskussionen über die Adäquatheit eines Schrittes geben. Dies erinnert stark an ein Auftraggeber/Modellierer-Gespräch bei der Anwendungsprogrammierung. Das Besondere ist jedoch, dass das Ziel nicht ein Stück Software, sondern ein neuer Theorieteil (oder zumindest die Interpretation eines solchen) ist. Beide Partner ziehen ihren Vorteil aus diesem Vorgehen. Die Informatiker verstehen besser, was die jeweiligen soziologischen Theorien aussagen (zur Vorbereitung einer realen Ablaufmodellierung), während die Soziologen neue Struktureinsichten gewinnen.

Obwohl Petrinetze konzeptionell einfach sind und keine speziellen Kenntnisse (wie Programmierung) erfordern, ist zu beobachten, dass ihre Modellierung immer die Domäne der Informatiker bleibt. Die Soziologen machen große Fortschritte im Verständnis der Modelle, so dass es nach einiger Zeit möglich wird, qualitativ anspruchsvoll mit ihnen über deren Adäquatheit in Hinblick auf die originären soziologischen Theorien zu diskutieren und sogar zu streiten. Offenbar erfordert der Entwurf der Modelle eine langjährige Erfahrung, wie sie normalerweise Informatiker erwerben. Das Beispiel zeigt weiter, wie die Informatik als Methodendisziplin in anderen Bereichen in einer Weise wirksam werden kann, die weit über das gängige softwaremäßige Umsetzen von gegebenen Spezifikationen hinausgeht (Val97, Valk Positionspapier).

Informatik und Geschichte: Geschichte als Generationenfolge von immer schnelleren Rechnern oder Hierarchien von Programmiersprachen langweilt; Geschichten von Pionieren erfreuen, aber belehren nicht. In Texten von Informatikern nach kulturellen Orientierungen zu suchen, kann Studenten wie Dozenten, Wissenschaftlern wie Anwendern einen neuen Zugang zur Informatik eröffnen: Eine Ingenieurdisziplin, die nicht nur Probleme löst oder schafft, sondern Teil der Kultur ist, in der sie selber leben. Eine Wissenschaft, die ihre Verantwortung nicht leugnet oder verdrängt, sondern als Herausforderung annimmt.

In den großen Zyklen des Grundstudiums werden den Studenten Methoden, Praktiken und Theorien der Informatik als Fakten präsentiert. "Das ist die Informatik, auf diese Felsen müßt ihr bauen." Diese erdrückende Faktizität kann man aufbrechen, wenn man die Studenten die historischen Hintergründe erarbeiten läßt. Die Studenten sehen in historischen Texten, wie Fakten entstehen und vergehen, wie sie sich widersprechen oder unterstützen oder jede Beziehung verweigern. Sie lernen, nach den Gründen zu fragen und hinter die Faktizität des Gegenwärtigen zu schauen. Sie erkennen, daß die Entwicklung in die Zukunft nicht so überraschend, aber auch nicht so unausweichlich ist (Kor02, Sie02).

Der Anfang des 1. Abschnitt und der 5. Abschnitt zeigen, wie Informatiker in Zusammenarbeit mit Historikern eine Sozial- und Kulturgeschichte der Informatik in diesem Sinn schreiben können (SGI97 ff, Eul99, Sta01, Sie99 ff, Siefkes in Fre97, Bau01, NRS01). Historiker ohne Erfahrung in der Informatik können Informatik-Quellen nicht auf diese Weise auswerten. Die Art der Darstellung stößt daher bei vielen Historikern auf dasselbe Unverständnis wie bei Informatikern mit einer klassischen Vorstellung von Geschichte.

9. Theorie der Informatik

Die Teilnehmer der AG kamen mit ganz unterschiedlichen Vorstellungen, was eine oder die Theorie der Informatik sein und wozu sie dienen könne:

Zum einen: Über den Begriff "Theorie" kann man sehr verschiedene Meinungen haben. Von einem radikalen Standpunkt aus ist alles eine Theorie, was nicht unmittelbar wahrnehmbar ist, sondern aus den Wahrnehmungen als allgemein gültiges Prinzip, als Modell zur Erklärung und Prognose, destilliert wird (Lud02, Ludwig Positionspapier). Dass es mal wärmer, mal kälter ist, nehmen wir wahr. Dass es einen Rhythmus der Jahreszeiten gibt, ist eine Theorie. Diese Deutung unterscheidet sich drastisch von einer Definition, die auf die Formalisierung der Modelle abhebt. Eine Differentialgleichung für natürliche Phänomene ist natürlich eine besonders schöne Theorie, weil man damit rechnen kann, aber Brook's Law ("Adding manpower to a late project makes it even later") ist gleichwohl auch eine Theorie (mit der man leider *nicht* rechnen kann).

Zum anderen: Theorie und Praxis entwickeln sich in Wechselwirkung. Erfahrungen in der Anwendung von Informatiksystemen konsolidieren sich, steigen langsam auf in die Wissenschaft Informatik,

verbinden sich da mit vorhandenen Kenntnissen, die zu neuen Anwendungen führen, die wieder in die Praxis absinken. Theorie ist danach das Dach von Wissen und Methoden, das die Praxis behütet und von ihr getragen wird. Weniger poetisch: Theorie ist Invarianz in der Entwicklung.

Zum dritten: Theorie soll interdisziplinäre Arbeit ermöglichen. Konkret soll eine Theorie der Informatik und Pädagogik die Veränderungen erklären, die beide Disziplinen in der Kooperation durchlaufen; sie soll die unterschiedlichen Sichtweisen integrieren und so Analyse und Konstruktion von Technik *und* Lernprozessen unterstützen. Um etwas auf der Grenze (oder dem Begegnungsraum) von berechenbaren Prozessen und menschlichem Lernen behandeln zu können, liegt es nahe, sich auf Theorien (oder Disziplinen) zu beziehen, die sowohl die maschinelle als auch die menschliche Seite von Interaktion zu betrachten erlauben. Besonders in Zusammenhängen, wo es um die Gestaltung der Benutzung von Informatiksystemen geht, wird mit Bezug auf die Semiotik versucht, den Unterschied informatischer Signalprozesse und menschlicher Zeichenprozesse theoretisch "in den Griff" zu bekommen (auf den Begriff zu bringen, zu begreifen) und das Feld ihrer gemeinsamen Wirksamkeit zu beackern. Wenn man sich (mit Vorbehalten, vielleicht nur versuchsweise) auf den Gedanken einlassen kann, die Gemeinsamkeit von informatischem und didaktischem Handeln in der Konstruktion und Gestaltung von Zeichenprozessen zu sehen, liegt es doch eigentlich nahe, sich bei einer entsprechenden Fachdisziplin nach theoretischem Handwerkszeug umzuschauen und sich auf das Modell einer triadischen Zeichenrelation zu konzentrieren, die den Interpretationsvorgang als integralen Bestandteil der semiotischen Theorie definiert und eben dadurch die Unterscheidung in maschinelle Signalprozesse und menschliche Interpretationsprozesse (= vollständige Zeichenprozesse) im Rahmen *einer* Theorie behandelbar macht (Nake in Coy92, NRS01, Bau01; Wil00, Wilkens Positionspapier; Nöt02).

Zum vierten: Auf eine "theoretische Selbstbeschreibung" der Informatik können andere Disziplinen als Grundlage für interdisziplinäres Arbeiten zurückgreifen. Bei der Erstellung einer solchen Theorie könnte die Philosophie als "Mutter aller Wissenschaften" helfen, die sich seit langer Zeit bereits mit dem Problem der Abgrenzung und der Integration der verschiedenen wissenschaftlichen Disziplinen beschäftigt. Dabei sollte auch die Ethik einbezogen werden (Drews Positionspapier; Nöt02).

Schließlich: Bei der Frage nach einer Theorie der Informatik kann man sich an die oben geäußerte These halten, die die eigenständige Kompetenz der Informatiker gegenüber einer "anwendenden" Disziplin hervorheben (Val 97, Valk in NRS01). Diese Kompetenz ist in gewissen Grenzen wertneutral – ähnlich wie ein Jurist einen Fall auf der Basis von formalen Gesetzestexten behandelt und dabei von anderen Gesichtspunkten weitgehend abstrahiert. Damit die Kommunikation mit der anderen Disziplin auf anspruchsvollem Niveau zu führen ist, muss der Informatiker andererseits auch hinreichend "auf der anderen Seite des Grabens" agieren können (Val87). Eine "Theorie der Informatik" sollte auf Methodenkompetenz abheben, da sie im Verhältnis zu vielen anderen Partnerdisziplinen wirksam werden soll.

Dagegen: Wir müssen eine breitere Theorie der Informatik aus der Informatik heraus entwickeln (Coy92 und in Fre97, Siefkes in Fre97, NRS01 und in diesem Band). Die aus der Mathematik übernommene Theoretische Informatik hilft, Programme und andere formale Werkzeuge zu verstehen. Um Hybridisierung zu verstehen, müssen wir nicht nur in der Lage sein, menschliche Aktivitäten in unterschiedlichen Bereichen zu formalisieren und zu maschinisieren und die automatisierten Systeme so in menschliche Umgebungen zurück zu bringen, daß sie funktionieren. Beide Unterfangen – verharmlosend De- und Rekontextualisierung genannt – verändern die Umgebungen, insbesondere die beteiligten Menschen. Eine soziale Umgebung ist kein Text, den man mit "cut and paste" editieren kann. Auch die Theorien der Nachbardisziplinen genügen nicht für eine theoretische Grundlegung der Informatik; sie sind nicht im Zusammenhang mit der Informatik entstanden und helfen uns daher nicht, die Informatik zu verstehen. Eine Theorie der Informatik können nur Informatiker aus der Zusammenarbeit mit Mitgliedern anderer Disziplinen heraus entwickeln. Nur eine solche Theorie kann ihnen helfen, bessere Informatiksysteme zu schaffen.

Wir haben nicht versucht, zwischen diesen unterschiedlichen Positionen, die sich teils ergänzen, teils widersprechen, eine Einigung herbeizuführen. Stattdessen haben wir *Theoriearbeit* geleistet. Wir haben gefragt: Wo stehen wir mit der Informatik? Wozu arbeiten wir? Was erreichen wir? Welchen Sinn ergibt das für uns und andere? Die Antworten waren unterschiedlich, aber die Diskussion brachte uns zu einem gemeinsamen Vorgehen: Informatik als kulturelle Entwicklung zu betrachten. Und damit zu einer gemeinsamen Erkenntnis: Informatik ist nur im interdisziplinären Kontext zu betreiben und zu verstehen.

Haben wir damit den vier Sichtweisen des Bandes (Coy92) eine fünfte hinzugefügt oder gar bloß die Sichtweise "Kultur" von dort übernommen? Der Titel "Sichtweisen der Informatik" wird gern als

Eingeständnis einer Niederlage angesehen: Da sind ein paar, die mit der Informatik unzufrieden waren, aufgebrochen, um eine Theorie zu finden, die zeigt, was Informatik wirklich ist oder sein sollte. Weil sie die Theorie nicht finden konnten, haben sie sich damit begnügt, die Aussichtstürme zu beschreiben, auf die sie bei ihrer Suche gestiegen sind.

Diese Einschätzung resultiert aus einer bestimmten Vorstellung von Theorie: Die Theorie eines Gegenstandes ist eine Beschreibung, die den Gegenstand erklärt. Eine Theorie besteht also aus Aussagen *über* den Gegenstand. Der Gegenstand liegt vor uns, die Theorie steht *darüber*, wohl getrennt. Deswegen kommen Theorien von oben: für das Leben aus der Philosophie, für das Sterben aus der Religion, für das Glück aus der Astrologie. Für die Informatik aus der Mathematik.

Der Arbeitskreis von Coy hat aber nach einer Theorie der Informatik *in* der Informatik gefragt. Das kann keine informatische Theorie sein; sonst hätten wir einen Widerspruch wie beim lügenden Kreter. Also nicht draußen, nicht drinnen – wo dann? Wir haben in der AG nach dem Sinn unserer Arbeit gefragt, um zu einer Theorie der Informatik zu kommen. Die Antworten kommen aus der Arbeit und fließen wieder in sie ein; sie zeigen und bestimmen unser Selbstverständnis. Eine Theorie der Informatik gewinnen wir aus ihrer Entwicklung, um diese damit zu steuern. So wuchten wir ein Boot über eine Landenge: wir sammeln die Hölzer hinten auf und werfen sie vorn wieder drunter. An einer Theorie der Informatik zu arbeiten, heißt Informatikkultur zu entwickeln. Kulturen müssen vielfältig sein, haben wir oben gesagt. Entwicklung lebt von Vielfalt (Bro79). Daher wird die Theorie wie die Kultur nicht einheitlich sein; verschiedene Menschen werden Unterschiedliches verstehen wollen, sich dabei zu verständigen suchen, aber nicht einer Meinung sein. Einig werden sie sich nur im Ziel sein: Den Sinn ihrer Arbeit zu erfassen.

Literatur

- (Bau74) Bauer, Friedrich Ludwig: Was heißt und was ist Informatik? IBM-Nachrichten 24, 1974, S. 333-337.
- (Bau01) Bauknecht, Kurt et al. (Hrsg.): Informatik 2001. GI/OCG-Jahrestagung Wien. Österreichische Computer Gesellschaft 2001.
- (Bra73) Brauer, Wilfried: Die Informatik in der Bundesrepublik Deutschland. In GMD/ DAAD (Hg.): Studien- und Forschungsführer Informatik. Bonn 1973.
- (Brö97) Brödner, Peter: Der überlistete Odysseus. Über das zerrüttete Verhältnis von Menschen und Maschinen. Berlin: edition sigma 1997.
- (Bro79) Bronfenbrenner, Urie: The ecology of human development. Harvard University Press 1979.
- (Coy92) Coy, Wolfgang et al. (Hrsg.): Sichtweisen der Informatik. Vieweg 1992.
- (Des01) Desel, Jörg (Hrsg.): Das ist Informatik. Springer 2001.
- (Eul99) Eulenhöfer, Peter: Die formale Orientierung der Informatik. Zur mathematischen Tradition der Disziplin in der Bundesrepublik Deutschland. Dissertation TU Berlin, FB Informatik, 1999.
- (Flo92) Floyd, Christiane et al. (eds.): Software Development and Reality Construction. Springer 1992.
- (Fre97) Freksa, Christian et al. (eds.): Foundations of Computer Science – Potential, Theory, Cognition. Springer 1997.
- (Hin01) Hinck, D. et al.: Organisation etablierter Machtzentren: Modellierungen und Reanalysen zu Norbert Elias. Uni Hamburg, FB Informatik, Mitteilung 306, 2001.
- (Kor02) Korb, Joachim et al.: Geschichte als Zugang zur Informatik. Zu einem Studienreformprojekt. TU Berlin, Fakultät Elektrotechnik und Informatik, Bericht 02-15.
- (Kum02) Kummer, O.: Referenznetze. Dissertation Uni Hamburg, FB Informatik, 2002.
- (Lud02) Ludwig, Jochen: Modelle im Software Engineering. In M. Glinz et al. (Hrsg.): Modellierung 2002. Tutzing 2002, S. 7-22.
- (Nak93) Nake, Frieder (Hrsg.): Die erträgliche Leichtigkeit der Zeichen. Ästhetik Semiotik Informatik. Agis 1993.
- (NRS01) Nake, Frieder, Arno Rolf, Dirk Siefkes (Hrsg.): Informatik - Aufregung zu einer Disziplin. Tagung Heppenheim 2001. Uni Hamburg, FB Informatik, Bericht 235, 2001.
<http://tal.cs.tu-berlin.de/siefkes/Heppenheim>
- (Nau92) Naur, Peter: Computing - a Human Activity. ACM Press, Addison-Wesley 1992.
- (Nöt02) Nöth, Winfried: Semiotic Machines. Cybernetics and Human Knowing 9.1.2002, pp. 5-21.
- (Pup90) Puppe, Frank: Problemlösungsmethoden in Expertensystemen. Springer 1990.
- (Rec00) Rechenberg, Peter: Was ist Informatik? Hanser 1991, 3. Aufl. 2000.
- (Rol98) Rolf, Arno: Grundlagen der Organisations- und Wirtschaftsinformatik. Springer 1998.
- (Sch97) Schelhowe, Heidi: Das Medium aus der Maschine. Zur Metamorphose des Computers. Campus 1997.
- (SGI97a) Eulenhöfer, Peter et al.: Informatics as Cultural Development. TU Berlin, FB Informatik, Bericht 97-2.

- (SGI97b) - " - : Die Konstruktion von Hybridobjekten als Orientierungsmuster in der Informatik. TU Berlin, FB Informatik, Bericht 97-23.
- (SGI98) Siefkes, Dirk et al. (Hrsg.): Sozialgeschichte der Informatik. Kulturelle Praktiken und Orientierungen. Deutscher Universitätsverlag 1998.
- (SGI99) - " - : Pioniere der Informatik. Ihre Lebensgeschichte im Interview. Interviews mit F.L. Bauer, C. Floyd, J. Weizenbaum, N. Wirth, H. Zemanek. Springer 1999.
- (Sie92) Siefkes, Dirk: Formale Methoden und kleine Systeme. Lernen, leben und arbeiten in formalen Umgebungen. Vieweg 1992.
- (Sie93) - " - : Evolutionäre Modelle in der Informatik. TU Berlin, FB Informatik, Bericht 93-15.
- (Sie99) - " - : Die Rolle von Schemata in der Informatik als kultureller Entwicklung. TU Berlin, FB Informatik, Bericht 99-6.
- (Sie02a) - " - : Sozialgeschichte und kulturelle Theorie der Informatik. TU Berlin, Fak. Elektrotechnik & Informatik, Bericht 02-16.
- (Sie02b) - " - : Hybridization in Computer Science. TU Berlin, Fak. Elektrotechnik & Informatik, Bericht 02-17.
- (Sta01) Stach, Heike: Zwischen Organismus und Notation. Zur kulturellen Konstruktion des Computer-Programms. Deutscher Universitätsverlag 2001.
- (Val87) Valk, Rüdiger: Der Computer als Herausforderung an die menschliche Rationalität. Informatik-Spektrum 10, 1987, S. 57-66.
- (Val97) - " - : Die Informatik zwischen Formal- und Humanwissenschaften. Informatik-Spektrum 20, 1997, S. 95-100.
- (Val98) - " - : Petri nets as token objects: An Introduction to elementary object nets. In J. Desel et al. (eds.): Application and Theory of Petri Nets. LNCS 32, Springer 1998.
- (Wil96) Wilhelm, Rudolf: Informatik. Beck 1996.
- (Wil00) Wilkens, Ulrike: Das allmähliche Verschwinden der informationstechnischen Grundbildung. Zum Verhältnis von Informatik und Allgemeinbildung. Shaker 2000.
- (Zem71) Zemanek, Heinz: Was ist Informatik? Elektronische Rechenanlagen 13, 1971, S. 157-161.

Liste der TeilnehmerInnen

Corinna Bath
Universität Bremen
bath@informatik.uni-bremen.de

Peter Bittner
Humboldt-Universität Berlin
bittner@informatik.hu-berlin.de

Peter Brödner
IAT Gelsenkirchen
broedner@iatge.de

Johannes Busse
Universität Heidelberg
busse@ews.uni-heidelberg.de

Jörg Cassens
Universität Trondheim
jorg.cassens@idi.ntnu.no

Wolfgang Coy
Humboldt-Universität Berlin
coy@informatik.hu-berlin.de

Cecile K.M. Crutzen
Open Universiteit Heerlen
cecile.crutzen@ou.nl

Christian Dahme
Humboldt-Universität Berlin
dahme@informatik.hu-berlin.de

Paul Drews
Drews Consulting Hamburg
paul.drews@drewsconsulting.de

Martin Fischer
GIPA Berlin
m.fischer@gipa.de

Hans-Werner Hein
Verlässliche IT-Systeme Köln
hwh@acm.org

Eva Hornecker
Universität Bremen
eva@artec.uni-bremen.de

Barbara Kleinen
Universität Lübeck
kleinen@informatik.mu-luebeck.de

Hans-Jörg Krewowski
Universität Bremen
kreo@informatik.uni-bremen.de

Jochen Ludewig
Universität Stuttgart
ludewig@informatik.uni-stuttgart.de

Bernd Pape
Universität Hamburg
pape@informatik.uni-hamburg.de

Jörg-Martin Pflüger
Technische Universität Wien
jpflueger@igw.tuwien.ac.at

Christian Pothmann
Technische Universität Berlin
Christian.Pothmann@web.de

Frank Puppe
Universität Würzburg
puppe@informatik.uni-wuerzburg.de

Arno Rolf
Universität Hamburg
rolf@informatik.uni-hamburg.de

Britta Schinzel
Universität Freiburg
schinzel@modell.iig.uni-freiburg.de

Jochen Schneider
Universität Magdeburg
Jochen.Schneider43@epost.de

Peter Schuck
Spardat Wien
schuckp@spardat.at

Kai Seim
T Systems Darmstadt
k.seim@t-systems.com

Dirk Siefkes
Technische Universität Berlin
siefkes@cs.tu-berlin.de

Ralf E. Streibl
Universität Bremen
res@informatik.uni-bremen.de

Rüdiger Valk
Universität Hamburg
valk@informatik.uni-hamburg.de

Jutta Weber
Technische Universität Braunschweig
youddl@uni-bremen.de

Karsten Wendland
Technische Universität Darmstadt
wendland@zit.tu-darmstadt.de

Jörg Werda
Deutsche Lufthansa Hamburg
joerg.werda@dlh.de

Ulrike Wilkens
Hochschule Bremen
wilkens@informatik.hs-bremen.de

Gerhard Wohland
Diebold Deutschland
gwohland@debis.com