Modellierung – ein Handgriff zur Wirklichkeit

Zur sozialen Konstruktion und Wirksamkeit von Informatik-Modellen

Christiane Floyd, Ralf Klischewski

Universität Hamburg, Fachbereich Informatik, AB Softwaretechnik

Vogt-Kölln-Straße 30, 22527 Hamburg

floyd/klischewski@informatik.uni-hamburg.de

Abstract: Obwohl Informatik-Modelle selbst formal sind und technisch realisiert werden, werden sie nur bedeutsam durch soziale Prozesse der Entwicklung und Aneignung, wobei die subjektiven Perspektiven der Beteiligten entscheidenden Einfluß haben. Entwicklung und Nutzung von Informatik-Modellen sollten daher ihre sozialen Bedingtheit transparent machen und ihre sinnvolle Interpretation im Einsatzkontext gewährleisten.

A program is a model of a model within a theory of a model of an abstraction of a portion of the world or of some universe of discourse. (Lehman 1980)

1 Einführung

In der Informatik kommt der Modellierung eine zentrale Stellung zu. Das hier gewählte Motto macht deutlich, wie vielschichtig Modellierung mit der Entwicklung von Informatik-Systemen verbunden ist. Offenkundig geht es um verschiedene Arten und Ebenen der Modellierung, ja sogar um verschiedene Modellbegriffe, die der algorithmischen Realisierung, der formalen Theorie und dem Verständnis des Gegenstandsbereichs zuzuordnen sind. Um das Ineinandergreifen der Modelle auf diesen Ebenen zu benennen, sprechen wir von *Informatik-Modellen*. Informatik-Modelle bestimmen zunächst die Sichtweise des Gegenstandsbereichs. Somit sind sie "Fenster zur Wirklichkeit"[1] im Sinne von Perspektiven der Wahrnehmung. Im Hinblick auf ihre intendierte Wirksamkeit beim Systemeinsatz ist allerdings eher die Metapher "Handgriff zur Wirklichkeit" angemessen, die sich darauf bezieht, wie Informatik-Modelle entwickelt und verwendet werden.

Ausgangsfragen im folgenden Beitrag sind: Welche Art von Modellen sind für die Informatik bedeutsam? Wie entstehen Modelle in kommunikativen Erkenntnis- und Aushandlungsprozessen? Wie werden Informatik-Modelle realitätswirksam? Die hier vertretene Position ist: Obwohl Informatik-Modelle selbst formal sind und technisch realisiert werden, werden sie nur bedeutsam durch soziale Prozesse der Entwicklung und Aneignung, wobei die subjektiven Perspektiven der Beteiligten entscheidenden Einfluß haben. Entwicklung und Nutzung von Informatik-Modellen sollten daher ihre sozialen Bedingtheit transparent machen und ihre sinnvolle Interpretation im Einsatzkontext gewährleisten.

Zunächst werden die Besonderheiten der Modellierung in der Informatik charakterisiert, um dann auf die Prozesse der Modellbildung und -verwendung genauer einzugehen. Darauf aufbauend wird die hier vertretene Position erläutert.

---

[1] Vgl. "Organisationstheorie als Fenster zur Wirklichkeit" (Wolff u.a. 1997)

2 Informatik-Modelle

Wenn wir das eingangs als Motto gewählte Zitat von Lehman anschauen, so zeigt sich, daß dabei verschiedene Ebenen der informatischen Modellierung unterschieden werden:

- das *Anwendungsmodell* (Gegenstandsbereich),
- das *formalen Model* (Spezifikation),
- das *Berechnungsmodell*(Programm).

Dabei kommen verschiedene Modellbegriffe zum Tragen: Beim Anwendungsmodell geht es um idealisierte oder konkrete Darstellung des Gegenstandes im Sinne der Wissenschaftstheorie: „Ein Modell eines Gegenstandes oder Verfahrens bekommt Modellcharakter gewöhnlich aufgrund einer Analogie-Beziehung zwischen der Struktur des Modells und der Struktur, die das Modell repräsentiert." (Hügli/Lübcke 1991, 395). Beim formalen Modell geht es um den logischen Modellbegriff als Zuordnung von Bedeutung zu Satz- oder Formelmengen. Das Berechnungsmodell schließlich wird durch eine Menge von zusammenwirkenden berechenbaren Funktionen definiert.

Die bei der Modellierung maßgebliche Vorgehensweise läßt sich durch verschiedene Aktivitäten charakterisieren, die jedoch keine zeitlich trennbaren Schritte darstellen:

- *Informatisieren*, d.h. materielle oder ideelle Gegenstände als Informationsträger zu betrachten, aus deren Aspekten die für die Operationen relevanten Daten ermittelt werden können;
- *Diskretisieren*, d.h. ausgewählte Aspekte der Gegenstände durch diskrete Wertebereiche zu charakterisieren, Operationen auf diesen Wertebereichen zu definieren sowie Zustände, die durch Ausführung der Operationen transformiert werden;
- *Systemisieren*, d.h. ein Informatiksystem aus zu modellierenden Elementen, Beziehungen und Operationen sowie dessen Wechselwirkung mit dem (häufig ebenfalls als System betrachteten) Kontext zu konzipieren.

Mit informatischer Modellierung sind somit mehrere Fragen verbunden: Wie verhält sich das Anwendungsmodell zum Gegenstandsbereich? Wie können die einzelnen Ebenen informatischer Modellierung ineinander übergeführt werden? Wie verhält sich das Berechnungsmodell zur technischen bzw. sozialen Wirklichkeit des Einsatzkontextes?

Mit der Problematik von Informatik-Modellen haben sich bereits mehrere Autoren auseinandergesetzt. Dabei geht es schwerpunktmäßig um den Stellenwert eines formalen Modells. Colburn et al. (1993) haben dazu eine kontroverse Diskussion unter Rückgriff auf Originalbeiträge zusammengefaßt. Hier werden sowohl die in der Informatik klassischen Argumente für Formalisierung vorgestellt, als auch die Grenzen formaler Modelle ausgelotet. So befaßt sich Naur (1993a und 1993b) mit der Verankerung von formalen Modellen im informellen, intuitiven Denken und mit dem Bruch zwischen verschiedenen Modellebenen in der Informatik. De Millo et al. (1993) zeigen, daß selbst beim mathematischen Beweisen soziale Prozesse maßgeblich sind.

Dabei bleibt jedoch das Anwendungsmodell weitgehend ausgeklammert, obwohl gerade das als Referenz nach Möglichkeit über den gesamten Entwicklungsverlauf gültig sein soll. Veränderungen von zugrundegelegten Modellen führen in der Regel zu Inkonsistenzen und Qualitätsverlust, in jedem Fall aber zu höherem Arbeits- und Kostenaufwand. Denn sie beeinflussen das Berechnungsmodell, das im Informatik-System verkörpert und für den intendierten Einsatz maßgeblich ist.

Die für das Anwendungsmodell erforderliche Identifikation von Attributen, Operationen, Zuständen und zugehörigen Transformationen ist nur dann – sowohl erkenntnistheoretisch als auch in der Praxis – vergleichsweise unproblematisch, wenn der Gegenstandsbereich abgeschlossen und überschaubar ist. Dies ist nach Lehman (1980) nicht der Fall bei sogenannten *eingebetteten Programmen*, deren Gegenstandsbereich ein Ausschnitt aus technischer oder sozialer Wirklichkeit ist (z. B. maschinelle Verfahren, Arbeitsabläufe usw.) und die als Informatik-Systeme in der realen Welt zum Einsatz kommen.

Bei technisch eingebetteten Informatik-Systemen sind zwar im Prinzip die Eigenschaften des zugrundeliegenden technischen Systems festgelegt, doch ist die Komplexität ihrer Wechselwirkungen enorm. Dazu kommt der Übergang von einer kontinuierlichen zu einer diskreten Modellierung und

letztlich muß das autooperational wirksame Informatik-System mit dem technischen System in jeder Einsatz-Situation - auch im Fehlerfall - zuverlässig zusammenwirken.

Bei sozial eingebetteten Informatik-Systemen, wo die Abstraktion von der Wirklichkeit nicht unmittelbar einsichtig ist, werden in Wissenschaft und Praxis eine Reihe von – aus erkenntnistheoretischer Sicht keineswegs selbstverständlichen – Annahmen bzw. Vorurteilen zugrundegelegt, d.h. das System wird in einem komplexen sozialen Prozeß abgegrenzt und charakterisiert (vgl. Klischewski 1996). Um so wichtiger ist es daher, sich mit mit der sozialen Bedingtheit von Modellbildung und -nutzung auseinanderzusetzen.

3 Modellierung als soziale Konstruktion

Den sozialen Prozeß der Modellbildung haben ebenfalls schon mehrere Autoren und Autorinnen in den Blick genommen. Nygaard (1986) hat *Perspektiven* als konstituierend für die Bildung von Modellen hervorgehoben. Perspektiven können dabei für individuelle oder kollektive Sichten von Beteiligten über den Gegenstandsbereich stehen (z. B. die Sicht eines Benutzers oder die Position des Betriebsrats), aber auch durch die Modellierung selbst zustandekommen (z.B. die "objektorientierte" Sicht).

Den Einfluß von Methoden auf den sozialen Prozeß hat in besonderer Weise Jayaratna (1994) untersucht, der einen Rahmen vorstellt, um den sozialen Charakter von Methoden zu evaluieren. Verwandte Auffassungen werden von Andelfinger (1997) vertreten, der die Anforderungsermittlung als diskursiven Prozeß beleuchtet.

Klein/Lyytinen (1992) analysieren die metatheoretischen Grundannahmen der vorherrschenden Modellierungspraxis. Sie beziehen sich dabei auf die Datenmodellierung, u.E. betreffen die angesprochenen Aspekte den gesamten Bereich der Modellierung in der Informatik. Die zentralen Fragen sind:

- Was wird modelliert? *(Ontologische Frage)*
- Wie gut ist das Ergebnis [der Modellierung] dargestellt? *(Linguistische Frage)*
- Warum ist das Modell gültig? *(Epistemologische Frage)*
- Welches ist der soziale Kontext der Modellierung? *(Soziologische Frage)*

Zur Frage nach dem sozialen Kontext beispielweise identifizieren die Autoren drei gängige Grundannahmen: 1. Prozesse in Organisationen sind hauptsächlich an deren Stabilität und Ordnung orientiert; 2. Datenmodelle dienen der Verwirklichung organisatorischer Ziele, indem sie zur Verbesserung von Effizienz und Effektivität beitragen; 3. Strategien und Verfahrensweisen in Organisationen sind konsistent und wohldefiniert, die Verantwortlichen wissen, was sie wollen, und können dies auch anderen vermitteln. Dabei werden insbesondere Datenmodelle[2] (trotz ihrer Kontrollpotentiale) als betriebspolitisch neutral angesehen, d.h. es wird nicht vermutet, daß es sich um akteursspezifische Sichtweisen handelt, deren Implementation auch die Durchsetzung bestimmter Interessen beinhaltet.

Die o.g. metatheoretischen Fragen zur Modellierung beantworten Klein/Lyytinen selbst kontrastiv zur vorherrschenden Sicht der Informatik:

1. Modelle sind nicht als Realitätsabbilder zu verstehen, sondern sie rekonstruieren (teilweise) die Sprache, mit der Fachleute Sinnzusammenhänge im Anwendungskontext beschreiben und kommunikativ vermitteln.
2. Modelle tragen zur Aufrechterhaltung einer sozial konstruierten Realität bei und leiten Handlungen an, indem sie Wahrnehmung, Begründungs zusammenhänge usw. kanalisieren.
3. Modelle sind immer abhängig von Vorurteilen, die auf dem Vorverständnis basieren, mit dem die Modelle entwickelt wurden. Diese Vorurteile lassen sich durch reflektiven Dialog transparent machen. Datenmodelle führen verschiedene Bedeutungs horizonte zusammen.
4. Modellierung (re-)definiert den institutionellen Bezugsrahmen für Handlungsorientierung und Entscheidungsfindung. Damit ist die Modellierung unvermeidlich eine (betriebs-)politische Aktivität, die die Interessen der beteiligten Akteure berührt.

---

[2] Bei Ablaufmodellen (z.B. Workflow, Software Process Modelling) wird eher vermutet, daß mit deren Implementation betriebspolitische Interessen verknüpft sind (z.B. hinsichtlich Arbeitsautonomie, Benutzerpartizipation).

Analytisch betrachtet sind u.E. damit die wesentlichen Dimensionen von Modellierung als soziale Konstruktion im Kontext der Informatik benannt. Allerdings, was bedeutet dies für das konkrete Vorgehen im Systemdesign? Wenn man z.B. Informatik-Modelle als eine Menge von Gesetzen zu betrachtet und Modellierung als die Aktivität, die sie entwickelt und in Kraft setzt, so stellen sich folgende Fragen (vgl. Klischewski 1996):

- Wie ist Modellierung (Gesetzbildung) am besten als sozialer Prozeß unter gegebenen Rahmenbedingungen zu organisieren?
- Wie können dabei demokratische Ziele mit faktischen Herrschafts verhältnissen vermittelt werden?
- Wie läßt sich ein geeigneter Kompromiß zwischen dauerhafter Verbindlichkeit und Anpassung an sich permanent verändernde Bedingungen finden?

Floyd (1992) versteht Softwareentwicklung als Realitätskonstruktion. Dabei treffen die Beteiligten Unterscheidungen und Entscheidungen, um mit der Vielfalt der Möglichkeiten umzugehen. So entfalten sie einen Design-Raum, in dem die Anwendungswelt, die formale Methodenwelt und technische Realisierungswelt verknüpft werden und technischer Entwurf sowie soziale Gestaltung aufgehoben sind. Für gemeinschaftlich gelingende Gestaltung ist das dialogische Prinzip maßgeblich, indem eine Vielfalt individueller Perspektiven aufgezeigt und gekreuzt werden.

Schon Bråten (1973) hat die *Modellmacht* bei der Modellbildung ins Spiel gebracht. Wird ein Akteur befähigt, ein Modell über einen Gegenstandsbereich auszuarbeiten, so wird er zum modellstarken Handelnden und besitzt schließlich ein Modellmonopol. Andere Beteiligte sind zunächst modellschwach. Je mehr sie versuchen, ihre Sichtweisen in das bereits erarbeitete Modell einbringen, desto mehr stützen sie den Modellstarken. Sie können das Modellmonopol nur auflösen, indem sie ein eigenständiges Modell entwickeln. Um diese Problematik zu umgehen, muß von vornherein ein symmetrischer Dialog zwischen den Beteiligten mit unterschiedlichen Perspektiven zugelassen werden.

Diesen Ansatz hat J. Pasch (1994) auf die Softwareentwicklung bezogen, indem er die Rolle des Modellmonopols während des Entwurfs analysiert: Modellmacht kommt demjenigen zu, der im Dialog als erster ein Modell präsentiert und somit alle anderen veranlaßt, sich damit auseinanderzusetzen. Um die gleichberechtigte Bildung eines gemeinsamen Modells durch intersubjektives Kreuzen von Perspektiven zu fördern, empfiehlt Pasch die Anwendung von Moderationstechniken in Entwurfssitzungen.

Ansätze, um den Einfluß subjektiver Perspektiven bzw. die *soziale* Konstruktion bei der Modellierung auch in der Praxis zu verdeutlichen, sind also bereits vorhanden. Dies hat auch zu Qualitätsverbesserungen geführt, z.B. durch gezielte diskursive Einbeziehung von Anwenderperspektiven. Doch zwischen Entstehung und Wirksamkeit von Informatik-Modellen vergeht in der Regel einige Zeit, und (als Beispiel) die schließlich von der Modellnutzung Betroffenen sind häufig nicht an der Modellierung beteiligt gewesen – die Kluft zwischen Entwicklung und Nutzung von Modellen läßt sich durch Dialog allein nicht überbrücken.

4 Modell und Wirksamkeit

Informatik-Modelle werden mit der Absicht entwickelt, in der Realität wirksam zu werden. Doch gibt es weder die Gewähr, daß die inkorporierten Modelle der Einsatz-Situation angemessen sind, noch daß die Modelle beim Einsatz so verstanden bzw. genutzt werden (können), wie es bei der Entwicklung antizipiert wurde. Genau genommen sind sie auf zwei Ebenen wirksam:

- *mental:* Modelle vergegenständlichen eine bestimmte Abstraktion bzw. konstruieren eine virtuelle Realität (z.B. elektronischer Schreibtisch). Die beteiligten Akteure müssen sich mit diesen Modellen auseinandersetzen, sie auf ihre Art und Weise aneignen und in ihrem Handeln interpretieren.
- *autooperational:* Als selbstablaufende Modelle haben Computerprogramme direkte Auswirkungen auf die Realität und verändern die Bedingungen des Handelns in sozialen Kontexten (z.B. technischen Systemen oder Organisationen). Sie eröffnen bzw. schließen Wahlmöglichkeiten und sind mit Handlungsräumen für die beteiligten Akteure verbunden.

Auf sprachlicher Ebene hat sich u. a. Dittrich (1997) mit der Interpretatierbarkeit von Informatik-Modellen befaßt und Hinweise gegeben, wie Softwaresysteme für Interpretation gestaltet werden können.

Hierin liegt u.E. eine Notwendigkeit für ein eigenständiges Forschungsprogramm: Wie läßt sich die Interpretierbarkeit von Informatik-Modellen unterstützen – und zwar bezogen sowohl auf ihre mentale als auch auf ihre autooperationale Wirksamkeit? Auf diesem Weg sollten aktuelle Praxisprobleme den Ausgangspunkt bilden, gleichzeitig sind aber auch die prinzipiellen Grenzen im Verhältnis von Informatik-Modellen und Wirklichkeit von Anbeginn zu verdeut lichen.

In der Praxis wird immer sichtbarer, daß insbesondere große Informatik-Systeme Strukturen in Organisationen verfestigen und daß die notwendige organisatorische Weiterentwicklung schon bei der Modellierung nicht beachtet wird bzw. werden kann.[3] In der Regel sind tayloristische Vorstellungen (insbesondere die arbeitsorganisatorische Trennung von Planung und Ausführung) Ausgangspunkt der Modellierung. Die Diskussion zu CSCW und Organisationsentwicklung zeigt zwar, daß es zu dieser Herangehensweise durchaus Alternativen gibt. Aber das grundsätzliche Dilemma bleibt: Soll ein Informatik-System als Infrastruktur einen Organsationszusammenhang als Ganzes unterstützen, sind Annahmen über die Organisation – insbesondere hinsichtlich Eigenschaften, die als stabil angesehen werden – unverzichtbar.

Unvermeidlich ist ferner, daß zwischen der Modellierungssituation und der Einsatzsituation eine unüberwindbare Kluft verbleibt: Zeit vergeht, der Gegenstandsbereich verändert sich, neue Bedingungen entstehen usw. Dies verweist auf fundamentale pragmatische Grenzen beim Einsatz von Informatik-Systemen, die nach Smith (1993) nicht nur aus den formalen Grundlagen der Berechenbarkeit sondern aus den verantwortbaren Bedingungen des Zusammenwirkens von Menschen und Informatik-Systemen im Einsatz resultieren.

5 Modelle als Werkzeuge im sozialen Prozeß

Theorie und Praxis der Informatik lassen den Schluß zu, daß die bisherige Modellbildung und -verwendung hauptsächlich auf die Einbettung von Informationstechnik in kontroll- bzw. hierarchieorientierte Zusammenhänge mit voraussehbarer Entwicklung bezogen wurde und für die Einbettung in dynamische, autonome bzw. selbstorgani sierende (soziale) Zusammenh änge ungeeignet ist.

Die soziale Bedingtheit von Modellierung wird innerhalb der Informatik zunehmend thematisiert. Den darauf aufbauenden Ansätzen ist gemeinsam, daß sie Verlauf und Ergebnis der Modellbildung nicht vollständig vorplanen, sondern den Prozeß immer wieder für den Einfluß der Erfahrung von konkreten sozialen Situationen öffnen wollen, die mit der Modellbildung in Zusammenhang stehen. Der Unterschied zu den herkömmlichen (ingenieurwissenschaftlich geprägten) Sichtweisen liegt u.a. darin, daß soziale Subjekte als Akteure der Modellierung identifiziert werden und dabei keine Kontrolle, sondern eine methodische Unterstützung des Modell bildungsprozesses angestrebt wird.

Aus dieser Sicht lassen sich als Anforderungen an Modellierung im Rahmen der Informatik benennen, daß

- als Ausgangspunkt der Modellierung nicht die „objektive Welt“ gesehen, sondern die Perspektivität der beteiligten Akteure anerkannt wird,
- Modellbildung als sozialen Prozeß verstanden wird, in dem es gilt, die vielfältigen Einflüsse und Rahmenbedingungen der Entwicklungssituation zu reflektieren und eine gestaltungsförderliche Kommunikationskultur herauszubilden,
- der Verlauf der sozial bedingten Gestaltungsentscheidungen im Rahmen der Modellierung transparent bleibt, um die spätere situative Aneignung von Modellen (gerade auch für die Fortsetzung von technischen und sozialen Gestaltungsprozessen) zu erleichtern, und
- der Einsatz von Informatik-Modellen so gestaltet wird, daß in der Situation Raum für verantwortliches Handeln und Veränderung verbleibt.

---

[3] „Die Systeme sind mit der Betriebsorganisation verwachsen, wie die Lianen mit dem Urwald. Wie alle Technik, kann Datenverarbeitung nicht auf überraschende Anforderungen angepaßt werden. Wenn die Organisation auf unerwartete grundlegende Art gändert werden muß, wirkt auch die flexibelste Datenverarbeitung ‘konservativ’.“ (Wohland 1994, 25) Diese Analyse bezieht sich auf Anforderungen an Informatik-Systeme im Kontext von ‘Lean Management’, ‘Lernende Organisation’ und anderen Bemühungen, die Wettbewerbsfähigkeit von Unternehmen durch organisatorsiche Flexibilität zu stärken (als Übersicht vgl. GMD 1994).

Zusammengefaßt: Informatik-Modelle stehen nicht als formale Gebilde für sich genommen, sondern werden in den von uns getragenen sozialen Prozessen konstituiert. Informatik-Modelle sind so zu gestalten, daß sie die sozialen Prozesse, in den sie wirksam werden, als Werzeuge („Handgriffe“) unterstützen und nicht behindern. Was dies für die Kunst der Modellierung bedeutet, ist noch zu erforschen und zu diskutieren.

5 Literatur

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

Bråten, S. (1973): Model Monopoly and Communication. Systems Theoretical Notes on Democratization. In: Acta Sociologica, Vol. 16 (2), S. 98-107

Colburn, T. R., Fetzer, J. H., Rankin, T. L. (Hg.) (1993): Program Verification. Fundamental Issues in Computer Science. Dordrecht: Kluwer Academic Publishers

De Millo, R. A., Lipton, R. J., Perlis, A. J. (1993): Social Processes and Proofs of Theorems and Programs. In: Colburn u.a. 1993, S. 297-320

Dittrich, Y. (1997): Computeranwendungen und sprachlicher Kontext. Zu den Wechselwirkungen zwischen normaler und formaler Sprache bei Einsatz und Entwicklung von Software. Frankfurt a. M.: Peter Lang Verlag

Floyd, C. (1992): Software Development as Reality Construction. In: Floyd, C., Züllighoven, H., Budde, R., Keil-Slawik, R. (Hg.): Software Development and Reality Construction. Berlin: Springer, S. 86-100

Floyd, C. (1997): Autooperationale Form und situiertes Handeln. In: Hubig, Ch. (Hg.): Cognitio Humana – Dynamik des Wissens und der Werte. Akademie Verlag, S. 237-252

GMD (Hg.) (1994): Die Revolution in der Unternehmenskultur – Herausforderung für die Informationstechnik. In: Der GMD-Spiegel 3/1994, S. 12-46

Jayaratna, N. (1994): Understanding an Evaluating Methodologies. London: McGraw-Hill

Klein, H., Lyytinen, K. (1992): Towards a New Understanding of Data Modelling. In: Floyd, C., Züllighoven, H., Budde, R., Keil-Slawik, R. (Hg.): Software Development and Reality Construction. Berlin: Springer, S. 203-219

Klischewski, R. (1996): Anarchie – ein Leitbild für die Informatik. Von den Grundlagen der Beherrschbarkeit zur selbstbestimmten Systementwicklung. Frankfurt a. M.: Peter Lang Verlag

Lehmann, M. M. (1980): Programs, Life Cycles, and Laws of Software Evolution. In: Proceedings of the IEEE, Vol. 68, No. 9 (Sept. 1980), S. 1060-1076

Naur, P. (1993a): Formalization in Program Development. In: Colburn u.a. 1993, S. 191-212

Naur, P. (1993b): The Place of Strictly Defined Notation in Human Insight. In: Colburn u.a. 1993, S. 261-274

Nygaard, K. (1986): Program Development as a Social Activity. In: Kugler, H.-J. (Hg.): Information Processing ‘86 (IFIP). Amsterdam: North-Holland, S. 189-198

Pasch, J. (1994): Softwareentwicklung im Team. Mehr Qualität durch das dialogische Prinzip bei der Projektarbeit. Berlin: Springer

Smith, B. C. (1993): Limits of Correctness in Computers. In: Colburn u.a. 1993, S. 275-296

Wohland, G. (1994): Jenseits von Taylor – Irritation als Methode. In: GMD 1994, S. 22-26

Wolff, B., Fuchs-Kittowski, K., Klischewski, R., Möller, A., Rolf, A. (1997): Organisationstheorie als Fenster zur Wirklichkeit. Vortrag auf der Tagung “Wirtschaftsinformatik und Wissenschaftstheorie”, Universität Münster, 10. Oktober 1997 (schriftliche Fassung in Vorbereitung)