

Informatiksystem und Dekonstruktion als didaktische Kategorien - Theoretische Aspekte und unterrichtspraktische Implikationen einer systemorientierten Didaktik der Informatik

Johann S. Magenheim
Universität Paderborn
Didaktik der Informatik

Abstract

Der Begriff 'Informatiksystem' wird in der Informatik sowohl in fachwissenschaftlichen als auch in didaktischen Diskussionszusammenhängen verwendet. Hierbei bleiben vor allem im didaktischen Diskurs, die theoretischen Dimensionen des Begriffs oft unklar.

Der Beitrag versucht daher, den Begriff des Informatiksystems als sozio-technischem Handlungssystem hinsichtlich seiner didaktischen Relevanz zu analysieren. Der aus diesen Überlegungen resultierende Begriff eines Informatiksystems wird zunächst daraufhin untersucht, welche seiner Aspekte und ihm impliziten fachwissenschaftlichen Methoden in einem Informatikunterricht mit allgemeinbildendem Anspruch thematisiert werden sollten. Dies geschieht vor allem anhand des für die systemorientierte Didaktik zentralen wissenschaftstheoretischen Konzepts der Dekonstruktion. Als Resultat des Beitrags werden die Konturen eines systemorientierten fachdidaktischen Zugangs zum Informatikunterricht deutlich, die zugleich Aufgabenfelder für künftige fachdidaktische Forschung beschreiben.

1. Informatiksysteme in der didaktischen Diskussion

In zahlreichen Grundsatzpapieren, Lehrplänen und didaktischen Konzeptionen wird mittlerweile der Begriff 'Informatiksystem' verwendet, um Gegenstandsbereiche, Zielsetzungen und auch die Methodik des Informatikunterrichts herzuleiten und zu begründen. Der wissenschaftstheoretische Hintergrund und die Reichweite des Begriffs bleiben aber oft unklar.

So wird in der GI-Empfehlung zum Informatikunterricht der Begriff Informatiksystem im Zusammenhang mit der Vernetzung und den technisch-kommunikativen Fähigkeiten von Computersystemen verwendet: *"Die Computertechnik wird durch den Aufbau vernetzter informationsverarbeitender Systeme in einen medialen und gesellschaftlichen Kontext eingebunden. Diese Vernetzung wird im Informatikunterricht unter der Sichtweise Informatiksysteme, Gesellschaft und Umwelt aufgezeigt."* (GI 1993, S. V) Thematisiert werden sollen im Informatikunterricht unter der Perspektive des Informatiksystems vor allem Fragen der Anwendungen von Informatiksystemen und deren Auswirkungen auf Lebens- und Arbeitswelt, Politik und Umwelt. Die Genese von Informatiksystemen durch Modellierung und Softwareentwicklung, die damit verbundenen sozialen Handlungs- und Entscheidungsprozesse, welche die technische Basis für gesellschaftlichen Auswirkungen darstellen, die informatischen Konzepte, die sich in Software materialisieren, werden vom Begriff des Informatiksystems abgekoppelt und isoliert z.B. unter der Perspektive *'Modellieren mit dem Computer'* (ebd.) betrachtet. Es fehlt ein konsistenter theoretischer Rahmen zur Analyse und Thematisierung von Informatiksystemen, der Zusammenhänge von Systemgenese, Systemkomponenten, impliziten informatischen Prinzipien, sozialen Handlungen im Kontext des Informatiksystems und deren Bewertung aufzeigt.

Als Repräsentant für die Verwendung des Begriffs Informatiksystem unter den Didaktikern sei Baumann (1996) erwähnt. Er bemüht sich, in seiner Abhandlung den Systembegriff zu klären und den von ihm verwendeten Begriff Informatiksystem in einen theoretischen Kontext einzuordnen. Er verwendet den Begriff schließlich in einem formalen wissenstechnischen Sinn, in Orientierung an fachwissenschaftlichen Ansätzen wie etwa dem von Luft (1992).

"Unter einem Informatiksystem versteht man ein verteiltes, heterogenes, technisches System, das Wissen unterschiedlichster Art und Herkunft repräsentiert, diese Wissensrepräsentationen in Gestalt von Daten und Programmen verarbeitet und den Benutzern in geeigneter Form zur Verfügung stellt." (Baumann 1996, S.164) Diese formale auf Wissensakquisition und -repräsentation orientierte Auffassung von Informatiksystemen ermöglicht zwar das Einbinden von KI-Ansätzen. Es ist jedoch zu stark auf das Produkt Software und seine formalen Dimensionen fixiert und vernachlässigt die Prozesse seiner Erstellung und Anwendung. Die Gefahren eines solchen theoretischen Zugangs zu Informatiksystemen lauern vor allen in seiner unterrichtspraktischen Umsetzung. Im Mittelpunkt des Unterrichts stehen formale Operationen und Verfahrenstechniken, wie Algorithmen, Datenstrukturen, Kommunikationsprotokolle, Sprachsyntax, Fakten und Regeln etc. - sicher wichtige Elemente des Informatikunterrichts. Entscheidend ist jedoch ihre methodische Einbindung in Modellierungsprozesse, die Probleme der Genese von Software im Unterricht erschliessen. Bei einem ausschließlich an Formalismen orientierten Zugang entstehen Unterrichtskonzepte, bei denen die von Lehrplänen immer wieder eingeforderte Behandlung von gesellschaftlichen Auswirkungen von Informatiksystemen im Unterricht bestenfalls in Form von knappen Exkursen und als Alibiveranstaltungen an die 'wahren' informatischen Inhalte angefügt werden. Macht man hingegen die Modellierung von Informatiksystemen aus der Perspektive eines sozio-technischen Handlungssystems

zum Ausgangspunkt des Informatikunterrichts, könnte es gelingen, formale Operationen und informatische Prinzipien als Elemente des Softwareentwicklungsprozesses mit Fragen von Anwendungen und Auswirkungen von Informatiksystemen zu verbinden. Die letztgenannten Fragestellungen müssten nicht exkurshaft abgehandelt werden, sondern erweisen sich als zentrale Probleme der Softwareentwicklung bei Entwurfs- und Designentscheidungen. Indem bei der Entwicklung von Informatiksystemen zugleich auch die Ebene der Gestaltung von sozialen Handlungs- und Kommunikationsprozessen mit zu berücksichtigen ist, wird die Systemgestaltung und Modellierung zum eigentlichen Ausgangspunkt für Fragen der Anwendung und Auswirkung von Informatiksystemen und damit zum zentralen Unterrichtsinhalt.

In der Unterrichtspraxis hat z. B. Lehmann (1995) gezeigt, dass selbst ohne einen umfassenden systemtheoretischen Hintergrund im Rahmen eines Spiralcurriculums die Auseinandersetzung mit einem komplexen System im Informatikunterricht fruchtbare Ergebnisse bringen kann. Komplexe Systeme erfordern die Arbeit im Team und fördern damit eine Schlüsselqualifikation. Sie fördern vernetztes problemlösendes Denken und eignen sich für fächerübergreifenden Unterricht. Schüler müssen nicht sämtliche Module des Softwaresystems selbst entwickeln, sondern können, nachdem sie Funktionalitäten des vorhandenen Systems erkundet haben, dem System weitere Module hinzufügen. Kenntnisse über informatische Konzepte und Sprachsyntax werden hierbei ebenfalls erworben. Lediglich der Versuch, Prozesse der Modellbildung als auf sozialen Handlungs- und technischen Funktionszusammenhängen beruhende Entscheidungsprozesse vor allem in der Entwurfs- und Designphase der Softwareentwicklung im Unterricht zu thematisieren wird bei Lehmann nicht unternommen. Diesen weiteren Schritt versucht die systemorientierte Didaktik mittels des Konzepts der Dekonstruktion zu gehen. Wie er theoretisch begründet und unterrichtspraktisch umgesetzt werden kann, soll Gegenstand der weiteren Ausführungen sein.

2. Informatiksysteme aus fachwissenschaftlicher Perspektive

In den folgenden Kapiteln soll skizziert werden, wie systemorientierte Didaktik der Informatik und die Methode der Dekonstruktion in den fachwissenschaftlichen Diskussionen von Informatik, Techniksoziologie und Erziehungswissenschaften verankert werden könnte.

2.1 Systemgestaltung und Modellierung von Informatiksystemen

Zunächst soll der Frage nachgegangen werden, wie das Konzept der Dekonstruktion mit dem wissenschaftspropädeutischen Auftrag einer Didaktik der Informatik in Einklang zu bringen ist. Bezugspunkt ist das wissenschaftstheoretische Selbstverständnis des Faches Informatik und die dadurch begründeten zentralen Fachinhalte und Methoden. Hier stößt didaktische Theoriebildung an eine erste schwer zu überwindende Hürde. Die relativ junge Fachwissenschaft Informatik befindet sich hinsichtlich eines geschlossenen fachwissenschaftlichen Selbstverständnisses und der Entwicklung einer Theorie der Informatik noch in einem Anfangsstadium. Informatik wird u.a. als Ingenieur-, Arbeits-, Kognitions-, spezifische (Software)Technik-, Struktur- oder Gestaltungswissenschaft angesehen (vgl. Coy 1992). Eine Vielzahl von unterschiedlichen Sichtweisen erfordert von didaktischer Theoriebildung in ihrem Entdeckungs- und Begründungskontext (vgl. Magenheim 1992) unter Beachtung von bildungstheoretischen Konzepten eine fachwissenschaftliche Fokussierung auf die für den Unterricht wesentlichen Inhalte und Methoden. Eine entsprechenden Versuch hat beispielsweise Schwill (1993) mit seinem Konzept der fundamentalen Ideen unternommen. Hier wird jedoch eine weitgehende Beschränkung auf die formalen Methoden der Fachwissenschaft unternommen.

Systemorientierte Didaktik geht pragmatisch von einem zentralen Auftrag an die Informatik aus, wissenschaftliche Methoden und Verfahren bereitzustellen, die es ermöglichen, für gegebene Zwecke gebrauchsfähige Informatiksysteme erstellen können. Damit rücken Prozesse und Methoden der Systemgestaltung und Softwareentwicklung in den Mittelpunkt didaktischer Betrachtung. Um dies zu ermöglichen, sind einerseits formale, strukturierende Methoden z. B. zum Zwecke der Abstraktion von großer Bedeutung. Andererseits hat die für die Systemgestaltung wesentliche Modellierung technisch funktionale und kommunikative Prozesse, z.T. auch interpersonale Handlungsabläufe zum Gegenstand (vgl. z.B. Floyd 1992) Somit können, die mit einem Informatiksystem modellierten und bei dessen Einsatz vollzogenen Prozessveränderungen auf technisch-funktionaler, kommunikativer und interpersonaler Handlungsebene in didaktischen Betrachtungen nicht ausgeklammert bleiben. Probleme und Methoden der Schnittstellengestaltung, der Softwareergonomie, der Kommunikation in vernetzten Systemen und der Veränderung von sozialen Handlungssystemen etc. sollten deshalb ebenfalls Unterrichtsgegenstände sein.

Informatiksysteme bilden in der Perspektive systemorientierter Didaktik auch Zweck-Mittel Systeme. In bildungstheoretischer Tradition und im Gegensatz zu positivistischen wissenschaftstheoretischen Positionen stellt die Analyse der Zwecke und Ziele und der hiermit verbundenen Interessen einen weiteren wichtigen Ansatzpunkt für die Gewinnung von Methoden und Zielsetzung für den Informatikunterricht. Werden Zwecke und Ziele der Systemgestaltung eines Informatiksystems mit Prozessen der realen Praxis in Beziehung gesetzt, kann hierdurch die in vielen Lehrplänen geforderte Bewertung der Anwendungen und Auswirkungen von Informatiksystemen eingelöst werden.

Gestaltung von Informatiksystemen bewegt sich im Spannungsfeld von formalem Modell und nicht formaler Wirklichkeit, wobei der Gestaltungsprozess der Informatiker durch die Zweck-Mittel-Relation des Systems, z.B. von den Interessen der Auftraggeber, den ökonomischen Rahmenbedingungen der Systementwicklung, dem realen Ausgangszustand des sozio-technischen Systems, abhängt. Informatisches Modellieren ist deshalb auch nicht mit dem Versuch gleichzusetzen, eine möglichst realistische Abbildung eines Ausschnitts der realen Welt zu schaffen: *"Softwaretechnische Systeme als objektivierende Abbildungen von Weltausschnitten zu sehen, führt in die Irre. Sie sind vielmehr als partielle Definitionen von Handlungssystemen zu bewerten. Die Softwaretechnik muß Wandelbarkeit, Interpretierbarkeit in Handlungssystemen berücksichtigen. Dies ist nur möglich, wenn die ZWECKMÄßIGKEIT eines formalen Konstruktes in einem Anwendungskontext im Mittelpunkt der Überlegungen steht."* (Scheffe 1999, S.134)

Problemlösen im Kontext der Entwicklung von Informatiksystemen ist immer den Bedingungen dieser Zweck-Mittel Relation unterworfen. Informatisches Modellieren unterscheidet sich insofern wesentlich von in anderen Fachdidaktiken verwendeten Modellbegriffen. Problemlösen im Informatikunterricht, wird in der Methodik neben formal analytischen Anteilen von daher auch immer normative und reflexive Anteile, im Hinblick auf die Bewertung des eigenen Handelns, beinhalten.

2.2 Informatiksysteme als sozio-technische Systeme

Modellieren bezieht sich auf die Gestaltung von Informatiksystemen. Versuchen wir nun diesen Begriff in seinem wissenschaftstheoretischen Kontext zu charakterisieren. Der Begriff des sozio-technischen Informatiksystems, in dem ein von den mit dem Informatiksystem interagierenden Personen gebildetes Handlungssystem als Subsystem enthalten, bzw. mit ihm unauflösbar verbunden ist, hat seine wissenschaftstheoretischen Wurzeln nicht nur in der Informatik sondern auch in der Techniksoziologie. Die systemorientierte Technikwissenschaft hat ihre Ursprünge in der allgemeinen Systemtheorie von Bertalanffy (1949) und in der Kybernetik von Wiener(1948). Beeinflusst wurde sie sicher auch von Kontroversen in der Soziologie um die Systemtheorie von Luhmann (1984). Es liegt jedoch außerhalb der Möglichkeiten und Notwendigkeiten dieser Abhandlung, diese Debatte zu rezipieren. Syrbe (1995) hat in Anlehnung an Ropohl (z. B. 1999)versucht, den Begriff des sozio-technischen Informatiksystems für die informatische Diskussion nutzbar zu machen. Er bezieht sich auf Ropohls Konzept eines abstrakten Handlungssystems. Ropohl unterscheidet vier Systemmodelle: (1) abstrakte Handlungssysteme (Instanz, die Handlungen vollzieht; (2) menschliche Handlungssysteme (kooperierende Personen und Personengruppen, Organisationsmodelle); (3) Sachsysteme/technische Systeme (künstlich hergestellte, planmäßig nutzbare, gegenständliche Gebilde); (4)sozio-technische Systeme (soziale und personale Funktionsträger sind mit Sachsystemen aggregiert)(vgl. z.B. Syrbe 1995, S. 224) In die letzte Gruppe fallen auch die sozio-technischen Informatiksysteme, die es daher auch hinsichtlich ihrer sozialen Dimensionen zu analysieren gilt: *"The concept of the socio-technical system was established to stress the reciprocal inter-relationship between humans and machines and to foster the program of shaping both the technical and the social conditions of work, in such a way that efficiency and humanity would not contradict each other any longer."* (Ropohl 1999a, S. 1)

Die Zweck-Mittel Relation, die wir oben für Informatiksysteme konstatieren, wird dort auf einer allgemeineren Ebene für sozio-technische Systeme formuliert: *"I spelled out three relations that constitute technology: use-relations, causal relations between objects, and the evaluative relations of the archive. Use-relations define the technology's meaning, independently from the inventor's vision or the producer's design. This pragmatic concept helps us to avoid false generalizations from modern or western types of technologies. It is always technology-in-use and technology-in-a-situation that we experience. The causal relations or the fixed relations between input and expected output concern the interrelations of objects that I called interobjectivity. This relation limits the idealist view, but also the approach of radical constructivism that "anything goes," or that technology can be entirely socially shaped. And it also qualifies the notion of material causes, taking materiality as timely-emergent resistance against certain practices of human intervention. The evaluative relations of the technological archive are the mechanism to establish the state-of-the-art within technological fields."* (Rammert 1999, S. 11) Dies bedeutet vor allen Dingen auch, dass Informatiksysteme hinsichtlich ihrer realen Praxis und nicht nur hinsichtlich ihres ideellen (Software)konzepts zu beurteilen sind.

Hesse u.a. (1994) verwiesen von Seiten der Informatik bei ihrer Begriffsbestimmung für die Softwaretechnik auf den Systembegriff. In einem sozio-technischen Informatiksystem kann zwischen der technischen Repräsentationsebene (Maschinen) der Kommunikationsebene (Gruppe von Menschen) und der

Wissensebene (subjektive Sicht des einzelnen) unterschieden werden. Auf der technischen Ebene werden Daten in Form von Signalen und Symbolen ausgetauscht und verarbeitet. Auf der Kommunikationsebene dient die Sprache zum Austausch von Nachrichten. Auf der Wissensebene ist schließlich die Information angesiedelt, die des Interpretationskontextes für die Nachrichten bedarf. Daten bedürfen der Codierung in einer Sprache bevor sie als Nachricht und durch individuelle Interpretation zu subjektivem Wissen verarbeitet werden können. Dieses systemische Konzept stellt einen Zusammenhang zwischen Wissen, Information und Daten her und begründet mit seiner subjektivistischen Sicht von Wissen faktisch einen Gegensatz zu Begriffen wie 'Wissensgesellschaft' oder 'Informatiksystem als Wissenssystem', die einen dem Individuum entäußerten auf Maschinen übertragbaren und damit scheinbar objektivierte Wissensbegriff postulieren. Subjektive Wissensbestände und normative Positionen können die Bewertung und Entwicklung sozio-technischer Systeme maßgeblich beeinflussen, vor allem dann, wenn sie sich als wesentliche Bestandteile öffentlicher Meinung in einer Gesellschaft zum Maßstab praktischen Handelns werden: "The process of integrating knowledge, value-related, and other global, intellectual, or spiritual factors, and some religious ideas, should not be ignored when explaining the genesis and key motives and factors of technologically relevant initiatives, technological changes, and major innovations." (Tondl 1999, S. 12)

Subjektives Wissen und subjektive problembezogene Einstellungen (z.B. der Entwickler, Auftraggeber und Anwender) sind es auch, die Gestaltungsentscheidungen und die Nutzung von Informatiksystemen bestimmen. Diesen impliziten Prozessen nachzuspüren, die sich in der Software materialisiert haben, ist neben der formal-technischen Analyse von Informatiksystemen eine weitere wichtige Aufgabe der didaktischen Analyse einer systemorientierten Didaktik der Informatik. Dekonstruktion kann möglicherweise derartige Fragestellungen für den Informatikunterricht erschliessen.

Nygaard (1986) hat in seinem Perspektivenkonzept zur Softwareentwicklung betont, dass es sehr unterschiedliche und dennoch schlüssige Sichtweisen auf ein Informatiksystem geben kann, die die Bewertung von sozio-technischen Informatiksystemen bestimmen können. Damit ist es auch aus systemtheoretischer Perspektive keineswegs erforderlich, dass die Dekonstruktion von Informatiksystemen eindeutige 'objektive' Ergebnisse liefern muß.

Ropohls Ansatz beinhaltet auch die Forderung nach einer stufenweisen Konkretisierung der Begriffe und ihrer Füllung mit empirischen Gehalt. In Bezug auf sozio-technische Systeme und damit auch auf Informatiksysteme bedeutet dies, Technik hinsichtlich ihrer naturalen (ingenieur- und naturwissenschaftlich, ökologisch) humanen (ästhetischen) und sozialen (ökonomisch, technologisch) Dimensionen zu beurteilen. Die Umsetzung einer derartigen Mehrperspektivigkeit kann in der Schule sicher nur im Rahmen eines Spiralcurriculums erfolgen.

3. Systemorientierte Didaktik und Allgemeinbildung

Ohne auf die Allgemeinbildungsdebatte eingehen zu wollen, wie sie um die Bedeutung des Informatikunterrichts für allgemeinbildende Schulen von unterschiedlichsten Autoren geführt wurde (vgl. z.B. Bussmann, Heymann 1987) und wird (vgl. z. B. Rechenberg 1999) soll an dieser Stelle in knappster Form der besondere Bildungswert des Faches aus der Perspektive einer systemorientierten Didaktik der Informatik thematisiert werden. In Orientierung an Klafkis Begriff der epochaltypischen Schlüsselprobleme und an einem Konzept bildungstheoretischer Didaktik mit dem Begriff der kategorialen Bildung als Einheit von materialen und formalen Bildungsprozessen (vgl. Klafki 1995) sowie mit Bezug auf das oben skizzierte Konzept eines sozio-technischen Informatiksystems kann dem Informatikunterricht in einer Reihe von Bereichen eine allgemeinbildende Funktion zugesprochen werden.

An erster Stelle wäre hier nicht, wie in vielen anderen Legitimationskonzepten üblich, die Entwicklung von allgemeiner Problemlösekompetenz zu nennen, die mit dem Informatikunterricht an allgemeinbildenden Schulen erreicht werden soll. Sicher ist dies auch ein wichtiger Aspekt, der jedoch auch von anderen Fächern reklamiert wird und der angesichts der sich widersprechenden empirischen Ergebnisse hinsichtlich eines möglichen Transfers von Problemlösungskompetenzen in andere Kontexte kritisch hinterfragt werden müßte (vgl. z.B. Eberle 1996).

An erster Stelle wäre aus der Perspektive einer systemorientierten Didaktik der Informatik darauf zu verweisen, dass Informatik als einziges Fach (evtl. neben Arbeitslehre) mit technologischen und ingenieurwissenschaftlichen Bezügen an allgemeinbildenden Schulen in der Lage sein könnte, den Schülerinnen und Schülern den Umgang und die Auseinandersetzung mit Technologie insbesondere mit Informationstechnologien näher zu bringen. Hierdurch werden mehrfach allgemeinbildende Funktion im Sinne der o.g. epochaltypischen Schlüsselqualifikationen erfüllt:

In einer zunehmend von computerbasierten Medien und Technologien unterschiedlichster Art geprägten Welt kann den Schülerinnen und Schülern im Sinne der o.g. Zweck-Mittel Relation die sozialverträgliche

Technikgestaltung als interessen geleiteter Entscheidungsprozess verdeutlicht und so ein wichtiger Beitrag für ihre künftige Handlungskompetenz im Umgang mit Medien und Informationstechniken am Arbeitsplatz und in den Sphären von politischer Öffentlichkeit und Privatheit geleistet werden.

So verstandener Informatikunterricht kann auch einen wesentlichen Beitrag zum fundierten Umgang mit computerbasierten Medien in anderen Fächern leisten und somit zum Grundlagenfach im schulischen Fächerkanon avancieren. Anzuführen wären auch elementare Schlüsselqualifikationen die sich auf den Bereich von Teamfähigkeit und kooperativem Arbeiten und Lernen, insbesondere in vernetzten Systemen, beziehen. Ferner ist ein wesentlicher Beitrag zur Entwicklung der Fähigkeit zu erwartenden Informationen zu einem gegebenen Zweck sinnvoll zu nutzen, indem Konzepte der Informationsrecherche und der Informationsorganisation, die im Informatikunterricht vermittelt werden, in individuelle Lern- und Arbeitsprozesse integriert werden.

Informatikunterricht ist damit ein zentrales Fach mit allgemeinbildendem Anspruch in einer postindustriellen Informationsgesellschaft mit einem hohen Grad an Zukunftsrelevanz für Schülerinnen und Schüler.

4. Dekonstruktion

Die Einsicht, dass Informatiksysteme als sozio-technische Systeme aufgrund der ihnen impliziten Modellierung soziale Interaktionen und aufgabenbezogene Funktionen in ihrem Nutzungsumfeld in einem bestimmten Umfang festzuschreiben, führt in einer systemorientierten Didaktik zu der Fragestellung, ob es jenseits traditioneller Unterrichtsmethodik und Unterrichtsgegenstände, nämlich Software im Informatikunterricht von Schülern anwenden oder selbst entwickeln zu lassen, andere Zugänge zu Informatiksystemen gibt. Zugänge, die möglicherweise Prozesse der Softwareentwicklung erschließen in dem sie bei einer im Unterricht verwendeten komplexeren Software einzelne Modellbildungsphasen wie z.B. potentielle Konfliktsituationen bei Entwurfsentscheidungen nachvollziehen lassen. Dies führt uns zum Begriff der Dekonstruktion von Informatiksystemen. Der Begriff Dekonstruktion ist eine zentrale Kategorie des Konzepts einer systemorientierten Didaktik der Informatik. Er beschreibt einerseits eine wissenschaftstheoretische Methode zur Analyse und Erschließung von Informatiksystemen für didaktische Fragestellungen im Informatikunterricht. Andererseits kann daraus auch ein konkretes unterrichtsmethodisches Vorgehen im Informatikunterricht abgeleitet werden.

Es soll zunächst in der gebotenen Kürze auf den sehr komplexen wissenschaftstheoretischen Hintergrund des Begriffs Dekonstruktion und seine Verwendung im Rahmen eines systemorientierten Ansatzes zur Informatikdidaktik eingegangen werden. Anschließend wird seine unterrichtspraktische Bedeutung herausgearbeitet.

4.1 Wissenschaftstheoretische Bezüge

Ursprünglich entstammt der Begriff Dekonstruktion wissenschaftlichen Diskursen vor allem in der Philosophie aber auch in Literaturwissenschaft, Wissenschaftstheorie, Kunst und Architektur. Ansätze dieser Diskussion lassen sich in der bereits von Kant bis hin zu Adorno thematisierten Frage der Ästhetik erkennen, ob Kunstwerken ein begriffliches Äquivalent zuzuordnen sei (vgl. Zima 1994). Später konzentrierte sich die Diskussion in der Literaturwissenschaft auf die Frage nach der begrifflichen Erfassung von Texten. Man könnte diese Fragestellung in dem hier diskutierten Kontext sicher um die Frage erweitern, ob Software primär ingenieurwissenschaftliches Produkt oder partiell auch Kunstwerk sei und ob es vor allem hinsichtlich seiner impliziten Modelldeterminanten begrifflich und sprachlich kommunikativ charakterisiert werden kann.

Mitte der sechziger Jahre entwickelte Jacques Derrida (1997) eine Methode zur Beurteilung von Texten, die er als Dekonstruktion bezeichnete. Derrida entwarf seinen Begriff u.a. in Anlehnung an den Begriff der Destruktion des Philosophen Heidegger, der damit nicht Zerstörung sondern kritische Würdigung und Abkehr von tradierten Methoden des Erkenntnisprozesses bezeichnete. Dekonstruktion steht in der Literaturtheorie im Widerspruch zum Konzept des Strukturalismus. Nicht ein externes Normensystem wird zur Bewertung und zum Vergleich von Texten herangezogen, sondern durch den intensiven Nachvollzug des argumentativen Aufbaus des Textes wird versucht, Widersprüchlichkeiten des im Text etablierten Begriffssystems zu entdecken. Dekonstruktion bedeutet Auflösung und Entstrukturalisierung des Textes. Vor allem Ungesagtes, Angedeutetes kann von zentraler Bedeutung sein. Dies gilt auch für die Software, deren Modellierungsprämissen und Entwurfsentscheidungen, wenn überhaupt, nur implizit oder gar nicht mehr erkennbar sind und quasi archäologisch in Schichten freigelegt werden müssen. Ein Ansatz, der in der Medientheorie vor allem von Foucault (1999) mit seiner Wissensarchäologie verfolgt wurde, indem er das Verfahren der Diskursanalyse von Texten auf andere Medien zu übertragen suchte.

Während nach strukturalistischem Konzept der Text hinsichtlich bedeutungstragender Strukturen zu erforschen ist und eine logische widerspruchsfreie Analyse objektive wissenschaftliche Resultate liefert, gehen die Dekonstruktivisten von im Text implizit verschlüsselten Botschaften, Widersprüchen und Brüchen aus, die es aufzudecken gilt, die sich letztlich jedoch objektiver Bewertung entziehen. In diesem Konzept zeigen sich Berührungspunkte und Kontrapositionen zur Diskursanalyse der Kritischen Theorie und dem Konzept der Hermeneutik.

Dieses Vorgehen trägt dem Dekonstruktivismus von seinen Kritikern u.a. von Seiten der Kritischen Theorie zu Recht den Vorwurf ein, kommunikative Elemente in ihrer Theorie auszublenden und die Verschränkung interpretativer Bemühungen mit dem subjektiven theoretischen Kontext des Interpretieren zu vernachlässigen.

Diese vor allem im Umgang mit Texten angewandte Methode kann sicher nicht ohne Einschränkungen auf andere Wissenschaftsbereiche übertragen werden. Vor allem das differenzierte Begriffsinstrumentarium der Literaturtheorie kann im Kontext der Informatik kaum oder nur sehr eingeschränkt verwendet werden. Das zentrale Konzept des Ansatzes, dass Textanalyse keine eindeutigen objektiv nachvollziehbaren Bewertungsergebnisse liefert, sondern eine Vielzahl von Sichtweisen gleichberechtigt nebeneinander stehen können, gilt auch für die Textsorte Software.

4.2 Dekonstruktion im Konzept einer systemorientierten Didaktik der Informatik

In der systemorientierten Didaktik der Informatik wird auf diese Diskussion insofern Bezug genommen, als Software auch als Text zur Beschreibung und Steuerung von maschinellen Aktivitäten des Informatiksystems und von Mensch-Maschine Interaktionen angesehen werden kann. In diese 'Beschreibung' sind implizit Entwurfs- und Designentscheidungen der Entwickler eingeflossen, die jedoch z.T. irreversibel und daher nicht mehr reproduzierbar sind. Dekonstruieren von Software heißt insofern, in einer ersten Annäherung an den Begriff, interpretieren von Software, um bei den Lernenden im Informatikunterricht Sensibilität für Gestaltungsräume und für Prozesse der Softwareentwicklung zu erzeugen. Dekonstruktion ist daher mehr als das Lesen von Quellcode einer Software, zum Zwecke des Erlernens der Syntax einer Programmiersprache anhand eines Beispiels. Modellannahmen und Entwurfs- und Designentscheidungen können hypothetisch extrahiert, nicht aber eindeutig belegt werden. Damit werden aber Spielräume für Gestaltungsalternativen offengelegt. Softwareentwicklung wird als interessensgeleiteter Entscheidungsprozess identifiziert, der von den Rahmenbedingungen abhängt, unter denen das Entwicklerteam arbeitet, insbesondere von den Interessen der Auftraggeber und der unterschiedliche soziale und gesellschaftliche Folgen hervorrufen kann.

Diese Zusammenhänge deutlich zu machen, sollte eine wesentliche Aufgabe des Informatikunterrichts sein. Dekonstruktion kann den Blick auf verschiedene Abstraktionsebenen und Repräsentationsformen von Informatiksystemen eröffnen: Die Benutzungsoberfläche und den dort vorhandenen interaktiven grafischen Elementen und ikonischen Repräsentationen von Softwarefunktionalitäten, die Funktionen der Software und deren potenzielle Einbettung in den sozialen Interaktionskontext des Informatiksystems, den Quellcode als textuelle Repräsentationsform der Software des Informatiksystems, den Assemblercode oder gar den binären Maschinencode, der zugleich auch die Verbindung zur Hardwarearchitektur herstellt.

Informatiksysteme erscheinen aus der letztgenannten Perspektive als semiotische, zeichenverarbeitende Maschinen, die Zeichenketten (Daten und Programme) verarbeiten. Dieses Konzept beinhaltet verschiedene Ansatzpunkte für theoretische Überlegungen und didaktische Zugänge zu Informatiksystemen im Unterricht, die die Vielfalt der Methoden und Gegenstände der Fachwissenschaft Informatik auch der Praxis des Informatikunterrichts erschliessen können und diesen nicht zu einem Codierunterricht reduzieren.

Auf der Ebene der elementaren symbolischen Operationen und der Formalismen steht das zentrale Konzept der theoretischen Informatik, die Turingmaschine, als Synonym für den Kern des Informatiksystems. Die Transformation der Zeichen gehorcht formalen, ebenfalls mit Zeichen codierten Regeln, deren Sinnhaftigkeit außerhalb dieses Betrachtungshorizonts liegt. Der Formalismus genügt den Kriterien von Schriftlichkeit, Schematisierbarkeit und Interpretationsfreiheit (Krämer 1988), d.h. die Symbole sind typografisch fixierbar und der Sinn der Zeichen ist für die formalen, nach einem festen Schema ablaufenden Operationen irrelevant. Der Versuch von Dekonstruktion von Text auf dieser Ebene erschließt wesentliche Fragestellungen der theoretischen Informatik, wie etwa die der Berechenbarkeit, der formalen Sprachen oder der Grenzen der Algorithmisierung.

Wegners interaction machine (Wegner 1997) zeigt diese Grenzen hinsichtlich Interaktivität und Kommunikation auf und verdeutlicht, dass die textuelle Materialisierung von Software, egal auf welcher Abstrak-

tionsebene betrachtet, die unendliche Vielzahl von Interaktionshandlungen mit dem Informatiksystem, letztlich codierbar in Sequenzen von Zeichen, zunächst einmal ausgrenzt. Ohne diese Textualisierung sozialer Interaktionen im Kontext des Informatiksystems bleibt dessen Beschreibung jedoch unvollständig. Die Dekonstruktion von Informatiksystemen hat diese Dimension mit zu berücksichtigen, indem sie, zumindest exemplarisch, typische Interaktionsmuster (use cases) als Handlungsabläufe mit und in dem System herausarbeitet.

Die symbolische oder bildliche Präsentation von Informationen, Kommunikation und Interaktion mit dem Computer, genauer mit seiner durch Software definierten Benutzungsschnittstelle, rückt die medialen und kommunikativen Funktionen von Informatiksystemen in den Vordergrund der Betrachtung. Deren (software)technische Realisierung und Nutzung im Informatiksystem kann ebenfalls Gegenstand von Dekonstruktionsprozessen sein, indem Kommunikations- und Handlungsprozesse innerhalb des Informatiksystems im Sinne einer symbolvermittelten Kommunikation verstanden werden. Sie führt in didaktischer Perspektive zu einer Vielzahl von Fragestellungen, die das Verständnis vom Computer als technischem Part eines Informatiksystems und als instrumentalem Medium (Schelhowe 1997) fördern. Damit wird dem Informatikunterricht zugleich eine wichtige Aufgabe bei der Fundierung schulischer Medienbildung zugeordnet (vgl. Gl 1999).

Das Eingehen auf derartige Fragestellungen führt von der Arbitrarität der Zeichen zur Sinnfrage und der Interpretierbarkeit von Software als Text. Damit wird die Ebene des Formalismus zumindest partiell verlassen. Dekonstruktion bewegt sich hier zwischen der Ebene der Formalismen und Regeln und der in der Phase der Modellierung beim Softwareentwurf und Design angelegten Funktionalität der Software im Kontext des Informatiksystems. Die funktionale Einbettung der Software in den Kontext sozialer Interaktionen des Informatiksystems erfolgt nicht zuletzt über die Benutzungsschnittstelle. Folglich muß Dekonstruktion auch die Medialität der Benutzungsschnittstelle berücksichtigen, um die in der Software typisiert angelegten Interaktionsschemata als Impulse für real ablaufende soziale Interaktionsprozesse im Informatiksystem sichtbar zu machen.

Hierbei ist deutlich zwischen dem vorliegenden Produkt 'Software', dem zeitlich vorgelagerten Prozess seiner Herstellung und den gegenwärtig ablaufenden und künftig zu antizipierenden Prozessen im Informatiksystem zu unterscheiden.

Bezogen auf die Softwareerstellung kann Dekonstruktion einen Prozess der Formalisierung und Operationalisierung partiell revitalisieren. In diesem Prozess wurden Funktionen des Informatiksystems im Rahmen der Modellierung und den verschiedenen, sich z.T. wiederholenden Phasen der Softwareentwicklung als Resultat von Kommunikationsprozessen zwischen Entwicklern und Auftraggebern mittels formaler Strukturen (Folgen von Zeichen) beschrieben. In dem so entstandenen (Quell)Text sind Algorithmen und Datenstrukturen, formalisierte Interaktionen in Form definierter Benutzungsschnittstellen, Metastrukturen wie Klassen und Objekte oder Fakten und Regeln auszumachen. Über das Produkt wird der in der Vergangenheit liegende Prozess der Modellbildung mittels Dekonstruktion erschlossen. Der Prozess der Analyse der gegenwärtigen Nutzung des Produkts kann hierzu weitere Impulse für die Dekonstruktion liefern. Dekonstruktion erschließt hierbei vielfältige Sichtweisen auf den Modellierungsprozess ohne jedoch den 'wahren, objektiven' tatsächlichen Prozess von Entwicklerentscheidungen je nachvollziehen zu können.

An dieser Stelle zeigt sich die Nähe des Konzepts der Dekonstruktion zu Nygaards Theorie der Vielsichtigkeit auf Informatiksysteme (Nygaard 1986) und zu konstruktivistischen Sichtweisen der individuellen Erschließung von Weltsicht. Insofern ist der Prozess der Modellbildung, der zum Produkt 'Software' führt, wichtig für die angemessene Funktionalität des Informatiksystems und hierüber sollte während der Dekonstruktion eine interpersonale Verständigung in der Bewertung versucht werden. Marginal bleibt die Fragestellung, ob Modellierung in der Informatik soziale Wirklichkeit abbildet; wesentlich ist das adäquate Funktionieren der Software im Kontext des Informatiksystems.

5. Modellieren und Dekonstruieren im IU

Der im oben beschriebenen Sinne systemtheoretisch verstandene Begriff des Informatiksystems und das Konzept der Dekonstruktion erschließt nicht nur verschiedene eng miteinander verbundene Sichtweisen auf ein Informatiksystem, sondern bietet auch den Rahmen für die weitere Bestimmung grundlegender Gegenstandsbereiche und Methoden der Informatik, die in einem bildungstheoretischer Didaktik verpflichteten Informatikunterricht behandelt werden können.

In der weiteren Darstellung wollen wir uns auf die unterrichtsmethodischen Implikationen der Dekonstruktion konzentrieren und mit dem Verhältnis von Dekonstruktion und Konstruktion von Informatiksystemen im Unterricht auseinandersetzen. Fast seit Beginn der Einführung des Schulfaches Informatik

dominierte das Entwickeln von Software begrenzter Komplexität - im schlimmsten Fall das Codieren eines vorgegebenen Algorithmus in der Syntax einer Programmiersprache - die Praxis des Informatikunterrichts. Das traditionelle Wasserfallmodell der Softwareentwicklung stand bei der methodischen Gestaltung des Informatikunterrichts in diversen didaktischen Ansätzen Pate. Ein an dieser Konzeption orientierter Informatikunterricht gerät jedoch schnell in Widerspruch zu seinem wissenschaftspropädeutischen Anspruch. Das Wasserfallmodell ist im Bereich des Softwareengineerings längst von Methoden partizipativer Softwareentwicklung mit einem hohen Grad an Interaktivität, Kommunikation und Phasen intensiver Rückkopplung zwischen Entwicklerteams und Auftraggebern abgelöst worden. Die Komplexität der Softwareprojekte und der zu ihrer Konstruktion eingesetzten CASE-tools im fachwissenschaftlichen Bereich kann angesichts der im Unterricht entwickelten 'Projekte' mit geringer Komplexität dort nur unzureichend abgebildet werden. Fachwissenschaftliche Neuorientierungen, wie etwa die Objektorientierung, ermangelt im unterrichtlichen Einsatz daher oft die fachlich begründete Notwendigkeit.

Mittels einer aus system- und handlungstheoretischer Sicht auf Informatiksysteme gewonnenen Unterrichtskonzeption werden dem Informatikunterricht im Vergleich mit dieser Praxis weitergehende inhaltliche Schwerpunkte zugeordnet, die einen mehrperspektivischen Blick auf die Gegenstandsbereiche ermöglichen:

- Modellierung und Softwareentwicklung bleiben in der Tradition anwendungsorientierter Ansätze der Didaktik der Informatik wichtige Gegenstände des Informatikunterrichts. Um die für einige Fragestellungen (Objektorientierung, Analyse von Handlungsabläufen im sozio-technischen Informatiksystem) notwendige Komplexität im Unterricht zu gewährleisten und Modellierungsprozesse in den Mittelpunkt des Unterrichts zu rücken, werden Entwicklungsumgebungen zur Konstruktion und Visualisierung von Zusammenhängen, didaktische Software für Zwecke der Dekonstruktion eingesetzt.
- Kommunikation mit, in und zwischen Informatiksystemen unter technischen und handlungstheoretischen Gesichtspunkten können einen weiteren zentralen Unterrichtsgegenstand bilden. Dies bezieht sich sowohl auf das Nutzen und Anwenden als auch auf die Analyse und Bewertung derartiger Kommunikationsabläufe. Die medialen Funktionen von Informatiksystemen können unter dieser Betrachtungsperspektive ebenso erschlossen werden, wie kooperative und arbeitsorganisatorische Aspekte.
- Dekonstruktion eröffnet den Blick auf die Theoretische Informatik nicht nur aus der Sicht von Grammatik, Sprache und Algorithmus sondern auch aus der Perspektive von Symbolverarbeitung und Kommunikation und erschließt die Medialität von Informatiksystemen auf theoretischer Ebene.
- Sozio-technische Informatiksysteme können hinsichtlich der in ihnen enthaltenen informatischen Prinzipien, wie z.B. Informationsorganisation, Arbeits- und Lastenverteilung, Kooperation und Arbeitsorganisation untersucht werden.

Konstruktion und Dekonstruktion von Informatiksystemen haben als Methoden im Informatikunterricht einen gleichberechtigten Stellenwert. Dekonstruktive Unterrichtsphasen können einerseits als methodische Bereicherung angesehen werden. Andererseits vermögen sie schneller als konstruktives Vorgehen, komplexe Betrachtungsweisen von Informatiksystemen im Unterricht einzuführen und anhand von didaktischer Software zu konkretisieren. Modellieren von Informatiksystemen, insbesondere objektorientiertes Modellieren, sollte wesentlicher Gegenstand des Informatikunterrichts sein. Interaktive computergestützte Modellierungswerkzeuge (z.B. auf UML-Basis) eröffnen die Chance, diesen Prozess nicht nur deklamatorisch sondern auch unterrichtspraktisch und fachlich fundiert mittels geeigneten Visualisierungen im Informatikunterricht zu verankern.

Unterrichtsmethodisch gesehen bedeutet Dekonstruktion in diesem Kontext: Erkunden der Funktionalität der Software, von Gestaltungsmerkmalen ihrer Benutzungsoberflächen und partieller Analyse des implementierten Sprachcodes an ausgewählten Beispielen sowie das sukzessive Erlernen von Modellierungs- und Gestaltungsprozessen anhand des Beispiels. Methodischer Ausgangspunkt von Informatikunterricht nach dem Konzept der systemorientierten Didaktik der Informatik ist Software, mit hinreichender jedoch unter didaktischen Gesichtspunkten reduzierten Komplexität. Hierbei ist von der Phänomenologie des Systems auszugehen und über die Erprobung der Systemfunktionalität zu analytischen Arbeitsphasen, die informatische Prinzipien erschliessen, fortzuschreiten. Dekonstruktion beschränkt sich jedoch nicht auf die Analyse von Quellcode. Sie soll vielmehr darüber hinausgehend das Informatiksystem und seine assoziierten sozialen Handlungszusammenhänge in ihren Funktionen erschliessen und das der Software implizite Modell eines Realitätsausschnitts so weit als möglich explizieren. Verfahren des Modellierens können erprobt, Entwurfsentscheidungen in ihrer Konsequenz für die Gestaltung des Informatiksystems partiell nachvollzogen, eigene Kenntnisse im Umgang mit Modellierungs- und Entwicklungsumgebungen erworben werden. Potenzielle Alternativen zu realisierten Entwurfs- und Design-

entscheidungen und deren Bewertung können Anlaß zur Auseinandersetzung mit Fragen der Anwendung und Auswirkung von Informatiksystemen dienen und auf diese Weise die gesellschaftliche Dimension von Informatiksystemen sinnvoll in den Unterricht integrieren.

Unterrichtsmethodisch ist auch eine Verschränkung von konstruktiven und dekonstruktiven Arbeitsphasen sinnvoll (vgl. Hampel u.a. 1999). Während durch Dekonstruktion einer Software größere systemische Zusammenhänge erschlossen werden, können durch ergänzende themenbezogene kleinere Beispiele informatische Konzepte vertieft eingeübt werden. Dekonstruktion und praxisbezogene Übungsphasen mit kleineren Beispielen wechseln sich ab. Diese Unterrichtsmethodik kann als Oszillieren zwischen Dekonstruktion und Konstruktion, als Wechsel zwischen komplexerem System und kleinerer Aufgabenstellung mit Programmieranteilen, als inhaltlich aufeinander bezogene Abfolge zwischen 'Modellieren im Großen' und 'Programmieren im Kleinen' angesehen werden.

Realisierbar ist dieses Konzept nur, wenn entsprechend gestaltete didaktische Software für Unterrichtszwecke vorliegt. Diese Software sollte eine Reihe von Kriterien erfüllen: Hinreichende Komplexität, Quellcode gut strukturiert und dokumentiert, zugrunde liegende Entwurfs- und Designkonzepte widerspiegeln wesentliche Konzepte des Softwareengineerings (z.B. Objektorientierung, Ereignisorientierung, model-view-control Konzept..), Benutzungsoberflächengestaltung entspricht zentralen Anforderungen der Softwareergonomie, zentrale informatische Ideen und Methoden über Software zugänglich, Nutzungskontext der Software für Schüler erschließbar, exemplarische 'didaktische Fenster' auf tieferliegende Schichten des Informatiksystems möglich (z.B. Symbolverarbeitung), mediale Funktionen des Softwaresystems beispielhaft analysierbar, ergänzende Dokumentation über Modellierungsprozess verfügbar, Visualisierung von impliziten Konzepten (z.B. Klassenhierarchien, Sequenzdiagramme etc..) mit geeigneten Entwicklungsumgebungen möglich etc.

In diesem Bereich ist sicher noch viel Entwicklungsarbeit zu leisten, sowohl was die didaktische Software und die im Unterricht einzusetzenden computergestützten Modellierungswerkzeuge als auch was die unterrichtspraktische Erprobung und Evaluation des Konzepts angeht. Um den Prozess der Entwicklung von Software voranzutreiben, die den oben genannten Kriterien Rechnung trägt, wäre ein open source Konzept denkbar, das didaktisch Interessierten übers Internet, etwa über einen portal server, die Möglichkeit gibt, vorhandene Kerne didaktischer Software weiter auszubauen und zu dokumentieren. Die AG Didaktik des Fachbereichs Informatik der Universität Paderborn arbeitet an einem solchen Vorhaben.

Weiteres wichtiges Element eines derartigen didaktischen Service wäre es, unterrichtspraktische Erfahrungen mit dem Konzept der Dekonstruktion auszutauschen und insbesondere (computergestützte) Modellierungstechniken im Informatikunterricht fest zu verankern (CRC-Techniken, UML-Beschreibungstechniken, Einsatz von Entwicklungstools etc.; vgl. z.B. Booch 1996) Hierzu gehört auch das Einbeziehen von multimedialen Elementen in den Unterricht, wie etwa den Aufbau einer softwarebezogenen multimedialen Datenbank mit Videosequenzen über use cases, die im Unterricht zur Systemmodellierung genutzt werden könnte.

6. Forschungsfragen

Für die didaktische Forschung ergeben sich aus der Perspektive der systemorientierten Didaktik der Informatik eine Fülle von zu klärenden Fragestellungen. Auf unterrichtspraktischer Ebene ist z.B. die Entwicklung didaktischer Software für Zwecke der Dekonstruktion im Sinne des oben beschriebenen open source Konzepts voranzutreiben. Mit Methoden empirische Unterrichtsforschung ist zu klären, ob die mit dem Ansatz eingeforderte Komplexität von Unterrichtsgegenständen Schülerinnen und Schüler nicht überfordert. Es sollte untersucht werden, wie Phasen der Dekonstruktion mit konstruktiven Unterrichtszugängen, wie Systemkomplexität und Fokussierung auf Detailprobleme informatischen Inhalts, wie Fragen der gesellschaftlichen Anwendungen und Auswirkungen von Informatiksystemen mit dem Problem der Systemgestaltung unterrichtsmethodisch sinnvoll zu verknüpfen sind. Von besonderer Relevanz ist auch die empirische Klärung der Frage, ob mit dem Konzept der Dekonstruktion sowohl operational formale als auch normativ bewertende Qualifikationen der Systemgestaltung im Informatikunterricht vermittelt werden können.

Die unterrichtspraktische Umsetzung von Dekonstruktion und Modellbildung, erfordert zunehmend den Einsatz computerbasierter Modellierungswerkzeuge und Entwicklungsumgebungen. Neben der Frage der Adaption der Werkzeuge für schulische Zwecke ist hier vor allem die Entwicklung von unterrichtsmethodischen Konzepten gefragt, die die Integration derartiger Modellierungswerkzeuge und der mit ihnen zu erzeugenden Sichten auf die Modellierung von Informatiksystemen in Lernprozesse des Informatikunterrichts fördern.

Die u.a. mit der Dekonstruktion verbundene Perspektive auf Informatiksysteme als semiotische zeichenverarbeitende Maschinen mit hoher Kommunikativität innerhalb ihres sozio-technischen Kontexts kann einen theoretischen Rahmen für die didaktische Auseinandersetzung mit dem Verhältnis von Informatikunterricht und Medienbildung bieten. Hier wäre vor allem zu klären, welchen fundierenden Beitrag der Informatikunterricht in den einzelnen Schulstufen für die Medienbildung zu leisten vermag. Auf theoretischer Ebene sind verschiedene Aspekte des Begriffs 'sozio-technisches Informatiksystems' und dessen Bezüge zu einer möglichen Theorie der Informatik zu analysieren, um auf diese Weise unter Berücksichtigung empirischer Evaluationen des didaktischen Konzepts einen Beitrag zur sukzessiven Formulierung einer Theorie der systemorientierten Didaktik der Informatik zu leisten.

Literatur

- Baumann, R., Didaktik der Informatik, Stuttgart u.a. 1996
- Belde, H., Informatik als Strukturwissenschaft, in: LOG IN 19, H. 2, S.17 - 27, 1999
- Bertalanffy, L.v., General System Theory, New York 1968
- Booch, G., Objektorientierte Analyse und Design, Bonn u.a. 1996
- Bussmann H. / Heeymann H.W., Computer und Allgemeinbildung, in: Neue Sammlung 27, H 1. S. 3 - 39,1987
- Coy, W. u.a. (Hrsg.), Sichtweisen der Informatik, Braunschweig/Wiesbaden 1992
- Derrida, J., Die Struktur, das Zeichen und das Spiel im Diskurs der Wissenschaften vom Menschen, in: Engelmann, P. (Hrsg.) s.u. 1997
- Engelmann, P. (Hrsg.), Postmoderne und Dekonstruktion, Texte französischer Philosophen der Gegenwart, Stuttgart 1997
- Eberle, F., Didaktik der Informatik bzw. einer informations- und kommunikations- technologischen Bildung auf der Sekundarstufe II, Aarau 1996
- Floyd, C. u.a.(eds), Software Development and Reality Construction, Berlin u.a. 1992
- Floyd, C. u.a. Scanorama, Methoden, Konzepte, Realisierungsbedingungen und Ergebnisse von Initiativen alternativer Softwareentwicklung und -gestaltung in Skandinavien, Werkstattbericht Nr. 30, hrsg. v. Ministerium für Arbeit, Gesundheit und Soziales des Landes Nordrhein-Westfalen, Reihe Mensch und Technik, Sozialverträgliche Technikgestaltung, Düsseldorf 1987
- Foucault, M., Botschaften der Macht, Stuttgart 1999
- Gesellschaft für Informatik (Hrsg.):Informatische Bildung und Medienerziehung, Empfehlung der Gesellschaft für Informatik e.V. erarbeitet von einem Arbeitskreis des Fachausschusses „Informatische Bildung in Schulen" (7.3), 1999
- Gesellschaft für Informatik (Hrsg.): Veränderte Sichtweisen für den Informatikunterricht, GI-Empfehlungen für das Fach Informatik in der Sekundarstufe II allgemeinbildender Schulen. Erarbeitet vom Arbeitskreis 7.3.1 "Informatik in der Sekundarstufe II" im FA 7.3 der Gesellschaft für Informatik e.V., 1993
- Hesse, W. u.a., Terminologie der Softwaretechnik, ein Begriffssystem für die Analyse und Modellierung von Anwendungssystemen, Teil 1: Begriffssystematik und Grundbegriffe, in: Informatik-Spektrum 17, S. 93-47, 1994
- Hampel, T.; Magenheim, J.; Schulte, C.: Dekonstruktion von Informatiksystemen als Unterrichtsmethode. Zugang zu objektorientierten Sichtweisen. In: Schwill, A. (Hrsg.): Informatik und Schule. Fachspezifische und fachübergreifende didaktische Konzepte. Berlin, Heidelberg, New York u.a. (Springer) 1999
- Lehmann, E., Komplexe Systeme, Eine fundamentale Idee des Informatikunterrichts, in LOG IN 15, Heft1, S.29-37, 1995
- Luhmann, N., Soziale Systeme, Grundriß einer allgemeinen Theorie, Frankfurt/M 1984
- Luft, A. L., Wissen und Information bei einer Sichtweise der Informatik als Wissenstechnik, in: Coy, W. u.a. (Hrsg.), Sichtweisen der Informatik, Braunschweig/Wiesbaden S. 49 -70, 1992,
- Klafki, W., Schlüsselprobleme als thematische Dimension einer zukunftsbezogenen "Allgemeinbildung" - Zwölf Thesen, in: Die Deutsche Schule, 3. Beiheft 1995, Schlüsselprobleme im Unterricht, S. 9 - 14, 1995
- Magenheim, J, Neue Technologien in der Schule, in: Tagungsband zur Fachtagung der Gesellschaft für Informatik Thüringen am 21. 3. 1992, 'Computer in der Schule', Jena 1992
- Nygaard, K., Program Development as a Social Activity. in: H. Kugler (Hrsg.), Information Processing 86, Amsterdam 1986
- Rammert, W. Relations that constitute technology an media that make a difference: Toward a social pragmatic Theory of technicization, in: : Society for Philosophy & Technology, Vol. 4. No. 3 1999, URL: <http://scholar.lib.vt.edu/ejournals/SPT/v4n3/>
- Rechenberg, F., Mythen und Fetische des Computerzeitalters, in: LOG IN 19, Heft 2, S. 28 - 33, 1999
- Ropohl, G., Allgemeine Technologie, Eine Systemtheorie der Technik, München u.a. 1999
- Ropohl, G., Philosophy of Socio-Technical Systems in: Society for Philosophy and Technology, Vol. 4, No 3, Spring 1999a, URL: <http://scholar.lib.vt.edu/ejournals/SPT/v4n3/>
- Schefe, P., Softwaretechnik und Erkenntnistheorie, in.: Informatik-Spektrum 22, S.122-135, 1999
- Schelhowe, H., Das Medium aus der Maschine, Zur Metamorphose des Computers, Frankfurt/ New York, 1997
- Schwill, A., Fundamentale Ideen der Informatik, in: Zentralblatt für Didaktik der Mathematik - ZDM, 25, H.1, S. 20-31, 1993
- Syrbe, M., Über die Notwendigkeit einer Systemtheorie in der Wissenschaftsdisziplin Informatik, in: Informatik-Spektrum 18, S.222 - 227, 1995
- Tondl, L., Information and Systems Dimensions of Technological Artifacts, in: Society for Philosophy & Technology, Vol. 4. No. 3 1999, URL: <http://scholar.lib.vt.edu/ejournals/SPT/v4n3/>
- Wegner, P., Why Interaction is More Powerful Than Algorithms, in: Communications of the ACM, Vol. 40, No.5, p 81-91, May 1997
- Wiener, N., Cybernetics or Control and Communication in the Animal and the Machine, Cambridge, Mass. (zuerst 1948) 1961
- Zima, P.V., Die Dekonstruktion, Tübingen, Basel 1994