

# Zur wissenschaftlichen Fundierung der Schulinformatik

Ludger Humbert

Vom Fachbereich Elektrotechnik und Informatik der Universität Siegen als Dissertation  
zur Veröffentlichung freigegeben.

Erste Gutachterin: Prof. Dr. S. Schubert

Zweiter Gutachter: Prof. Dr. A. Schwill

Tag der mündlichen Prüfung: 13. August 2003

**urn:nbn:de:hbz:467-481**

# **Zur wissenschaftlichen Fundierung der Schulinformatik**

Ludger Humbert

© pad-Verlag

Witten 2003

Printed in Germany

ISBN 3-88515-214-2

*Danke ... Natti, Perli, Grinsel, ...*

## Kurzfassung

Mit dem Modulkonzept wird eine Grundlage für die Schulinformatik entwickelt und evaluiert. Zur wissenschaftlichen Fundierung werden die Beziehungen zwischen Wissenschaftstheorie und Informatik herangezogen. Ergebnisse lerntheoretischer Forschungen beleuchten den Forschungsgegenstand aus der Perspektive des Lernens. Unter Berücksichtigung der Geschichte der Schulinformatik findet der aktuelle Stand der fachdidaktischen Forschungen Eingang in das Forschungsprogramm. Die empirische Untersuchung der Lehrexpertise von Informatiklehrerinnen gibt Aufschluss zu weiteren Gesichtspunkten, die im Forschungsprogramm berücksichtigt werden.

Im Zentrum der Überlegungen steht die Fachwissenschaft Informatik und ihre Bezüge zum Schulfach Informatik. Auf dieser Basis wird das Modulkonzept entwickelt. Es stellt einen konstruktiven Vorschlag für den Informatikunterricht für die allgemein bildende Sekundarstufe II dar und umfasst die Module

- Informatiksysteme verantwortlich nutzen,
- Elemente der theoretischen Informatik,
- informatische Modellierung.

Die Prüfung des Konzepts erfolgt in drei Richtungen:

1. Durch exemplarische Studien bezüglich des Bildes der Informatik bei Schülerinnen wird eine Faszette des Forschungsgegenstandes näher ausgeleuchtet.
2. Das Konzept wird darüber hinaus durch die Anwendung zur Analyse verpflichtenden Informatikunterrichts in der Sekundarstufe I und
3. als gestaltungsleitender Konstruktionshintergrund zur Erstellung von Informatiksystemen als Lernhilfen für den Informatikunterricht geprüft.

Es wird gezeigt, dass die entwickelte Grundlage nicht nur als konstruktionsleitender Hintergrund für den Informatikunterricht geeignet ist, sondern darüber hinaus sowohl zur Analyse des Informatikunterrichts beitragen und nicht zuletzt Hilfestellung bei den Anforderungen zur Gestaltung von Informatiksystemen für den Einsatz im Unterricht bieten kann. Als Folgerung aus der erfolgreichen Prüfung des vorgestellten Konzepts wird eine Reihe von offenen Fragestellungen ausgewiesen, die über den Informatikunterricht in der allgemein bildenden Sekundarstufe II hinausweisen.

## Abstract

The modular approach as presented in this paper is a viable basis for teaching and assessing informatics in secondary schools. The complex relations between the general theory of science and informatics help to give the modular approach a valid scientific foundation. The subject matter is further elucidated by empirical data and research taking into consideration the learners' point of view. A closer look at the history teaching informatics at secondary level makes it possible to include the current discussion of how to teach informatics in this research program. A second source is the set of reports based on the informatics teaching experience of members of staff. They are included in this project, too.

Hence the focus of this paper is on the relationship between informatics at university and secondary school level. It is the basis for the modular approach. The modular approach is a practical suggestion for teaching IT to advanced students at secondary schools. It is comprised of three modules

- the responsible use of informatics resources
- elements of theoretical informatics
- modelling informatics systems

This modular approach is assessed in three different ways:

1. One aspect of the subject matter is the students image of informatics. This is illustrated and exemplified by a sample.
2. A second aspect is the issue of introducing informatics as a requirement in years 5-10 at secondary schools.
3. Finally by developing guiding principles as teaching aid for informatics in secondary schools.

The objective of this paper is threefold. It argues that the modular approach is a viable background for teaching informatics. It also provides a framework for planning and assessing informatics lessons as well as integrating informatics resources for classroom use. The successful verification of the modular approach opens the way for further questions and key issues that need to be looked at more carefully in future research. They are of a more general nature and go well beyond teaching informatics at secondary school level.

Kaum zwei, drei Jahrzehnte später scheint das Spiel unter den Musikstudenten an Beliebtheit eingebüßt zu haben, dafür aber von den Mathematikern übernommen worden zu sein, und lange Zeit blieb das ein kennzeichnender Zug in der Geschichte des Spiels, daß es stets von derjenigen Wissenschaft bevorzugt und benutzt und weitergebildet wurde, welche jeweils eine besondere Blüte oder Renaissance erlebte.

---

[Hesse 1943, S. 31]

## Einleitung – Übersicht und Struktur

- 1 Ausgangsüberlegungen
- 2 Wissenschaftstheorie und Informatik
- 3 Lerntheoretische Grundlagen
- 4 Stand der Schulinformatik
- 5 Lehrexpertise zum Informatikunterricht
- 6 Das Modulkonzept
- 7 Einschätzung der Informatik durch Lernende
- 8 Evaluation des Informatikunterrichts
- 9 Informatik als Gegenstand und Mittel des Lernens
- 10 Zusammenfassung
- 11 Offene Fragen

# Inhaltsverzeichnis

<b>Einleitung – Übersicht und Struktur</b> . . . . .	<b>1</b>
<b>1 Ausgangsüberlegungen</b> . . . . .	<b>3</b>
<b>2 Wissenschaftstheorie und Informatik</b> . . . . .	<b>9</b>
2.1 Gegenstände der Informatik . . . . .	10
2.2 Methoden der Informatik . . . . .	13
2.3 Bewertung von Definitionen von Informatik . . . . .	16
2.4 Konzepte in der Informatik . . . . .	20
<b>3 Lerntheoretische Grundlagen</b> . . . . .	<b>31</b>
3.1 Didaktische Grundorientierungen . . . . .	37
3.2 Unterrichtskonzepte – Prinzipien methodischen Handelns . . . . .	41
3.3 Anwendung der Erkenntnisse der Lerntheorie . . . . .	51
<b>4 Stand der Schulinformatik</b> . . . . .	<b>55</b>
4.1 Fachdidaktische Empfehlungen der Informatik . . . . .	55
4.2 Didaktik der Informatik für Schulen . . . . .	62
4.2.1 Entwicklung in der Bundesrepublik Deutschland . . . . .	62
4.2.2 Internationale Diskussion . . . . .	73
4.3 Problemlösen – die zentrale Kategorie des Informatikunterrichts . . . . .	79
4.4 Besondere Zieldimensionen des Informatikunterrichts . . . . .	81
<b>5 Lehrexpertise zum Informatikunterricht</b> . . . . .	<b>83</b>
5.1 Untersuchungsgestaltung . . . . .	83
5.1.1 Ziele der Expertise . . . . .	83
5.1.2 Zur Auswahl der Forschungsmethode . . . . .	84
5.1.3 Auswahl der Expertinnen . . . . .	85
5.1.4 Phasierung der Interviews . . . . .	85
5.1.5 Ausgestaltung – Konkretisierung der Fragen . . . . .	86

5.2	Ergebnisse . . . . .	87
5.2.1	Daten und Fragen zur Person . . . . .	87
5.2.2	Konzepte des Informatikunterrichts . . . . .	92
5.2.3	Methodisch-didaktische Aspekte . . . . .	96
5.2.4	[Software-]Technische Unterstützung des Informatikunterrichts . . . . .	97
5.3	Schlussfolgerungen . . . . .	100
<b>6</b>	<b>Das Modulkonzept . . . . .</b>	<b>103</b>
6.1	Gestaltungsanforderungen . . . . .	103
6.2	Entwicklung des Modulkonzepts . . . . .	105
6.2.1	Vorstellung der Module . . . . .	106
6.2.2	Konkretisierung – Beispiele zur Umsetzung . . . . .	110
6.3	Umsetzungsvorschläge . . . . .	111
<b>7</b>	<b>Einschätzung der Informatik durch Lernende . . . . .</b>	<b>115</b>
7.1	Erste Untersuchung – Einzelfallstudie . . . . .	116
7.2	Bild der Wissenschaft Informatik bei Schülerinnen . . . . .	117
7.3	Schülergruppeninterview . . . . .	120
7.4	Bild der Wissenschaft Informatik bei Schülerinnen – Längsschnitt . . . . .	121
7.5	Auswertung und Zusammenfassung der Ergebnisse . . . . .	123
7.6	Überlegungen zur Vermittlung des Bildes der Informatik . . . . .	125
<b>8</b>	<b>Evaluation des Informatikunterrichts . . . . .</b>	<b>127</b>
8.1	Einordnung in den Forschungsgang . . . . .	127
8.2	Evaluation – zum Begriff . . . . .	128
8.3	Begleitende Untersuchungen zur Einführung des Pflichtfachs Informatik in Bayern . . . . .	128
8.4	Überlegungen zur Anwendung des Modulkonzepts in der Sekundarstufe I . . . . .	134
<b>9</b>	<b>Informatik als Gegenstand und Mittel des Lernens . . . . .</b>	<b>137</b>
9.1	Beitrag der Fachdidaktik zur Nutzung von Informatiksystemen für die Lehr-/Lernprozessgestaltung . . . . .	139
9.2	Gestaltungsanforderungen . . . . .	139
9.3	Beispielhafte Umsetzung ausgewählter Anforderungen . . . . .	143
9.4	Überlegungen zur Gestaltung von Informatiksystemen als Lernmittel für den Informatikunterricht . . . . .	146
<b>10</b>	<b>Zusammenfassung . . . . .</b>	<b>149</b>
<b>11</b>	<b>Offene Fragen . . . . .</b>	<b>153</b>

<b>Anhänge</b>	<b>157</b>
<b>A Expertise durch Interviews – Übersicht und Ergebnisse</b>	<b>161</b>
A.1 Anfrage mit der Bitte um Mitarbeit	161
A.2 Leitfaden für das Expertinneninterview	162
A.2.1 Abschnitt I – Daten und Fragen zur Person	162
A.2.2 Abschnitt II – Interviewthesen	162
A.2.3 Abschnitt III – Konzepte	163
A.2.4 Abschnitt IV – Methodisch-didaktische Aspekte	163
A.2.5 Abschnitt V – [Software-] Technische Unterstützung	163
A.3 Auswertungen	164
A.3.1 Abschnitt I – Daten und Fragen zur Person	164
A.3.2 Abschnitt II – Interviewthesen	165
A.3.3 Abschnitt III – Konzepte	165
A.3.4 Abschnitt IV – Methodisch-didaktische Aspekte	166
A.3.5 Abschnitt V – [Software-] Technische Unterstützung	166
<b>B Befragung zur Prädikativen Modellierung</b>	<b>167</b>
B.1 Dokumentation der Fragen	167
B.2 Dokumentation der Ergebnisse	167
<b>C Bild der Schülerinnen von der Informatik</b>	<b>171</b>
C.1 Dokumentation des Fragebogens	171
C.2 Dokumentation der Ergebnisse	174
<b>D Gruppeninterview</b>	<b>179</b>
D.1 Leitfaden	179
D.2 Transkriptausschnitte	180
<b>E Hospitationsprotokolle zur Evaluation</b>	<b>187</b>
E.1 Ordnerbäume – objektorientierte Beschreibung	187
E.2 Textverarbeitung – objektorientiert betrachtet	189
E.3 E-Mail – objektorientierte Beschreibung	191
E.4 Verweisstruktur – objektorientiert beschreiben und modellieren	193
E.5 E-Mail – Aufbau der Adresse, Transport im Netz	195

<b>F Exemplarische Unterrichtseinheiten</b> . . . . .	<b>197</b>
F.1 Informatik in der Sekundarstufe II: Automatentheorie . . . . .	197
F.2 Gestaltung eines Grundkurses Informatik . . . . .	210
F.3 Ausgewählte Unterrichtsplanung und -durchführung . . . . .	213
F.3.1 Objektorientierte Modellierung am Beispiel des Schülercomputerarbeitsplatzes . . . . .	213
F.3.2 Modellierung grundlegender Elemente für ein „Blinklicht“ . . . . .	214
F.3.3 Einführung in die Arbeit mit einem webbasierten CSCW zur Kursplanung . . . . .	215
F.3.4 Das Fachkonzept Keller - Realisierung als Klasse . . . . .	217
F.3.5 Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient . . . . .	220
<b>G Schemata zum Problemlösen</b> . . . . .	<b>225</b>
G.1 Imperativ – strukturiertes Problemlösen . . . . .	225
G.2 Objektorientierte Modellierung . . . . .	226
<b>Verzeichnisse, Listen, Index</b>	<b>227</b>
<b>Literaturverzeichnis</b> . . . . .	<b>229</b>
<b>Tabellenverzeichnis</b> . . . . .	<b>259</b>
<b>Abbildungsverzeichnis</b> . . . . .	<b>262</b>
<b>Liste der Algorithmen</b> . . . . .	<b>264</b>
<b>Namens- und Organisationsindex</b> . . . . .	<b>267</b>
<b>Sachindex</b> . . . . .	<b>270</b>
<b>Abkürzungsverzeichnis</b> . . . . .	<b>283</b>

# Einleitung – Übersicht und Struktur

Mit dieser Forschungsarbeit soll die Schulinformatik<sup>1</sup> auf eine wissenschaftlich begründete und abgesicherte Basis gestellt werden. Diese Fundierung der Schulinformatik auf dem Hintergrund der Fachwissenschaft Informatik leistet einen notwendigen Beitrag zur Weiterentwicklung der Didaktik der Informatik. Im 1. Kapitel wird die Wahl des Forschungsgegenstandes begründet, um eine erste Annäherung an das Forschungsziel und methodische Vorüberlegungen darzustellen.

Das 2. Kapitel dokumentiert Innensichten, die im Laufe der Zeit von Informatikerinnen<sup>2</sup> über ihren Gegenstandsbereich entwickelt, verfeinert und kontrovers diskutiert wurden.

Im 3. Kapitel werden Ergebnisse aktueller lerntheoretischer Forschungen für schulbezogene Zusammenhänge ausgewählt und auf ihre Eignung im Forschungskontext untersucht.

Um den Stand der Schulinformatik einordnen und bewerten zu können, werden im 4. Kapitel Ergebnisse der Schulinformatik mit Hilfe der entwickelten theoretische Grundlage einer Prüfung unterzogen.

Die Sicht auf aktuell wirksame Faktoren in der Schulinformatik wird durch die Auswertung einer Expertise im 5. Kapitel vorgenommen. Die Expertise umfasst einen explorativen Teil, der die Spannweite und die Diversifikation – aber auch die Gemeinsamkeiten in den grundlegenden Konzepten – von Informatiklehrkräften verdeutlicht. Die dokumentierten Studien wurden durchgeführt, um eine möglichst große Bandbreite der Anforderungen, Positionen und Einstellungen von Experten im Forschungsgefüge zu berücksichtigen.

Diese Vorüberlegungen führen zu dem Modulkonzept zur Informatischen Bildung<sup>3</sup>. Die Entwicklung des Modulkonzepts als Möglichkeit, einen ausgewiesenen Kern der fachwissenschaftlichen Grundlagen der Schulinformatik unter lerntheoretischen Maßgaben und orientiert an den zu erwartenden Entwicklungen umzusetzen, erfolgt im 6. Kapitel. Damit die dargestellten Überlegungen im Kontext wirksam werden können, werden konzeptionelle Entscheidungen detailliert dargestellt, die später in einen evaluativen Kontext eingebunden werden.

Das vorgestellte Modulkonzept wird mittels Beobachtungen im Feld (d. h. in der Schule) einer ersten qualitativen Bewertung unterzogen. Die Ergebnisse der Studie zur Entwicklung und Veränderung der Sicht von Schülern auf die Informatik, die nach Maßgabe der im Modulkonzept dargestellten Leitideen unterrichtet wurden, werden im 7. Kapitel vorgestellt und hinsichtlich ihres Einflusses auf das Bild der Informatik ausgewertet.

Spezielle Elemente der Informatischen Bildung – insbesondere der informatischen Modellierung – werden in der Sekundarstufe I (ab 6. Jahrgang) unterrichtlich umgesetzt. Diese Umsetzung wird im 8. Kapitel auf

---

<sup>1</sup> Der Begriff Schulinformatik wird im Zusammenhang mit dem Fachgebiet „Didaktik der Informatik“ benutzt, um das Schulfach Informatik zu bezeichnen (vgl. [Arlt u. a. 1982, Titel: Einführung in die Schulinformatik], [Hubwieser 2000, S. 77f]). Da es in der Informatik unüblich ist, von wissenschaftlichen Schulen zu sprechen, wird die Gefahr einer Verwechslung oder gar Fehlinterpretation als sehr gering eingeschätzt.

<sup>2</sup> Redaktionelle Anmerkung:

Geschlechtsspezifische Zuweisung von Rollen durch die offizielle Bezeichnung beider Geschlechter in einem langen Text wirken ermüdend und machen einen Text schwer lesbar. Auswege aus dieser misslichen Situation stellen geschlechtsneutrale Bezeichnungen (wie *Lehrkraft* statt *Lehrerinnen und Lehrer*) dar oder die von mir in diesem Text vorgenommene Variante, ein Geschlecht zu benutzen und die Leser darauf hinzuweisen, dass das je andere Geschlecht sich nicht missachtet fühlen mag.

<sup>3</sup> *Informatische Bildung* wird in der vorliegenden Arbeit als Bezeichnung verwendet.

der Grundlage der entwickelten Kriterien an Hand konkreten Unterrichts einer ersten Evaluation unterzogen.

Die Janusköpfigkeit der Wissenschaft Informatik – einerseits Gegenstand der Erkenntnis und andererseits Mittel zur Erkenntnisgewinnung im Informatikunterricht – führt zur Frage der informatischen Gestaltung von Lernhilfen. Im 9. Kapitel wird gezeigt, wie die Gestaltungsfähigkeit der Informatik zur Unterstützung unterrichtlicher Prozesse so nutzbar gemacht wird, dass zu erwarten steht, dass die Vermittlung abstrakter Konzepte konkret durch Unterrichtsmittel, die ihrerseits Informatiksysteme<sup>4</sup> darstellen, besser gelingen kann. Exemplarisch wird die Unterstützung ausgewählter Elemente von Lehr-/Lernprozessen durch Informatiksysteme vorgestellt, die unter fachdidaktischen Maßgaben/Kriterien entwickelt wurden.

Die zusammenfassende Darstellung der Ergebnisse in einer Übersicht, die im 10. Kapitel vorgenommen wird, soll sowohl Antworten auf die Forschungsfragen liefern, weist aber darüber hinaus auf bisher nicht zufriedenstellend geklärte Fragen. Die Fachwissenschaft Informatik kann nicht umhin, die ihr in den modernen, entwickelten Gesellschaften zugewachsene Rolle aufzunehmen und damit ihre Gestaltungsfähigkeit/-verpflichtung für weitere Felder einzulösen. Dies wird nicht ohne Reibungsverluste von Statten gehen.

Im 11. Kapitel werden Hinweise auf notwendige weitere Forschungsvorhaben zur Didaktik der Informatik, die sich ausdrücklich als Teil der Informatik versteht, vorgestellt.

---

<sup>4</sup> Informatiksystem wird in der vorliegenden Arbeit als eine spezifische Zusammenstellung von Hardware, Software und Netzverbindungen zur Lösung eines Anwendungsproblems verstanden (vgl. [Claus und Schwill 2001, S. 301 – Stichwort: Informatiksystem]).

Es gibt wohl in keinem Bereich ähnlich verzerrte Vorstellungen über die Berufsbilder wie in der Informatik. In der breiten Öffentlichkeit werden mit dem Berufsbild Informatiker oder Softwareentwickler immer noch technikbegeisterte Programmierer verstanden. Unter diesem Eindruck kann es nicht verwundern, dass insbesondere Frauen sich nur sehr begrenzt für ein derartiges Studium interessieren. [...] Grundlagen müssen bereits in den Schulen flächendeckend vermittelt werden.

[BMBF 2000, S. 187]

## Kapitel 1

# Ausgangsüberlegungen

### Feststellungen zum Problemfeld – Motivation

Ab 1972 wird das Schulfach Informatik in der allgemein bildenden gymnasialen Oberstufe in den (alten) Bundesländern der Bundesrepublik Deutschland angeboten. Dies ist ein Ergebnis der Oberstufenreform. Sie erweiterte die Wahlmöglichkeiten für die Schülerinnen. In einigen Schulen Nordrhein-Westfalens wurde das Schulfach Informatik in der gymnasiale Oberstufe bereits ab 1969 unterrichtet. Im Wesentlichen wurde ein sehr pragmatischer Zugang zu dem neuen Unterrichtsfach Informatik in der Schule umgesetzt. Die Schulinformatik orientierte sich am damaligen Entwicklungsstand der Fachwissenschaft. Ansätze zur Unterrichtsgestaltung, die die Inhalte und Methoden der zugrunde liegenden Fachwissenschaft für schulische Lehr-/Lernprozesse kopieren, werden als Abbilddidaktik bezeichnet.<sup>5</sup> Sie wird von den Fachdidaktiken negativ annotiert, da für Bildungsprozesse im allgemein bildenden Schulwesen nicht unreflektiert didaktische Ansätze, die für berufliche Anforderungen erfolgreich qualifizieren, verwendet werden dürfen. Die Orientierung einer fachwissenschaftlichen Ausbildung unterscheidet sich bezogen auf die Zielsetzung grundlegend von den Erfordernissen einer allgemeinen Bildung.

Die zunehmende Durchdringung aller Lebens- und Arbeitsbereiche mit Informatiksystemen führte zu neuen gesellschaftlichen Anforderungen. Mit der Vermittlung informatischer Inhalte und Methoden im allgemein bildenden Unterricht verbundene Fragestellungen waren in der Anfangsphase des Informatikunterrichts nicht theoriegeleitet fachdidaktisch aufgearbeitet. Durch die Unterrichtspraxis wurde bei Lehrerinnen Erfahrungswissen zu den Vermittlungsprozessen aufgebaut, das als empirisches Wissen und über Reflexionsprozesse i. d. R. personal gebunden, beispielsweise über Fortbildungen und regionale/lokale Austausch von Lehrerinnen untereinander reflektiert und weitergegeben wurde.

Die Zielsetzung bei der Einführung des Schulfaches Informatik bestand und besteht in der Einbindung informatischer Kompetenzen in die allgemeine Bildung. Um diese Kompetenzen für ihre Funktion im allgemein bildenden Sinn gewichten zu können, bedarf es anerkannter Qualitätskriterien, da andernfalls die oben beschriebene Gefahr einer Abbilddidaktik besteht. Untersuchungen anderer Fachdidaktiken zeigen, dass der Prozess der Vermittlung allgemein bildender Elemente der jeweiligen Bezugswissenschaft nicht ohne weitere Anstrengungen gelingt. Zu den Qualitätskriterien, mit denen Entscheidungen über den allgemein bildenden Charakter von Fachinhalten getroffen werden können, wird beispielsweise von FISCHER, einem anerkannten Vertreter der Fachdidaktik Mathematik festgestellt: „Was nicht prinzipiell auch einem Volksschüler vermittelt werden kann, kann keine fundamentale Idee sein“ [Fischer 1984, zitiert nach [Schwill 1993, S. 21] ]. Jerome S. BRUNER formuliert allgemeiner: „Jedes Kind kann auf jeder Entwicklungsstufe jeder Lehrgegenstand in einer intellektuell ehrlichen Form gelehrt werden“ [Bruner 1980, S. 44].

<sup>5</sup> Zur Charakterisierung des Begriffs „Abbilddidaktik“ im Kontext der Mathematikdidaktik siehe [Wittmann und Müller 1988] mit Bezug auf [Andelfinger und Voigt 1986, S. 3]. Hilbert MEYER verdeutlicht durch die Diskussion der These: „Der Inhalt bestimmt die Methode(n)“ [Meyer 1988, S. 72ff], welche allgemeinen Probleme mit einer „Abbilddidaktik“ verbunden sind.

Für die Fachdidaktik Informatik wird 1993 von Andreas SCHWILL mit dem Ansatz der „Fundamentalen Ideen der Informatik“ [Schwill 1993] der Versuch unternommen, die Prüfung ausgewählter Fachinhalte für den Informatikunterricht zu ermöglichen. Dabei wurde das 1960<sup>6</sup> von BRUNER fachneutral formulierte Prinzip der Orientierung des Lernens an Fundamentalen Ideen [Bruner 1960, Bruner 1980] gewählt. In der Argumentation zieht SCHWILL darüber hinaus ausgewählte fachdidaktische Erkenntnisse anderer Fächer (insbesondere der Mathematik) heran. Mittels Kriterien, die SCHWILL ausgehend von diesen Ergebnissen diskutiert, werden für gegebene Inhalte Möglichkeiten zur Prüfung bereitgestellt. Die von SCHWILL präsentierten Fundamentalen Ideen lassen allerdings wichtige Bereiche der Informatik außer Acht: hier seien die *Entwicklung von soziotechnischen Systemen*, aber auch das *Konzept der partizipativen Softwareentwicklung* genannt. Damit ist ein grundlegendes Problem dieses Ansatzes, nämlich die Auswahl der zu prüfenden Fachinhalte, nicht befriedigend gelöst, da die Auflösung der Frage nach den allgemein bildenden Fachinhalten und -methoden, sowie die Frage der Vermittlung im Informatikunterricht weiter offen bleiben.

Einige Probleme des Informatikunterrichts an allgemein bildenden Schulen können aus Sicht der Fachdidaktik auf einer empirischen Erfahrungsbasis folgendermaßen charakterisiert werden: Nicht nur für die erste Zeit, in der ohne eine eigene theoretisch fundierte Fachdidaktik Informatik unterrichtet wurde, sondern bis heute kann festgestellt werden, dass die Lehrenden typischerweise von einer relativ schmalen fachlichen Basis ausgehen, da sie mehrheitlich nicht primär für das Fach Informatik<sup>7</sup> qualifiziert wurden. Zudem fehlen spezielle fachdidaktische Kenntnisse aus der Informatik. Damit befinden sich die Lehrerinnen in der Situation, einen Transfer von ihrem primären Lehramt auf das Lehramt für das allgemein bildende Schulfach Informatik zu leisten. Dieser Transfer kann so weder fachdidaktisch noch fachwissenschaftlich reflektiert werden. In den zurückliegenden 30 Jahren wurden verschiedene – sich zum Teil widersprechende – Vorschläge zur Umsetzung informatischer Fachkonzepte empfohlen, die Eingang in die Unterrichtspraxis des Schulfaches Informatik fanden. Viele dieser Vorschläge der fachdidaktischen Diskussion beruhen auf der Kenntnis ausgewählter Elemente der Fachwissenschaft, wobei angemerkt werden muss, dass die Auswahl nicht sorgfältig begründet und damit transparent stattfindet. Dies führt im schlimmsten Fall zu einer zufälligen Auswahl und damit dem Vorwurf einer „Beliebigkeit der Inhalte“. Der Gestaltung der konkreten Lernorte<sup>8</sup> wurde in der zurückliegenden Zeit nicht genug Aufmerksamkeit gewidmet. Damit konnten Möglichkeiten zur fachdidaktisch notwendigen Gestaltung lernförderlicher Szenarios nicht berücksichtigt werden.

Für ausgewählte Fragen konnten fachdidaktische Mittel für den Vermittlungsprozess, die für die Entwicklung einer Fachdidaktik eine grundlegende Orientierung ermöglichen, entwickelt werden. Das grundlegende Vermittlungsproblem wurde allerdings bereits sehr früh durch Auseinandersetzungen über „die richtige [Programmier-]Sprache“, grundlegende methodische Fragen zur Umsetzung im Kontext der Programmierung und den aus fachdidaktischen Gesichtspunkten zu wählenden Informatiksystemen (hier vor allem der Betriebssystemstreit) überdeckt. Konsequenzen dieser Diskussionsprozesse: im Bundesland Thüringen (Zentralabitur) wird beispielsweise die Nutzung der Programmiersprachen Pascal (oder Oberon) und Prolog für die Vermittlung grundlegender Implementierungen verlangt.<sup>9</sup> Auf der anderen Seite finden sich in Richtlinien und Lehrplänen für das Schulfach Informatik ausgewiesene Projektphasen, wobei konstatiert werden kann, dass solche ausgewiesenen Projektphasen von anderen Fachdidaktiken zwar gefordert werden, aber nicht durchgängig bis auf die Ebene der Lehrplanentwicklung ihren Niederschlag gefunden haben.

<sup>6</sup> Die Suche nach zentralen, fachtypischen „Fragestellungen, Bezeichnungsweisen, Strukturierungs- und Begründungsformen“ im Kontext der Entwicklung eines Mathematik Curriculums wurde bereits 1913 von Alfred N. WHITEHEAD angegangen [Whitehead 1962, S. 261].

<sup>7</sup> Ein Großteil der Lehrerinnen, die Informatikunterricht erteilen, sind für das Lehramt in Mathematik und/oder Physik qualifiziert. Darüber hinaus ist festzustellen, dass der Unterricht im Schulfach Informatik in der Sekundarstufe I bundesweit zu großen Teilen fachfremd unterrichtet wird.

<sup>8</sup> Informatikraum, Informatiklabor, Computerraum, Computerfachraum oder auch Computerkabinett genannt, je nach Bundesland/Region spezifisch ausgeprägt/bezeichnet – dabei handelt es sich um den Fachraum für den Unterricht im Schulfach Informatik. Für Ausstattungsfragen wurden regelmäßig Empfehlungen erstellt (siehe exemplarisch [LSW 1991, LSW 1993, LSW 1995c] – nicht aber [LSW 1995b, auf S. 59 findet sich ein Muster für die Ausgestaltung des Raumes, das in den anderen Vorschlägen (leider) nicht erwähnt wird] und [Vaupel und Hoffmann 2001], da letztere für Multimediaausstattungen entwickelt wurde).

<sup>9</sup> „Die Programmiersprachen PASCAL, OBERON und PROLOG sind in diesem Lehrplan verbindlich festgelegt. Diese Festlegung hat vorläufigen Charakter. [...] Die Einschränkung auf konkrete Programmiersprachen begründet sich durch die zentral gestellten Prüfungsaufgaben (schriftliches Zentralabitur)“ [Kultusministerium Thüringen 1999, S. 14].

In beiden Forschungsfeldern sind in den letzten Jahren Entwicklungen zu verzeichnen, die von der weiter zu entwickelnden Fachdidaktik Informatik produktiv aufgenommen und konstruktiv diskutiert werden müssen. Exemplarisch wird dies an der Klärung des Begriffs Informatiksystem aus der Fachsystematik verdeutlicht:

Heute ist Informatik u. a. gekennzeichnet durch die objektorientierte Konstruktion von Informatiksystemen. Der Begriff Informatiksystem wird zur Bezeichnung der Einheit von Hard- und Software unter Betonung ihrer interaktiven Eigenschaften verwendet. Informatiksysteme werden zunehmend unter stärkerer Berücksichtigung verteilter und vernetzter Systeme gestaltet.

Die beiden Aspekte objektorientierte Modellierung und verteilte, vernetzte Systeme stellen eine Umorientierung des informatischen Wissenschaftsverständnisses dar, die als Paradigmenwechsel (vgl. [Kuhn 1969]) bezeichnet werden kann. Für den Begriff »Paradigma« soll die „Definition“ von [Kuhn 1969, S. 10] als Grundlage herangezogen werden: „Darunter verstehe ich allgemein anerkannte wissenschaftliche Leistungen, die für eine gewisse Zeit einer Gemeinschaft von Fachleuten maßgebende Probleme und Lösungen liefern.“ Dies ist im Kontext dieser Arbeit insoweit bedeutsam, da der Begriff Paradigma häufig – in unterschiedlichen Zusammenhängen verwendet wird. Der Paradigmenwechsel der Informatik muss auch zu einer Neuorientierung der Ansätze der Didaktik der Informatik führen. Bevor die Frage nach der Auswahl und Umsetzung der dazu notwendigen Fachkonzepte beantwortet werden kann, ist die Zielsetzung der Informatischen Bildung in der sogenannten Informations- beziehungsweise Wissensgesellschaft zu analysieren.

Hier reichen die traditionellen Studien zur Allgemeinbildung der Informatik, die vor ca. 20 Jahren entwickelt wurden, nicht mehr aus (vgl. [Balzert 1977a, Claus 1977, Menzel 1984]), da sie den Paradigmenwechsel noch nicht berücksichtigen konnten. Der allgemein bildende Charakter informatischer Konzepte muss nach dem Paradigmenwechsel erneut untersucht, geprüft und zu Gestaltungsvorschlägen verdichtet werden, die nachhaltigen Informatikunterricht ermöglichen. Die grundlegende wissenschaftliche Fundierung der Schulinformatik im Kontext der Fachwissenschaft Informatik stellt damit einen notwendigen Beitrag zur Weiterentwicklung der Didaktik der Informatik dar.

Damit konkretisiert sich der hier diskutierte Forschungsgegenstand. Die der Schulinformatik zugrunde liegenden fachlichen und methodischen Dimensionen sind zu strukturieren. Dies hat Konsequenzen für den konkreten Unterricht im Schulfach Informatik. Mit dieser Arbeit soll eine umfassende, theoretische Strukturierung der Schulinformatik geleistet werden.

Von den Ergebnissen wird erwartet, dass sie eine wissenschaftliche Fundierung der Fachdidaktik als Teil der Fachwissenschaft liefern. Primär von der Fachwissenschaft Informatik ausgehende Impulse sollen auf ihre fachdidaktische Umsetzung untersucht werden. Das Ergebnis dieser Untersuchungen ist auf aktuell erfolgreiche, lerntheoretisch anerkannte Konzepte zu beziehen. Für die Gestaltung von Lehr-/Lernprozessen müssen – neben fachlicher Integrität – methodische Hinweise, die in der Tradition der fachwissenschaftlichen Entwicklung als zentrale und besondere Arbeitsweisen der Informatik erkannt wurden, Berücksichtigung finden.

Damit kann eine erste forschungsleitende Zielvorstellung expliziert werden:

#### **Forschungsziel** – erste Annäherung

Das Schulfach Informatik umfasst – bezogen auf die Bezugswissenschaft – zentrale allgemein bildende Elemente. Dies gilt sowohl auf der inhaltlichen Ebene, als auch auf der Ebene der durch die Fachwissenschaft maßgeblich zur Anwendung kommenden Methoden und Arbeitstechniken.

## Überlegungen zur Forschungsmethodik – Forschungsansätze

Damit eine dem aktuellen Stand der Fachwissenschaft angemessene Fachdidaktik Informatik verankert werden kann, ist es notwendig, in einem ersten Schritt die wissenschaftstheoretische Fundierung der Informatik zu untersuchen. Dabei werden als Elemente der Fachwissenschaft hier die grundlegenden Prinzipien und Methoden verstanden, die die Informatik zum jetzigen Zeitpunkt ausmachen. Es sind theoriegeleitet Ergebnisse und Ansätze zu prüfen, um die diesem Bereich zukommende Theoriebildung angemessen berücksichtigen zu können. Als Ergebnis sollen Hypothesen formuliert werden, die in den folgenden Teilen der Forschungsarbeit näher zu beleuchten sind.

Die Entwicklung einer fundierten Fachdidaktik besteht in der Verschränkung der fachlichen und der fachmethodischen Ebenen mit der lerntheoretischen Ebene. Dabei wird nicht angestrebt, die Entwicklung der für diesen Bereich maßgeblichen Theorien in ihrer Komplexität und historischen Tiefe nachzuzeichnen, sondern eine begründete Auswahl der als konstitutiv erkannten Theorieelemente für das Konzept heranzuziehen. Gerade an Schnittstellen fachlicher Methoden und lerntheoretischer Konstrukte können Querbezüge die Verbindung der Ebenen zu einer fachdidaktischen Theoriebildung unterstützen. Daher gilt es Kriterien herauszuarbeiten, mit denen Kandidaten für erfolgreiche Verbindungen zwischen der Fachwissenschaft und den Ergebnissen der Lerntheorie und ihre Abhängigkeiten ausgewählt werden. Die Sichten auf Schülerinnen, Lehrerinnen, Medien und Lernmittel sollen dabei Berücksichtigung finden. Gefundene Ergebnisse werden zu Hypothesen verdichtet.

Auf dem Hintergrund einer kritischen Analyse aktueller Ergebnisse der Schulinformatik werden die formulierten Hypothesen geprüft. Es gilt herauszuarbeiten, in wie fern die Hypothesen als Werkzeug zur Evaluation des Informatikunterrichts eingesetzt werden können.

Die Situation der Schulinformatik zu Beginn des Forschungsvorhabens wird mit Bezügen zu der Entwicklung dieses Forschungszweiges in einer bewerteten Zusammenstellung dokumentiert. Das Spannungsfeld zwischen der sich systematisch darstellenden Didaktik der Informatik und dem pragmatisch orientierten konkreten Informatikunterricht kann als Prüfstein für die formulierten Hypothesen dienen. Diese theoriegeleiteten Ansätze sind auf die Praxis zu beziehen. Das kann im Rahmen dieser Arbeit nur mit Hilfe von Fallstudien realisiert werden, die es erlauben, sowohl Einschätzungen von Experten aber auch Möglichkeiten einer unterrichtlichen Umsetzung zu erkunden.

Eine breite, empirische Untersuchung des Informatikunterrichts ist bis heute nicht erfolgt. In den einleitenden Bemerkungen seiner Darstellung „Informatik in Schule und Ausbildung im internationalen Vergleich“ stellt Peter GORNY 1984 fest:

In einem knappen Referat soll ich Ihnen darstellen, wie die Bemühungen der Bundesrepublik im Bereich Informatik im internationalen Vergleich zu bewerten sind. Zur Bewältigung dieser Aufgabe gibt es im Prinzip zwei Wege: entweder man berichtet von einem umfangreichen Forschungsprogramm mit ausführlichen Umfragen und einer klassifizierenden/statistischen Auswertung der auffindbaren Literatur aller in die Untersuchung einbezogenen Länder. Leider hat bisher niemand ein solches Forschungsprojekt finanziert [...]. Bleibt nur der zweite Weg: der Verzicht auf die statistische Objektivität, der Verzicht auf präzise Definitionen und Ableitung der Schlussfolgerungen aus bewiesenen Tatbeständen. [...] Ich berichte Ihnen über meine – subjektiven Eindrücke von Tagungen, Ausschusssitzungen der GI, ACM, ATEE, AEDS und IFIP, [...] sowie von Fachartikeln und Einzelgesprächen der letzten drei Jahre [...].

[Gorny 1984]<sup>10</sup>

Diese Beschreibung ist auch im Jahr 2002 noch zutreffend. Bis heute wurden keine umfangreichen, empirisch gesicherten Aussagen zum Informatikunterricht in der Bundesrepublik erstellt. Eine solche Studie kann aus naheliegenden Gründen nicht als Bestandteil dieser Arbeit angefertigt werden.

Die in dieser Arbeit zu entwickelnde theoretische Basis kann nicht die von GORNY geforderten Studien ersetzen. Allerdings ist es möglich, die explizierten Hypothesen auf Fallstudien<sup>11</sup> zu beziehen. Eine

<sup>10</sup> Eine Dekodierung der Akronyme findet sich im Anhang (ab S. 283) dieser Arbeit.

<sup>11</sup> z. B. Interviews mit Expertinnen (Informatiklehrerinnen)

solche Expertise kann die Sicht auf aktuell wirksame Faktoren in der Schulinformatik aus der Sicht von erfolgreichen Lehrerinnen deutlich machen. Dazu sollte die Expertise einen explorativen Teil enthalten, der die Spannweite, aber auch die Gemeinsamkeiten in der Gemeinschaft der Informatiklehrerinnen verdeutlichen kann. Die Studien sollen darüber hinaus einen deskriptiven Charakter haben. Es wird erwartet, dass dadurch eine große Bandbreite der Anforderungen, Positionen und Einstellungen von Expertinnen im Forschungsgefüge berücksichtigt werden kann.

Auf der Basis der Hypothesen zur Wissenschaftstheorie und Informatik und lerntheoretischer Erkenntnisse, verbunden mit der Expertise erfahrener Informatiklehrerinnen können konzeptionelle Überlegungen expliziert werden, die eine konstruktive Grundlage für die Umsetzung darstellen. Die Verbindung zwischen den theoretischen und den empirischen Ergebnissen herzustellen, kann durch ein modulares Konzept geleistet werden. Ein solches Modulkonzept zur Informatischen Bildung ist zu entwickeln. Damit kann eine Brücke zwischen den theoretischen Ergebnissen und der Praxis des Informatikunterrichts gebildet werden. Das Modulkonzept soll die Möglichkeit eröffnen, einen ausgewiesenen Kern der fachwissenschaftlichen Grundlagen der Schulinformatik unter lerntheoretischen Gesichtspunkten zur Gestaltung unterrichtlicher Prozesse heranzuziehen. Für die konkreten Entwicklungsschritte zu einem solchen Konzept darf nicht außer acht gelassen werden, dass die Entwicklung Teil eines Prozesses ist, der sowohl auf der analytischen, aber auch auf der Umsetzungsebene ein Rückkopplungsprozess ist.

Um das theoriegeleitete Konzept auf die konkrete Praxis zu beziehen, soll eine exemplarische Umsetzung erfolgen, die untersucht wird. So werden mittels weiterer Fallstudien empirische Ansätze mit theoretischen Überlegungen verbunden, in dem durch Beobachtungen im Feld, d. h. in der Schule, eine erste qualitative Bewertung vorgenommen wird. Diese Rückbindung an die konkrete unterrichtliche Praxis kann ein erstes Element für den evaluativen Kontext darstellen.

Ein weiterer Prüfstein für die mit dem Konzept entwickelte Fundierung der Fachdidaktik bietet sich durch die Anwendung auf Informatikunterricht an, der ohne diese Grundlage geplant und durchgeführt wird. Dies erlaubt die Rückbindung der Gestaltungsvarianten des Informatikunterrichts auf das Modulkonzept. Darüber hinaus kann ein Vergleich bezüglich der unterrichtlichen Prozessgestaltung in die Evaluation einbezogen werden.

Als Ergebnisse der Konstruktionsbemühungen und der dargestellten Prüfung der Hypothesen werden konkrete Hinweise für Lehrerinnen, aber auch für Informatikexpertinnen erwartet.

Die Doppelrolle der Fachwissenschaft Informatik für schulische Lehr-/Lernprozesse, nämlich einerseits mit ihren eigenen Techniken Gegenstand der Erkenntnis, aber andererseits mit ihren Systemen auch Mittel der Erkenntnis im Informatikunterricht zu sein, führt häufig zu Missverständnissen. Dies gilt vor allem auf der Seite der Erwartungshaltung der Lernenden, wenn auch nicht verkannt werden darf, dass sich dies auch in der Rollenzuweisung an Informatiklehrerinnen widerspiegelt.

Aus den Fallstudien erwachsen Anforderungen an Lehr- und Lernmittel für den Informatikunterricht. Diese Dimension ist bezüglich der Fachdidaktik insofern bedeutsam, als durch häufige Änderungen auf der Gestaltungsebene von Benutzungsoberflächen die Gefahr besteht, keine erfolgreichen Gestaltungsprinzipien formulieren zu können, da diese bereits zum Veröffentlichungszeitpunkt „überholt“ sind. Allerdings ist gerade für grundlegende Elemente in einem „Baukasten“ von Lehrmitteln festzustellen, dass Gestaltungsanforderungen dazu führen können, dass die mit modernen Konzepten erstellten Hilfsmittel für den Unterricht „immer aktuell aussehen“, da sie das „Look & Feel“ der jeweils aktuellen Benutzungsoberfläche zu adaptieren in der Lage sind.

Gerade für ein modulares Konzept ist es unabdingbar, variable Module zu extrahieren, ihre Vor- und Nachbedingungen zu charakterisieren, damit es möglich wird, die Einsatzmöglichkeiten nicht sequenziell sondern versetzt zu praktizieren.

In der zusammenfassenden Darstellung der Ergebnisse sollen Gestaltungshinweise bezogen auf die Gestaltung von Lehrmitteln, aber auch zur Lehrerbildung Informatik ausgewiesen werden.



Nach grundlegenden Begriffen, nach vorrangigen Methoden und nach herausragenden Ergebnissen zu fragen und daran die Frage nach der Disziplin zu entscheiden, solchermaßen klassisch auch im Falle der Informatik vorzugehen, sei wohl nicht angemessen.

[Nake 2001, S. 10]

## Kapitel 2

# Wissenschaftstheorie und Informatik

Für den Arbeitsschwerpunkt einer wissenschaftlichen Fundierung der Schulinformatik wird eine Analyse des Wissenschaftscharakters der Informatik<sup>12</sup> vorgeschaltet. Wissenschaftsgeschichtlich bedeutsame Meilensteine werden vorgestellt, um die Gegenstände (Abschnitt 2.1) und Methoden (Abschnitt 2.2) der Informatik näher zu betrachten. Anschließend werden verschiedene Definitionen (Abschnitt 2.3) der Informatik im geschichtlichen Kontext auf die Schulinformatik bezogen und bewertet. Mit der Entwicklung von Basis-konzepten (Abschnitt 2.4) der Informatik wird für den Paradigmenwechsel in der Informatik (vgl. Kapitel 1, S. 5, [Quibeldey-Cirke 1994, S. 1-19]) der notwendige wissenschaftliche Hintergrund verdeutlicht.

Die Benutzung des Begriffs Informatik für das 1968 »einrichtende« Studienfach ist Ergebnis einer politischen Vorgabe.<sup>13</sup> Damit kommt dem Begriff Informatik zu dieser Zeit (noch) keine definitorische Klarheit zu. Im Laufe der Entwicklung der Wissenschaft Informatik im deutschen Sprachraum ist mehr als einmal der Versuch unternommen worden, die Gegenstände und Methoden „festzuschreiben“, mit denen sich die Informatik wissenschaftlich beschäftigt oder beschäftigen sollte.<sup>14</sup>

Der politische Hintergrund für die ersten Definitionen ist u. a. darin zu sehen, dass es notwendig schien, die Eigenständigkeit der Informatik gegenüber etablierten Fächern deutlich zu machen. Die Wissenschaften, die die Informatik hervorbrachten und aus denen sich die ersten Informatikerinnen rekrutierten, waren die Mathematik, die Elektrotechnik, aber auch die Physik. Treibende Kräfte lassen sich auch im Bereich der Wirtschaftswissenschaften nachweisen<sup>15</sup>.

Nur die Eigenständigkeit der Informatik schien eine erfolgreiche Einwerbung von Ressourcen für die Einrichtung dieses neuen Forschungsgebiets durch den Staat zu gewährleisten. Die frühen Definitionen können

<sup>12</sup> Informatik kann etymologisch aus den Begriffen *Information* und *Automatik* abgeleitet werden. Dann ist zu klären, wie die Grundbegriffe **Information** und **Automatik** definiert sind. *Informatik* beinhaltet allerdings auch den Begriff **Form**, mit der Erläuterung „Form ist der Schlüssel zu »Information«“ [Floyd 2001, S. 29].

<sup>13</sup> Am 22. Juli 1968 benutzt der Bundesforschungsminister STOLTENBERG in der Eröffnungsrede einer Tagung an der TU Berlin Informatik als Bezeichnung für ein neu einzurichtendes Studienfach. Im französischen Sprachgebrauch läßt sich der Begriff *Informatique* (als Kunstwort aus „Information“ und „Automatique“ / „Electronique“) ab 1962 nachweisen (vgl. [Bittner 2002, Folie 32]).

<sup>14</sup> Zur Geschichte der Verwendung des Begriffs Informatik im deutschen Sprachraum: „STEINBUCH berichtet in einem Referat von 1970, dass das Wort *Informatik* etwa im Jahre 1955 der Firma Standard Elektrik Lorenz AG geschützt und mit dem *Informatik-System Quelle* der Öffentlichkeit vorgestellt wurden“ [Alex 2002, S. 2, Fußnote 2]. Die erste publizierte Definition des Begriffs Informatik im deutschen Sprachraum stammt von Karl STEINBUCH: „Vor etwa zwanzig Jahren entdeckten Ingenieure in USA und Deutschland unabhängig voneinander, daß die Verfahren der Nachrichtentechnik auch für andere Aufgaben nützlich sind, Aufgaben, bei denen die Überwindung der räumlichen Entfernung ganz unwesentlich ist. Sie fanden, daß man mit elektrischen Schaltungen Zahlenrechnungen durchführen kann, und zwar mit einer Schnelligkeit, wie sie bis dahin einfach unvorstellbar war. Damit begann die automatische Informationsverarbeitung. Wir nennen sie «INFORMATIK»“ [Steinbuch 1957].

<sup>15</sup> Beispiele: TU Berlin [Bamberg 1998, S. 28], Universität Dortmund [Decker 1998, S. 32], Universität Karlsruhe [Braun und Fuchs 2001, 2.51]. Die Aktivitäten der Wirtschaftswissenschaften hatten kaum Einfluss auf die Orientierung der Ausgestaltung der universitären Forschung: „In der Aufbauphase der Informatik nach 1968 entstehen ganze zwei betriebsinformatische Forschergruppen (P. Mertens in Erlangen-Nürnberg und H. Wedekind in Darmstadt)“ [Bittner 2002, Folie 35].

damit als Abgrenzungsversuche gegenüber der Mathematik und Elektrotechnik verstanden und charakterisiert werden.<sup>16</sup> Zudem spiegelt sich in den Definitionen das Verständnis der Wissenschaftsauffassung. In [Städtler u. a. 1997, S. 16] wird die individuelle, subjektive wissenschaftliche Grundhaltung der Forscher zur Klärung der Frage herangezogen, ob der Mathematiker John von NEUMANN oder der (Bau-)Ingenieur Konrad ZUSE als „Vater“ der Grundprinzipien des modernen Computers angesehen werden soll/kann. BAUER weist ebenfalls auf diese Problematik hin, in dem er formuliert: „Möglicherweise steckt tiefes gegenseitiges Mißtrauen der Ingenieure (Eckert, Mauchly) und der Mathematiker (von Neumann, Goldstine) dahinter – was man auch anderswo beobachten konnte“ [Bauer 1998, S. 86, Fußnote 10]. BRAUER geht noch einen Schritt weiter und verweist in [Brauer 2001, S. 18f] gleich an vier Stellen darauf, dass (zumindestens in Deutschland) eine Kluft zwischen innovativen Ideen(gebern) und den etablierten Wissenschaftlern – vor allem der Elektrotechnik, aber auch der Mathematik – bestand und (seiner Meinung nach) auch weiterhin besteht:

- TURING, ZUSE, AIKEN : etwa 1936 – und alle keine Elektrotechniker
- ZUSEs Plankalkül, PETRIs Theorie der "Kommunikation mit Automaten": Auch diese stammten nicht von Elektrotechnikern.
- ZUSEs Plankalkül ist [...] für die deutschen Mathematiker und E-Techniker, [...] wohl nicht akzeptierbar [...].
- Er<sup>17</sup> wurde ebenfalls in Deutschland von den meisten der Mathematiker und E-Techniker [...] nicht verstanden.

[Brauer 2001, S. 18f – Zitatenkollage]

Alternative Definitionen der Wissenschaft Informatik<sup>18</sup>, die von den jeweiligen Protagonisten nicht durchgesetzt werden konnten, wurden lange Zeit – auch von kritischen Autoren – nicht beachtet. Seit einigen Jahren findet zunehmend eine Auseinandersetzung mit diesen „vergessenen Alternativen“ statt.<sup>19</sup>

## 2.1 Gegenstände der Informatik

### Information als zentraler aber mehrdimensionaler Begriff der Informatik

Die Frage nach der Wortbedeutung von Informatik führt (vgl. Fußnote 12, S. 9) zu dem Begriff Information. Um diesen Begriff zu definieren, kann in einer ersten Näherung die SHANNONSche<sup>20</sup> Informationstheorie herangezogen werden, die eine mathematische Theorie zur Bestimmung des „Informationsgehalts einer Nachricht“ enthält [Shannon 1948, deutsche Übersetzung in [Shannon und Weaver 1973]]. Sie liefert ein Maß für Information und ermöglicht die (mathematisch exakte) Bestimmung der minimalen Codierung, um mit einer Nachricht ein Maximum an „Gehalt“ übertragen zu können. Dieser Informationsbegriff hat sich für die Informatik nicht als durchgängig tragfähig erwiesen, da Information in dieser Theorie auf den Aspekt der Übertragung von Daten (oder Nachrichten) reduziert wird. Dies ist für die Informatik nur in Teilbereichen von Interesse. Weitere im Zusammenhang mit der Informatik bedeutsame Dimensionen des Begriffs Information gibt Christiane FLOYD an. Information ist

- personal, um Kognition allgemein und insbesondere die Interpretation von Daten durch Menschen zu kennzeichnen,
- organisationsbezogen, um die Rolle von Information bei Aktion und Entscheidungsfindung zu zeigen,

<sup>16</sup> Andererseits führten die in den Definitionen zum Ausdruck gebrachten Vorstellungen über Gegenstände und Methoden des neuen Studienfachs zu strukturellen Konsequenzen beim Aufbau und der Ausstattung.

<sup>17</sup> gemeint ist PETRI – der Autor

<sup>18</sup> Eine Darstellung von Definitionen findet sich im Abschnitt 2.3 (ab S. 16).

<sup>19</sup> vgl. [Krabbel und Kuhlmann 1994], [Coy 1997], [Eulenhöfer u. a. 1997a], [Brauer 2001], [Bittner 2002]

<sup>20</sup> benannt nach Claude E. SHANNON

- medial, um Informationen als eigenständiges, speicherbares und weitergebbares Gut zu betrachten.

[Floyd 2001, S. 43]

Es wird deutlich, dass im Kontext der Informatik mit Information nicht nur ein technisches Ziel, sondern auch Absichten (von Menschen) verbunden sein können. Genau diese lassen sich nicht angemessen formalisieren. Bis heute ist es den Informatikerinnen nicht gelungen, den grundlegenden Begriff Information für ihre Wissenschaft zu definieren (vgl. [Claus und Schwill 2001, S. 303f]).<sup>21</sup>

Die Mehrdimensionalität des Begriffs Information kann dazu führen, dass vermeidbare Missverständnisse auftreten. Um dem zu begegnen, kann es hilfreich sein, den Begriff zumindestens für den jeweiligen Kontext zu konkretisieren. Damit besteht die Möglichkeit, in Teilbereichen der Informatik eine Übereinkunft zu erzielen, die darin besteht, dass der Begriff Information eine gemeinsam definierte Kommunikationsbasis darstellt.<sup>22</sup>

## Geschichtliche Bestimmung der Gegenstände der Informatik

In der Frühzeit der Informatik steht das Bemühen, das technische Artefakt Computer<sup>23</sup> zu beherrschen und nutzen zu können, im Mittelpunkt der Aktivitäten. Der Technikgeneseforschung kommt die Aufgabe zu, die Entwicklung und Beschreibung dieser Artefakte im historisch-gesellschaftlichen Kontext zu untersuchen.<sup>24</sup> Dies ist nicht unsere Aufgabe. Exemplarisch soll hier auf den Entstehungskontext der bekannten Beschreibung der [technischen] Prinzipien des Computers hingewiesen werden. Es können Verbindungen<sup>25</sup> zwischen dem Mathematiker John von NEUMANN und dem Kybernetiker<sup>26</sup> Norbert WIENER in dem „First Draft of a Report on the EDVAC<sup>27</sup>“ [von Neumann 1945] nachgewiesen<sup>28</sup> werden (vgl. [Stach 1997, S. 5f]). Eine allgemeinere Darstellung der Mensch-Maschine Analogie im Kontext der frühen Entwicklung von Computern findet sich in [Eulenhöfer u. a. 1997b, S. 15ff]).

Der Ingenieur Konrad ZUSE konstruierte 1941 den weltweit ersten funktionsfähigen Computer [Zuse 1984]. Darüber hinaus entwickelt ZUSE 1945 die erste höhere Programmiersprache (Plankalkül)<sup>29</sup>. Die

<sup>21</sup> Eine Zusammenstellung von Versuchen, Information zu definieren, findet sich in [Keller 1998, S. 9-29; Von der wissenschaftlichen Begründung der „Informationsgesellschaft“].

<sup>22</sup> Möglicherweise ist dieser Anspruch nicht einzulösen. Francisco J. VARELA bezeichnet »Information« als „eine Art modernes Phlogiston [...] »Phlogiston« bezeichnete im 18. Jahrhundert eine Substanz, die die Phänomene der Verbrennung erklären sollte.“ [...] »Information« darf nicht als eine an sich gegebene Ordnung aufgefaßt werden, sie entsteht erst durch die kognitiven Tätigkeiten“ [Varela 1990, S. 18]. Im Zusammenhang mit Lerntheoretischen Grundlagen wird in Kapitel 3 die diesen Überlegungen zugrunde liegende konstruktivistische Position näher beleuchtet (vgl. S. 39f).

<sup>23</sup> „Warum ist denn der Computer ein so besonderes Artefakt? Das Computer-Artefakt ist symbolisch (sprachlich) wie technisch (materiell) verfasst. Im Artefakt werden diese zwei Weisen der Welterzeugung zusammengeführt“ [Bittner 2002, Folie 14].

Bis heute werden Computer[systeme] unter Informatikerinnen als „Rechner[systeme]“ bezeichnet. Inzwischen wird allerdings zunehmend der Begriff Informatiksystem gewählt, da diese Beschreibung dem (gewünschten) Zusammenspiel verschiedener Komponenten (und damit sowohl Hardware wie auch Software und immer häufiger auch vernetzten Strukturen) besser gerecht wird.

<sup>24</sup> Mit einem 2001 vorgelegten Titel [Naumann 2001] wird der Eindruck erweckt, „Die Geschichte der Informatik“ in Händen zu halten. Allerdings handelt es sich dabei um die Darstellung der Geschichte der „sichtbaren“ [Rechen-]Technik. Die Geschichte der Informatik läßt weiter auf sich warten.

<sup>25</sup> „In 1945, when von NEUMANN wrote the "First Draft", he and WIENER had been in contact for several years and had discussed their work and ideas quite often“ [Stach 1997, S. 6].

<sup>26</sup> **Kybernetik** – Wissenschaftszweig, der die Gesetzmäßigkeiten von technischen und biologischen Regelungs- und Steuerungsvorgängen erforscht und anwendet

vgl. [Hermann 1980]

<sup>27</sup> EDVAC – Electronic Discrete Variable Calculator

<sup>28</sup> Ergebnis der Kontakte zwischen WIENER und von NEUMANN zeigen sich beispielsweise in der Darstellung von Ähnlichkeiten zwischen Neuronen und digitalen Schaltelementen.

<sup>29</sup> Der Plankalkül umfasst die Kontrollstrukturen Verzweigung und Zyklus (auf Prädikaten), keinen Sprungbefehl und enthält verschiedene Datentypen. Im Jahre 2000 wird Plankalkül (erstmalig) implementiert: <http://www.zib.de/zuse/Inhalt/Programme/Plankalkuel/index.html> (vgl. [Bauer und Wössner 1972], [Gilo 1990], [Rojas u. a. 2000b] und [Rojas u. a. 2000a]). Im Zuge der Darstellung der Implementierung der Sprache bemerken [Rojas u. a. 2000b]: „Für Zuse war die Prädikatenlogik zunächst eine reine Beschreibungssprache. Schrittweise hat er jedoch eine mögliche Computerimplementierung konzipiert. Dafür wurde die ursprüngliche Notation mit imperativen Konstrukten ergänzt. Zuse hat dann zwischen der "impliziten" (prädikatenlogischen) und "expliziten" (imperativen) Form eines Programms unterschieden.“

dem sogenannten „von Neumann“-Prinzip zugrunde liegenden Ideen wurden bereits 1822 durch BAB-BAGE im Zusammenhang mit der Beschreibung der „Analytical Engine“ formuliert (vgl. [Hyman 1987, S. 362ff]). Folgerichtig wäre ihm das Prinzip zuzuschreiben (vgl. [Bauer 1998, S. 87f, Fußnote 23])<sup>30</sup>. Des Weiteren ist die Rolle von Ada LOVELACE als erster Programmiererin im Kontext dieser Konstruktion angemessen zu berücksichtigen (vgl. [Kim und Toole 1999]).

Zu Beginn der Entwicklung der Informatik kommt der konkreten Software eine untergeordnete Rolle zu. Da die mit der anfänglichen Unzuverlässigkeit der Hardware zusammenhängenden Probleme nach und nach verringert werden, kehrt sich das Verhältnis zwischen dem Aufwand für die Nutzung von Hardware und Software mit der Zeit um, so dass zunehmend den Methoden zur Erstellung von Software größere Aufmerksamkeit gewidmet wird.<sup>31</sup>

Im Laufe der Zeit werden (im Kontext der Orientierung an den vor allem in den USA „gemachten“ Trends) – ausgehend von realisierten von-Neumann-Maschinen – höhere Ebenen der Abstraktion für die Beschreibung von Algorithmen und Programmen durch Menschen entwickelt. Für eine Darstellung dieser Entwicklung siehe Abbildung 2.2 (S. 22). Die so entstandenen [Programmier-]Sprachen werden heute als von-Neumann-Sprachen bezeichnet. Frühe gegenläufige Ansätze, die aus Problemkontexten heraus die Formulierung von Problemlösungen unterstützen, werden nicht durch die Hersteller von Computern<sup>32</sup> unterstützt, so dass sie keine weite Verbreitung fanden (z. B.  $\lambda$ -Kalkül (Alonzo CHURCH, 1941)  $\rightarrow$  LISP (LIST Processing) (John MCCARTHY, ab 1958), Plankalkül (Konrad ZUSE, 1945)). Darüber hinaus spezifizierten erstmalig Informatiker für die Publikation und Kommunikation von Algorithmen ab 1959 mit ALGOL (ALGOrithmic Language) [Backus u. a. 1963] eine von-Neumann-Sprache. Die syntaktische Beschreibung wird durch eine Grammatik in Backus-Naur-Form (BNF) vorgelegt (vgl. [Backus 1959]). ALGOL wird zwar kaum von den Computer-Herstellerfirmen durch Compiler unterstützt, hat aber großen Einfluss auf die in der Folge entwickelten Programmiersprachen, die als ALGOL-Sprachfamilie bezeichnet wird. „Jedes Programm sollte ein publizierbares Produkt sein. Das unterscheidet sich grundsätzlich vom üblichen Ziel, daß das Programm läuft!“ [Reiser und Wirth 1994, S. XIX]

Bis heute werden Algorithmen in der informatikbezogenen Literatur häufig in Pseudocode dargestellt. Die Formulierung erfolgt dabei in einer Mischung aus syntaktischen Elementen der Sprachen ALGOL und Modula-2 [Wirth 1985] sowie nicht formalen, also beschreibenden Anteilen natürlichsprachlicher Konstrukte. Damit ist ein grundlegender (und unstrittiger) Gegenstand der Informatik benannt: Algorithmen und Datenstrukturen, ihre Abbildung in Informatiksysteme, sowie damit zusammenhängende Fragen der Theoriebildung.

<sup>30</sup> 1943/44 beschrieb John von NEUMANN die Architektur der Trennung von Speicher, Steuereinheit, Rechenwerk (Gleitkommaarithmetik), und Ein- und Ausgabeeinheiten für Rechner (Computer) – Konrad ZUSE hatte dieses Prinzip 1936 entworfen und 1938 mit der Z1 realisiert – allerdings wurde das Programm nicht in den Speicher geladen. „Es hat [...] in den U.S.A. Kopfschütteln gegeben, wenn ZUSE den Z3 als ersten „computer“ bezeichnete. Umgekehrt sollte man in Deutschland „speicherprogrammierter Rechner“ sagen, wenn ein solcher gemeint ist, und nicht „von Neumann Rechner“. Auch sollte man von einem „Computer“ auch in Deutschland nur sprechen, wenn er als universeller Rechner programmiert werden kann“ [Bauer 1998, S. 86]. Das diese Einschätzung nicht unumstritten ist, zeigt Horst ZUSE [Zuse 1999], in dem er die zeitgebundene Definition des Computers auf die frühen „Rechner“ bezieht. Es wird deutlich, dass zwischen den allgemeinen Prinzipien und ihrer konkreten Umsetzung eine zeitliche Lücke klafft, die in den Beschreibungen nicht deutlich wird. Der Unterschied betrifft die Art der Speicherung des Programms: die Ablage der Daten und des Programms in einem gemeinsamen Speicher und dadurch bedingt, die Änderungsmöglichkeit (z. B. Selbstmodifikation) des gespeicherten Programms zeichnen die von Neumann – Beschreibung aus, die aber so zu dem Veröffentlichungszeitpunkt nicht realisiert war. Weder ZUSES Maschinen Z1 bis Z4, noch die in den Vereinigten Staaten gebauten Modelle ABC, MARK I, ENIAC (Electronic Numerical Integrator and Calculator) auch nicht COLOSSUS (UK) realisierten die vollständige von Neumann-Architektur, dies geschah erstmalig mit der EDSAC (Electronic Delay Storage Automatic Calculator) (UK) im Mai 1949 (nach [Zuse 1999, S. 18]).

<sup>31</sup> In Zukunft wird nach dem heutigen Stand die Wartung von Software einen immer größeren Stellenwert erhalten. Erste Vorboten davon werden angesichts des Y2K-Problems (Y2K – Year 2 Kilo (Jahr 2000)) deutlich.

<sup>32</sup> durch entsprechende Compiler

## 2.2 Methoden der Informatik

### Pragmatischer Ansatz

Eine „pragmatische Charakterisierung der Informatik“ kann zusammenfassend beschrieben werden als „Herstellung und Einsatz von Informatiksystemen<sup>33</sup> unter Berücksichtigung des Kontextes und ihrer Beziehung zur menschlichen geistigen Tätigkeit“ (vgl. [Floyd 1997, S. 238f]). Unter Benutzung der Begriffe operationale und autooperationale Form<sup>34</sup> wird die Frage nach dem informatischen Handeln in [Floyd 2001, S. 49] wie folgt beantwortet: „Informatik betreiben bedeutet, operationale Form zu modellieren und als autooperationale Form verfügbar zu machen“. Bezogen auf diese Zielbestimmung kommt der Methode zur Formalisierung als Voraussetzung zur Herstellung und Automatisierung zur Umsetzung in Informatiksysteme eine Schlüsselrolle zu.<sup>35</sup> Aktivitäten zur Umsetzung der o. a. Zielvorstellung werden als informatische Modellierung bezeichnet. Informatische Modelle zeichnen aus, dass sie eine Umsetzung erfahren, die das Modell wirksam werden läßt. Damit besteht eine enge Wechselwirkung zwischen der informatischen Modellierung und dem modellierten Realitätsausschnitt. Die Modellierung wirkt durch das erstellte Informatiksystem in den modellierten Bereich zurück und verändert diesen. Christiane FLOYD und Ralf KLISCHEWSKI charakterisieren in [Floyd und Klischewski 1998] die informatische Modellierung durch die Metaphern

- „Fenster zur Wirklichkeit“ zur Wahrnehmung der (ggf. virtuellen) Realität (in Anlehnung an [Wolff u. a. 1999]) und
- „Handgriff zur Wirklichkeit“ zur Entwicklung und Verwendung von Informatikmodellen.

Die Vorgehensweise zur Modellierung kann durch drei miteinander verschränkte Schritte dargestellt werden:

- Informatisierung (Anwendungsmodell des Gegenstandsbereichs),
- Diskretisieren (Spezifikation durch ein formales Modell) und
- Systemisieren (Definieren durch eine Menge von berechenbaren Funktionen).

nach [Floyd und Klischewski 1998, S. 22]

Um die mit der Methode der informatischen Modellierung verbundenen Probleme zu verdeutlichen, ist darauf hinzuweisen, dass ausgehend vom Problembereich eine Dekontextualisierung vorgenommen wird, die im Zuge des Einsatzes als Teil eines konkreten Informatiksystems eine Rekontextualisierung erfährt.<sup>36</sup> Zur Charakterisierung dieses Spannungsverhältnisses werden die Begriffe autooperationale Form [Floyd 1997], Hybridobjekte [Siefkes 2001] und algorithmische Zeichen [Nake 2001] vorgeschlagen. Diesen Begriffsbildungen ist gemeinsam, dass der Verantwortung der Informatikerinnen in dem Prozess der oben skizzierten Modellierung Rechnung getragen werden soll, wie [Fischer 2001] verdeutlicht. Nur die Berücksichtigung der sozialen Bedingtheit in allen Phasen der Modellierung führt dazu, dass Informatiksysteme als Werkzeuge soziale Prozesse unterstützen. Diese Berücksichtigung ermöglicht eine partizipative Softwareentwicklung. Dass ein Prozess, der die evolutionäre Softwareentwicklung berücksichtigt, methodisch nicht ohne Schwierigkeiten umgesetzt werden kann, wird in [Rauterberg 1992] herausgearbeitet und an einem Vorgehensmodell verdeutlicht.

Zunehmend wendet sich auch die Philosophie der hier dargestellten und in der Informatik innerfachlich ausführlich diskutierten Modellierung zu. Dabei werden erste Hypothesen über „technisch erzeugte Welten“ formuliert.

<sup>33</sup> Im Unterschied zu [Floyd 1997] wird hier nicht der Begriff Computerartefakt, sondern, wie in einer jüngeren Publikation [Floyd und Klischewski 1998] der Begriff Informatiksystem verwendet.

<sup>34</sup> von Fanny-Michaela REISIN in [Reisin 1992] eingeführt

<sup>35</sup> Zur Vertiefung des Begriffsapparates sei auf [Floyd 2001, S. 50-71] verwiesen.

<sup>36</sup> vgl. [Bittner 2001, S. 25 mit Verweis auf [Winograd und Flores 1986]]

Die Möglichkeiten einer technischen Welterzeugung für ein Subjekt schafft die Basis einer sinnlichen Erfassung derjenigen Aspekte der

- (i) realen Welt, die zugunsten anderer Aspekte unrealisiert bleiben, i. e. sie erlaubt eine Erkenntnis von Möglichkeitsaspekten der realen Welt. Sie schafft gleichzeitig durch Modifikationen gemäß und innerhalb der unterschiedlichen Idealtypen eine Basis für die
- (ii) sinnliche Erfahrung des eigenen Selbst, wie es auch hätte sein können, i. e. sie erlaubt eine Erfahrung mit Bezug auf die Möglichkeitsaspekte des eigenen Selbst.

Helmut LINNEWEBER-LAMMERSKITTEN auf dem XVII. Deutschen Kongress für Philosophie<sup>37</sup>  
– zitiert nach Stefan MÜNKER in [Münker 1996]

Von dem Blick auf die Tätigkeiten von Informatikerinnen unterscheiden sich die auf das Spannungsfeld von Theoriebildung und Praxisbezug und Entwicklung der Informatik als Methodologie gerichteten Überlegungen, die im Folgenden dargestellt werden.

### Verschränkung von Theorie und Praxis

Die Rolle der Entwicklung theoretischer Ergebnisse im Kontext der Informatik als Wissenschaft wird zunehmend bezogen auf eine deutlichere Praxisorientierung diskutiert. Die anfängliche Euphorie bezüglich der Nutzung formaler Methoden zur Erstellung von Software wird inzwischen kritisch gesehen. Aus der Entwicklung soll durch eine stärkere Berücksichtigung

- der Beziehungen zwischen Theorieentwicklung und Praxiswirksamkeit,
- der Relevanz und Anwendbarkeit theoretischer Ergebnisse und
- der Wichtigkeit von Experimenten

eine Neuorientierung der theoretischen Informatik erreicht werden (vgl. [Sannella 1997]). Die Grenzen der Church-Turing-These<sup>38</sup> werden in [Wegner 1997] plakativ dargestellt. Peter WEGNER fordert ein Überdenken des traditionellen Modells. Verschiedene Möglichkeiten zur Erweiterung der Turing-Maschine werden von van LEEUWEN und WIEDERMANN vorgestellt [van Leeuwen und Wiedermann 2000]. Sie beziehen sich in ihrer Turing-Maschine „mit Beratung“<sup>39</sup> auf Ergebnisse von SCHÖNING [Schöning 1995], und fordern, dass „... the classical Turing machine paradigm should be revised (extended) in order to capture the forms of computation that one observes in the systems and networks in modern information technology“ [van Leeuwen und Wiedermann 2000, S. 622].

Die aus der Mathematik übernommene Theoretische Informatik ist ebensowenig wie die Theorien anderer Nachbardisziplinen geeignet, Hybridisierung als genuine Aufgabe der Informatik sichtbar zu machen. Eine evolutionäre Theorie kann eher bei der gemeinsamen Entwicklung der gegensätzlichen Bereiche der Informatik helfen.

[Siefkes 2001, S. 802]

### Informatik als Methodologie

Erheblich über die vorgenannten Ansätze hinaus gehen die Forderungen, die Methoden der Informatik als dritte Modalität grundsätzlicher methodischer Ansätze der Wissenschaften auszuweisen. Jozef GRUSKA und Roland VOLLMAR stellen fest:

<sup>37</sup> Leipzig, 23.–27. September 1996, Kongresstitel: „Cognitio humana – Dynamik des Wissens und der Werte“

<sup>38</sup> Jede intuitiv berechenbare Funktion ist maschinell berechenbar und umgekehrt (vgl. [Hopcroft und Ullmann 1988, S. 172f]). Die Church-Turing-These wird auch als Church'sche Hypothese bezeichnet.

<sup>39</sup> im Original „Turing machine with advice“ – [van Leeuwen und Wiedermann 2000, S. 621]

Two basic methodologies of science and technology (but also of society in general) have been theoretical methodology and experimental methodology. Both of them have been very well worked out and very successful. The new methodology, that is emerging from the achievements of theoretical investigations and technological developments in informatics is a methodology that seems to have enormous potential. This new methodology allows to enlarge the power of theoretical and experimental methodologies, to bridge them and, in addition, to be a powerful tool in all the areas where the two basic methodologies have not really been fully successful. [...] An application of informatics as a methodology cannot only help to solve otherwise unsolvable problems but may also contribute important intellectual abstractions and discoveries to other fields and to create new conceptual frameworks needed there.

[Gruska und Vollmar 1997, S. 59f]

In einem Vortrag formuliert Roland VOLLMAR die zentralen Gedanken: „In den Natur- und Ingenieurwissenschaften bildet [das informatische Vorgehen ...] neben theoretischem und experimentellem Vorgehen die dritte Säule der wissenschaftlichen Arbeitsweise“ [Vollmar 2000, S. 8]. Ergebnisse aus den Bereichen Algorithmisierung, Formalisierung, Komplexitätsuntersuchungen, Untersuchung komplexer Systeme liefern für diese neue Methodologie der Informatik die Voraussetzungen (nach [Vollmar 2000, S. 6f]). „Wesentliche Fortschritte werden dabei erzielt durch Simulation und Visualisierung“ [Vollmar 2000, S. 7]. „Die Informatik erweitert die durch Theorie und Experiment gebotenen Möglichkeiten beträchtlich, insbesondere in den bisher nicht zugänglichen Bereichen komplexer Systeme [...]. Komplexe Vorgänge werden verstehbarer, es können Voraussagen über ihr (künftiges) Verhalten gemacht werden, die auch dazu benutzt werden können, entsprechende [...] Prozesse zu optimieren“ [Vollmar 2000, S. 8].

Ähnliche Überlegungen werden von Peter BERGER unter Bezugnahme auf Wolfgang FRÜHWALD (DFG-Präsident von 1992–1997) unter der Überschrift „Revolution des Wissens: Die Visualisierung des Abstrakten“ zusammengefasst (vgl. [Berger 2001, S. 181]).

Im Einzelfall kann gezeigt werden, dass sich für die Unterstützung von Klärungsprozessen, die nicht primär in ein Informatiksystem gegossen werden sollen, eine Analyse mit Informatikmethoden als nützlich und hilfreich erweist. Beispielsweise konnten durch informatikbasierte Strukturierung mittels Petrinetzen Klärungsprozesse für den Gegenstandsbereich kommuniziert werden, die in dieser Klarheit von der dem Gegenstandsbereich zugrunde liegenden Wissenschaft<sup>40</sup> vordem nicht geleistet worden sind<sup>41</sup> (vgl. [Hinck u. a. 2001]).

## Zusammenfassung

Pragmatisch angelegte Ansätze zu den Überlegungen, welche Methoden in der Informatik zum Einsatz gebracht werden, führen zu der Schlüsselbestimmung „Informatische Modellierung im Kontext“ und dabei insbesondere zu der Besonderheit von Informatiksystemen, die Modellierung eines Realitätsausschnitts in eben dieser Realität wirksam werden zu lassen.

Von einer methodologischen Warte aus ist festzustellen, dass der theoriegeleiteten Softwareentwicklung keine durchgängige Anwendung bei der Erstellung von Informatiksystemen zufällt<sup>42</sup> und daher diskutiert wird, den Aufgabenbereich der theoretischen Informatik weiter zu fassen.

Erheblich darüber hinaus reichen die Überlegungen, Informatik als dritte Säule der wissenschaftlichen Arbeitsweise auszuweisen. Den etablierten methodologischen Ausrichtungen Theoriebildung und Empirie wird Informatik zur Verbindung von Theorie und Praxis zur Seite gestellt. Damit soll der Rolle und der

<sup>40</sup> an Beispielen aus der Soziologie und den Sozialwissenschaften (siehe Fußnote 41)

<sup>41</sup> • Fallbeispiel aus der Soziologie [Köhler u. a. 2001], [Valk 2002];

• CASSENS illustriert mit dem "Fallbeispiel Sozialwissenschaften" die Zielrichtung der Informatik, die "eng verknüpft ist mit einer neuen Art wissenschaftlicher Arbeit", bei der von vornherein die Frage der problemlösungsorientierten Anwendung im Fokus steht (vgl. [Cassens 2001, S. 36f]).

<sup>42</sup> Forschungsbeiträge von Informatikerinnen, die weder ein theoretisches Modell noch eine Implementierung anbieten, sind fragwürdig, wie Gregor SNETLING ausführt (vgl. [Snelting 1998, S. 276]).

Anwendbarkeit der durch die Informatik zur Verfügung gestellten Hilfsmittel in modernen Gesellschaften Rechnung getragen werden.

Dieser Ansatz stellt eine wissenschaftstheoretisch interessante Vision dar, die allerdings in der Informatik bisher nur von einzelnen Autorinnen rezipiert wurde. Die Konsequenzen dieser Überlegungen würden zu einer Neuorientierung der Wissenschaftstheorie führen.<sup>43</sup>

Gerade diese extreme Position findet innerhalb der Informatik nicht ungeteilte Zustimmung. So führt Peter BITTNER aus: „Von GRUSKA/VOLLMAR war ich nicht wirklich überzeugt, aber das liegt vielleicht daran, dass ich nicht glaube, dass man Informatik geschickter über eine wissenschaftstheoretische Einordnung charakterisieren sollte, ich entwickle das handlungstheoretisch bzw. bzgl. bestimmter Handlungsqualitäten“<sup>44</sup>

## 2.3 Bewertung von Definitionen von Informatik

Im Laufe der Zeit wurden von vielen Wissenschaftlerinnen Definitionen von Informatik vorgelegt. Einige dieser Definitionen, die eine besondere Sicht auf die Fachwissenschaft deutlich werden lassen, werden im Folgenden dokumentiert. Zur Genese der Informatik im deutschen Sprachraum liegen einige Darstellungen vor.<sup>45</sup> Im Zusammenhang mit der Begriffsbildung der Informatik als Wissenschaft, ist die Zeit ab 1970 wegen der Bezüge zur informatischen Bildung von besonderem Interesse.

ADAM definiert: „Informatik = Semiotik  $\cap$  (Kybernetik  $\cup$  Bionik)“<sup>46</sup> und führt aus: „[Informatik ist die] Lehre von den »Integralen Informationssystemen« die sowohl Mitwelt, als auch die Umwelt und die Zeichenwelt im mannigfaltigen Zusammenspiel zu beschreiben, erklären und zu gestalten versuchen“ [Adam 1971, S. 9]. „Es ist sehr bedenklich, die Strukturen eines puristisch-mathematischen Denkens in die Sprache der abstrakten Automaten zu objektivieren und diese Schöpfungen über ein cleveres Marketing der manipulierbaren Gesellschaft aufzunötigen“ [Adam 1971, S. 11]. Der Verweis auf die Tradition, in der die Informatik (zu Beginn) steht und aus der diese Definition abgeleitet wird, macht deutlich, wie wichtig die Rezeption der „vergessenen Definitionen“ gerade heute ist. Beide Textstellen verdeutlichen, dass vor über 30 Jahren Perspektiven der gerade erst im Aufbau begriffenen Informatik „korrekt“ prognostiziert wurden.

ZEMANEK bezeichnet den Informatiker als „Ingenieur ganz neuer Art“. Er charakterisiert den Objektbereich der Informatik: „Beim Informatiker sind die Gebilde, über die er spricht, bereits abstrakt und auf dem Papier, nämlich Programme und Beschreibungen. [...] Der Informatiker konstruiert, aber was er konstruiert sind abstrakte Objekte [...] Das Ersatzteillager des Informatikers ist so abstrakt wie seine abstrakten Objekte – aber es muß existieren“ [Zemanek 1971, S. 160f]. Damit identifiziert ZEMANEK die Tätigkeit von Informatikerinnen als Ingenieurität über die Konstruktion von Gebilden. Der Unterschied zur klassischen Ingenieurin besteht in der Art der Gebilde. Die klassische Ingenieurin konstruiert konkrete Gebilde, während die Informatikerin als „Ingenieur neuer Art“ geistig-abstrakte Gebilde, nämlich Zeichensysteme ohne räumlichen oder gegenständlichen Charakter konstruiert. Damit wird von ZEMANEK das besondere Berufsverständnis von Informatikerinnen auf die Ingenieurwissenschaften bezogen. Die verwendete Metapher Gebilde macht darüber hinaus deutlich, dass sich im Gegenstandsbereich keine konkreten, sondern abstrakte Objekte befinden.

WEIZSÄCKER benennt in [von Weizsäcker 1971] zwei Strukturwissenschaften: Mathematik und Informatik. Damit verbindet er den Anspruch der Informatik, für andere Disziplinen die automatisierte Informationsverarbeitung bereitzustellen. Strukturierungskonzepte, wie z. B. Hierarchisieren und Modularisieren,

<sup>43</sup> Die Informatik bedarf der Auseinandersetzung mit wissenschaftstheoretischen Fragestellungen. Dies wird von Wilhelm BÜTTEMEYER in [Büttemeyer 1995] eindrucksvoll verdeutlicht. Er weist nicht auf die im letzten Abschnitt dargestellten, weit über die Informatik hinausgehenden Überlegungen hin. Dies ist der Tatsache geschuldet, dass die Diskussion dieser visionären Positionen (öffentlich) erst nach Erscheinen geführt wird.

<sup>44</sup> persönliche E-Mail im Kontext der Diskussion der Thesen von GRUSKA und VOLLMAR.

<sup>45</sup> vgl. [Mainzer 1979], [Brödner u. a. 1981], [Krabbel und Kuhlmann 1994], [Städtler u. a. 1997], [Vollmer 1999]

<sup>46</sup> **Semiotik** – Lehre von den Zeichen (und ihrer Bedeutung)

**Bionik** – Biologie und Technik – beschäftigt sich mit der technischen Umsetzung von Konstruktions-, Verfahrens- und Entwicklungsprinzipien biologischer Systeme

vgl. [Hermann 1980]

sind unabhängig von der technischen Realisierung wertvolle Methoden zur Kontrolle und Handhabbarkeit von Komplexität.

Im Studien- und Forschungsführer Informatik (Ausgabe 1973) wird ebenfalls der Schlüsselbegriff „Struktur“ zur Charakterisierung des Gegenstandsbereichs der Informatik verwendet. Die Informatik befasst sich demnach mit der Struktur von Informationsverarbeitungssystemen, den Strukturen von Informationsverarbeitungsprozessen und der Möglichkeit der Strukturierung, Formalisierung und Mathematisierung von Anwendungsgebieten (nach [Brauer u. a. 1973, S. 12]). Die Aufgabe der Informatik besteht nach BRAUER darin, „durch Abstraktion und Modellbildung von speziellen Gegebenheiten sowohl der technischen Realisierung existierender Datenverarbeitungsanlagen als auch von Besonderheiten spezieller Anwendungen abzusehen und dadurch zu den allgemeinen Gesetzen, die der Informationsverarbeitung zugrunde liegen, vorzustoßen [...]“ [Brauer u. a. 1973, S. 12]. Die Informatik setzt sich nach dieser Einschätzung mit Strukturen und allgemeinen Gesetzen auseinander, die mit geeigneten Methoden aus konkreten technischen Artefakten und Anwendungsgebieten erschlossen werden sollen.

BAUER konstruiert den Bereich der informatischen Objekte historisch, in dem er für die Informatik zwei Entwicklungen zusammenführt: die Verwendung formaler Zeichensysteme, vor allem zur Kodierung und die Konstruktion gegenständlicher Maschinen, Spielautomaten, mechanischer Rechenmaschinen und automatischer Webstühle. „Damit haben wir mit den Elementen Codierung durch Zeichen, Mechanisierung der Operationen mit Zeichen, programmierbare Ablaufsteuerung von Operationen die Grundlagen des Wissenschaftsinhaltes der Informatik, die in der Verbindung dieser Elemente in einem Programm, das einen Algorithmus darstellt, gipfelt und insofern als Wissenschaft von der Programmierung der Informations-, das heißt Zeichenverarbeitung aufgefaßt werden kann“ [Bauer 1974, S. 335]. Nach BAUER ist die Informatik „weder Mathematik, noch Elektrotechnik, sie ist eine Ingenieur-Geisteswissenschaft (oder eine Geistes-Ingenieurwissenschaft, wem das besser gefällt)“ [Bauer 1974, S. 336]. Damit wird eine Gegenposition zu ZEMANEK, aber auch eine Abgrenzung zu WEIZSÄCKER deutlich.

Übereinstimmend wird sowohl vom ZEMANEK, aber auch von BRAUER und BAUER für die Informatik in Anspruch genommen, eine praxisorientierte Disziplin zu sein und damit relevante Beiträge für die Praxis zu liefern, zugleich aber eine abstrakte Wissenschaft mit einem abstrakten Gegenstandsbereich zu sein: „[...] der Ausdruck Angewandte Informatik ist insofern pleonastisch“ [Bauer 1974, S. 336]. Abstraktion und Praxisorientierung stellen damit keine Gegensätze dar (vgl. [Städtler u. a. 1997, S. 16f]).

„Im Vordergrund stehen prinzipielle Verfahren und nicht spezielle Realisierungen [...]. Die Inhalte der Informatik sind daher vorwiegend logischer Natur und maschinenunabhängig“ [Claus 1975, S. 11]. CLAUS bestärkt damit ebenfalls die Sichtweise WEIZSÄCKERS auf die Informatik und betont den Wissenschaftsschwerpunkt von „Algorithmen und Datenstrukturen“, die mit formalen Sprachen beschrieben und mit logischen Kalkülen ausgeführt werden.

GENRICH und PETRI kommen ausgehend vom Informationsbegriff der Kybernetik zur Sicht auf die Informatik als „Wissenschaft vom streng geregelten Informationsfluß“ [Genrich 1975a], [Genrich 1975b, Petri 1983]. Damit wird die Beschreibung von Prozessen einschliesslich der gesellschaftlichen Nützlichkeitsbetrachtung in den Mittelpunkt gerückt. Zur Bewertung von Informatiksystemen tragen dann nicht ausschließlich Strukturierungsqualität und algorithmische Effizienz, sondern auch die Einbettung in die sozialen Anwendungsprozesse bei.

NYGAARD: „When it is argued that informatics is a formal discipline only, then »according to such a definition, the impact of an information system upon the social structure of which it is a part, is outside (the field of study of) informatics. Also case studies of how data processing actually is carried out in specific organizations fall outside informatics in this narrow sense« [Håndlykken und Nygaard 1981].“<sup>47</sup> Damit verdeutlicht NYGAARD, dass eine einengende Sicht auf die Informatik aufgehoben werden muss.<sup>48</sup>

Mit ihrer wachsenden gesellschaftlichen Bedeutung wurde die Informatik zum Gegenstand wissenschaftstheoretischer Vergleiche, die weit über die innerdisziplinären Aspekte hinausreichen. So charakterisierte

<sup>47</sup> zitiert nach [Nygaard 1986, S. 189]

<sup>48</sup> Allerdings wird die von NYGAARD angegebene Definition: „Informatics is the science that has as its domain information processes and related phenomena in artifacts, society, and nature“ [a. a. O.] von COY [Coy 2001, S. 9] als erheblich „zu weit“ charakterisiert: „wieso [sollten] Informationsprozesse in der Natur Thema der Informatik sein“ [a. a. O.].

CAPURRO die „Informatik als hermeneutische Disziplin“ [Capurro 1990, S. 315] mit der Aufgabe der „*technischen* Gestaltung [...] menschlicher] Interaktionen in der Welt“ [Capurro 1990, S. 317].

Diese Einschätzung geht vielen Informatikerinnen zu weit, da sie sich von den damit verbundenen Konsequenzen überfordert fühlen. Gerade für die Didaktik der Informatik eröffnet diese philosophische Dimension Chancen für Synergieeffekte zwischen Fächern.

Mit FLOYD gelang die produktive Verbindung der ursprünglich gegensätzlichen Positionen:

„That means, it views itself as a formal and an engineering science, relying strongly on the traditional scientific paradigm [...]“ [Floyd 1992, S. 19].

COY: „Aufgabe der Informatik ist also die Analyse von Arbeitsprozessen und ihre konstruktive, maschinelle Unterstützung. Nicht die Maschine, sondern die Organisation und Gestaltung von Arbeitsplätzen steht als wesentliche Aufgabe im Mittelpunkt der Informatik. Die Gestaltung der Maschinen, der Hardware und der Software ist dieser primären Aufgabe untergeordnet. Informatik ist also nicht ‚Computerwissenschaft‘. An dieser Stelle zeigt sich deutlich, daß sich Informatik von Nachrichten- oder Informationstechnik in ihrer Ausrichtung wesentlich unterscheidet; der Kern der Unterscheidung liegt in der viel engeren Kopplung der Informatik an reale Arbeitsprozesse“ [Coy 1992, S. 18f]<sup>49</sup>.

Der Autor stimmt mit dieser Position nur insofern überein, als Informatik nicht nur als Strukturwissenschaft und technische Wissenschaft betrachtet werden kann, sondern in der Informatikausbildung auch der soziotechnische Kontext zu thematisieren ist. Im Anhang werden dazu Unterrichtsbeispiele dokumentiert. Entschieden abgelehnt wird von dem Autor die Argumentation von COY, die die Informatik den Sozialwissenschaften zuordnet. Im Rahmen der Allgemeinbildung wurde mit dieser Begründung bisher das Pflichtfach Informatik für alle Schülerinnen abgelehnt und so die fachlich fundierte informatische Qualifizierung von Lehrerinnen als unnötig betrachtet.

LUFT charakterisiert Informatik als eine Disziplin, die „[...] im Hinblick auf Entwurf und Gestaltung der Architektur weitaus näher steht als der Elektro- und Nachrichtentechnik“ [Luft 1992, S. 50].

Diese Position ist durch die Erfolge von Entwurfsmustern in der Softwaretechnik ausgesprochen aktuell. Zu prüfen bleibt die kognitive Wirksamkeit von Grundmustern, die im Rahmen der Informatikausbildung Schülerinnen beim Wissenstransfer unterstützen sollen.

BRAUER: „Ausgangspunkt der Denkansätze in der Informatik [ist] fast immer das Bemühen, Aspekte intelligenten Verhaltens von Lebewesen formal zu modellieren, um entsprechende formale Modelle als Unterstützungssysteme für den Menschen praktisch zu realisieren – oder [...] um das Verhalten der Lebewesen mit Hilfe der Informatik besser zu verstehen. [...] Informatik ist die (Ingenieur-)Wissenschaft von der theoretischen Analyse und Konzeption, der organisatorischen und technischen Gestaltung sowie der konkreten Realisierung von (komplexen) Systemen aus miteinander und mit ihrer Umwelt kommunizierenden (in gewissem Maße intelligenten und autonomen) Agenten oder Akteuren, die als Unterstützungssysteme für den Menschen in unsere Zivilisation eingebettet werden müssen“ [Brauer und Münch 1996, S. 13].

Zum Einen wird die zielgerichtete informatische Modellierung als zentrale Methode der Informatik benannt. Darüber hinaus findet die gesteigerte Bedeutung der Informatik in der Gesellschaft ihren Niederschlag in der Definition, die die Anwendungsbereiche explizit berücksichtigt.

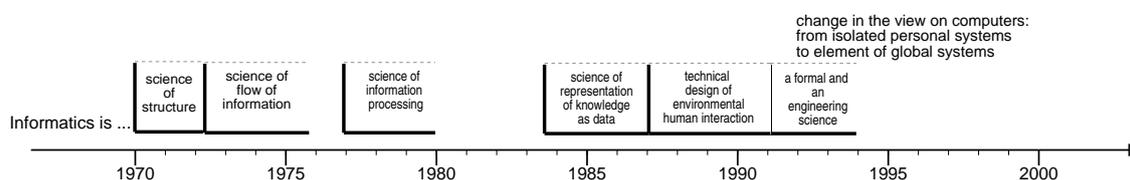
ACM veröffentlichte 1997 und 1999 Artikel zu den potenziellen Möglichkeiten der Informatik [Denning und Metcalfe 1997, Denning 1999b]. Aus dieser Vielfalt sei DENNING zitiert:

„[...] Computers have given us new ways of thinking about machines, communications, organizations, societies, countries, and economies. [...] A growing number of educators, for example, say that there is much more to learning than transferring information; they say the phenomenon of embodied knowledge, learned through practice and involvement with other people, is a process that cannot be understood simply as information transfer“ [Denning 1999a, S. 6f der Online-Fassung].

Hier wird in knapper Form zusammenfassend dargestellt, dass die Informatik zu neuen Denkweisen in unterschiedlichen Gegenstandsbereichen führt. Die Bindung an die informationsverarbeitenden Maschinen

<sup>49</sup> Die COY’schen Überlegungen wurden von RECHENBERG mit der Bemerkung: „Hier wackelt der (ideologische) Schwanz mit dem Hund!“ [Rechenberg 1991, S. 289] kommentiert. Damit wird deutlich, dass der Prozess der Selbstbestimmung des Faches keine von Polemik freie Diskussion ist.

(„Computer“) stellt m. E. eine Einengung bei der Angabe der Ursache dar. Dennoch ist das Zitat hilfreich, um Bildungskonzepte unter der Überschrift Medienkompetenz auf ihren Bezug zu den informatischen Basiskonzepten zu prüfen.



[Humbert und Schubert 2002, S. 6]

Abbildung 2.1: Zeitleiste zu Innensichten der Informatik<sup>50</sup>

Bei der Einordnung der vorgestellten Definitionen muss der historische Kontext Berücksichtigung finden. Bei den ersten vorgestellten Definitionen werden der eigenständige Charakter der neuen Wissenschaft und die Unterschiede zu anderen Wissenschaften in den Vordergrund gerückt. Daran schließt sich die Phase der innerfachlichen Diskussion an, für die 1989 durch DIJKSTRAS Forderung nach einer Brandmauer<sup>51</sup> zwischen dem formalen Kern und dem „Gefälligkeitsproblem“<sup>52</sup> das Basisproblem der Informatik benannt wird, ohne es lösen zu können (vgl. [Coy 1997, S. 24ff]). Auf beiden Seiten dieser Brandmauer sind informatische Qualifikationen erforderlich. Die jeweilig durch die Fachwissenschaft einzubringenden Elemente unterscheiden sich bezüglich der Möglichkeiten „richtige“ Lösungen für Problemstellungen zu finden: diesseits der Brandmauer werden Probleme bearbeitet, die exakt lösbar sind, während auf der anderen Seite der Brandmauer der Kontext, in dem Menschen agieren, berücksichtigt werden muss.

Nach der Etablierung der Informatik und den ersten größeren innerfachlichen Diskussionsprozessen um das Selbstverständnis<sup>53</sup> beginnt eine Phase, in der übergreifende Fragestellungen mit anderen Wissenschaften stärker in den Fokus des Prozesses der Selbstvergewisserung einbezogen werden. Inzwischen wird das Besondere – „das Neuartige“, wie BRAUER formuliert – auch darin gesehen, dass die Informatik eine „Kooperationspartnerin für jede Wissenschaft und jede Sparte praktischer Tätigkeiten“ [Brauer und Münch 1996, S. 12] ist. Jedoch sind zentrale Probleme nach wie vor nicht befriedigend geklärt, wie DIJKSTRA deutlich macht: „[...] most of our systems are much more complicated than can be considered healthy, and are too messy and chaotic to be used in comfort and confidence. The average customer of the computing industry has been served so poorly that he expects his system to crash all the time, and we witness a massive worldwide distribution of bug-ridden software for which we should be deeply ashamed. For us scientists it is very tempting to blame the lack of education of the average engineer [...]. You see, while we all know that unmastered complexity is at the root of the misery, we do not know what degree of simplicity can be obtained, nor to what extent the intrinsic complexity of the whole design has to show up in the interfaces“ [Dijkstra 2001]. Diesen Problemen hat sich die Wissenschaft Informatik zu stellen.

Es ist festzustellen, dass sowohl der Paradigmenwechsel (vgl. Kapitel 1, S. 5) innerhalb der Informatik, wie auch der Einfluss der Informatik auf andere Wissenschaften in konstruktiver Weise in den Definitionen berücksichtigt werden.

<sup>50</sup> Hier werden die Innensichten angegeben, die zur Fundierung der Schulinformatik bedeutsam sind.

<sup>51</sup> im amerikanischen Original: firewall – [Dijkstra 1989, S. 1414]

<sup>52</sup> dito – „pleasantness problem“

<sup>53</sup> Dabei geht es um Informatikerinnen, die sich – ausgehend vom Bemühen um Anerkennung als eigenständige Wissenschaft

- offiziell – Definition des Fachs
- inoffiziell – Sicherung des Zugriffs auf Ressourcen

auf den Weg zu einer (dem Anspruch nach) universellen Anwendbarkeit ihrer Methoden (Ingenieursaspekt) begeben haben.

## 2.4 Konzepte in der Informatik

Für die konkrete wissenschaftlich begründete Arbeitsweise von Informatikerinnen stellt die Umsetzung der wissenschaftstheoretischen Ergebnisse auf der Ebene konkreter Konzepte eine notwendige Arbeitsgrundlage dar. Die Wissenschaft liefert Ansätze, die auf verschiedenen Ebenen handlungsleitende Ideen explizieren. Mit Blick auf die Schulinformatik werden sowohl historisch bedeutsame Analysen, aber auch Entwicklungslinien vorgestellt, die es ermöglichen sollen, Konzepte zu identifizieren, die im Kontext eines allgemein bildenden Informatikunterrichts Berücksichtigung finden. Ausgehend von dem bereits im Kontext des „Pragmatischen Ansatzes“ im Abschnitt 2.2 (S. 13) verdeutlichten grundlegenden Begriff der operationalen Form<sup>54</sup> werden verschiedene Typen von operationalen Strukturen unterschieden. Die folgende Liste nach Christiane FLOYD stellt einen ersten Versuch für eine Typologie dar:

1. Grundlegende operationale Strukturen  
z. B. Suchen, endliche Automaten – Kontext „Algorithmen und Datenstrukturen“,
2. Verallgemeinerung der operationalen (Re-)Konstruktion durch Orientierung an einer Klasse verwandter Gegenstandsbereiche  
z. B. innerhalb einer Branche,
3. Identifikation verwandter operationaler Strukturen in verschiedenen Gegenstandsbereichen  
z. B. Konfiguration, Entscheidungsfindung – gemeinsame Grundstruktur, unterschiedliche Ausprägung.

nach [Floyd 2001, S. 68ff]

Zur Entwicklung von dekontextualisierten, operationalen Strukturen hat die Informatik besondere Abstraktionsmechanismen, namentlich die Prozessabstraktion und die Datenabstraktion entwickelt. Diese sind i. d. R. als parametrisierte Algorithmen beschrieben, die auf allgemeinen Datenstrukturen operieren<sup>55</sup>. Für den Bereich der Objektorientierung konnten mit Entwurfsmustern (vgl. [Gamma u. a. 1996], [Quibeldey-Cirke 1995]) erste Ansätze für die Darstellung und Katalogisierung typischer wiederkehrender Lösungsmuster vorgelegt werden.<sup>56</sup>

Bei der folgenden Diskussion soll nicht vergessen werden, dass es weder einen „Königsweg“ der Modellierung noch die vollständige Adaption der „Realität“ inklusiv der in diesen Kontext tätigen Menschen zur Konstruktion von Informatiksystemen gibt. Alle Versuche müssen sich auf ihre Relevanz bezüglich des modellierten Realitätsausschnitts hin prüfen lassen und werden in einem konkreten durch beschränkte Ressourcen und eine Mit-/Umwelt vorgegebenen Rahmenbedingungen entwickelt. Aus diesem Grund werden den folgenden Ausführungen zwei Zitate vorangestellt, die deutlich machen, dass es letztlich um Menschen geht, wenn Informatiksysteme modelliert werden. „Alle weiteren Betrachtungen müssen neben den Konzepten der Programmierung auch die Umwelt mit einbeziehen: Es sind Menschen, die in einem kooperativen, kreativen und/oder ingenieurmäßigen Prozess Realität modellieren und diese Modelle in Softwaresysteme umsetzen. Eine Schlüsselfunktion auf dieser Ebene sind Methoden zur Softwareentwicklung“ [Jähnchen und Herrmann 2002, S. 268]. „Beim evolutionären Vorgehen werden die unterschiedlichen Bereiche füreinander geöffnet. Wenn die Experten lernen, die Erfahrungen der anderen für sich nutzbar zu machen, können sich die unterschiedlichen Menschen und ihre Welten gemeinsam entwickeln, die Hybridisierung gelingt. Die Nutzer sind an der Entwicklung von Software und die Informatiker an der Entwicklung der Anwendungsbereiche beteiligt, ohne Experten in der jeweils anderen Welt zu sein“ [Siefkes 2001, S. 802].

Die Konzentration auf den Bereich der Softwaretechnik und damit auf die (konkrete) Modellierung ist Folge der in diesem Bereich deutlich hervortretenden Probleme. Diese Darstellung stellt keine Einschränkung

<sup>54</sup> entspricht der Symbolischen Maschine im Sinne von [Krämer 1988], siehe 4.2.1, S. 66 – nach [Floyd 1997] und [Floyd 2001]

<sup>55</sup> Beispiele: [Sedgewick 1992], [Ottmann und Widmayer 1990]

<sup>56</sup> Bereits früher entwickelte „Frameworks“ sind für einen breiten Einsatz nicht geeignet, da sie für eine spezielle Klasse von Problemen Lösungen zur Verfügung stellen (Beispiele: [Williams 1984, Lisa Toolkit], [Schmucker 1986, MacApp für die Macintosh Benutzungsoberfläche]).

Lösungsmuster der Gang of Four (GOF = Gamma u. a.) hingegen stellen ein allgemeineres Hilfsmittel für die Softwareentwicklung dar und sind zudem programmiersprachenunabhängig ausgelegt (auch wenn gewisse Muster ihren Bezug zu Defiziten konkreter Sprachen nicht verleugnen können – z. B. Singleton-Muster [Gamma u. a. 1996, S. 139ff]).

auf dieses Teilgebiet der Informatik dar, da Ergebnisse aus anderen Teilen der Informatik Eingang in die Modellierung finden, wie in den folgenden Überlegungen deutlich wird. Die Voraussetzung für die Überführung theoretischer Ergebnisse zur Anwendung ist im Bereich der Informatik oftmals nicht gegeben, da im informatiknahen Bereich überdurchschnittlich viele Menschen arbeiten, die nicht primär für die Arbeit in der Informatik qualifiziert sind (vgl. [Dostal 2000, Dostal 2001, Dostal 2002]).

## Paradigmen – Sprachklassen

Im Folgenden werden „Sichten auf die Welt“, die durch verschiedene Programmierparadigmen geprägt sind, dargestellt, um sie konzeptionell einzuordnen. Die „angemessene“ Sicht auf den Problembereich ist die Voraussetzung für eine aufgabenadäquate Analyse.

Paradigmen werden in diesem Zusammenhang<sup>57</sup> als übergreifende Prinzipien verstanden, die dazu geeignet sind, durch eine bestimmte „Brille“ auf einen Problembereich zu sehen und damit eine bestimmte Sicht in den Problemlösungsprozess einfließen zu lassen. Es werden verschiedene Arten der „Weltsicht“ als Paradigmen für die Softwareentwicklung ausgewiesen. Diesen „Auffassungen der Problemwelt“ können – bezogen auf die Möglichkeiten zur Umsetzung und zur Implementierung – Programmiersprachklassen zugeordnet werden.

Auffassung	Sprachklasse
Auswertung von Ausdrücken (einer formalen Sprache)	funktionale und applikative Sprachen <sup>58</sup>
Beantwortung von Anfragen (an ein Informationssystem)	relationale und logische Sprachen <sup>59</sup>
Manipulation von Objekten (der realen Welt)	prozedurale, imperative und objektorientierte Sprachen <sup>60</sup>

nach [Padawitz 2000, S. 5]

Die Fußnoten verweisen auf Erläuterungen und Umsetzungen für den schulischen Zusammenhang.

Tabelle 2.1: Auffassungen und ihre Ausprägung in Sprachklassen

Die Zuordnung einer konkreten Programmiersprache zu genau einer Sprachklasse fällt nicht immer leicht. Eine Reihe von imperativen, prozeduralen aber auch funktionalen Sprachen wurden um Elemente der Objektorientierung erweitert, so dass hier keine eindeutige Zuordnung (mehr) möglich ist (z. B. Object-Pascal [Tesler 1984, Schmucker 1986], Modula-3 [Harbison 1990], CLOS (Common LISP Object System) [Keene 1989]). Bei sogenannten 4GL<sup>61</sup>-Sprachen ist die Zuordnung zu einer Sprachklasse erschwert, da diese typischerweise mehreren Paradigmen zuzuordnen sind. Darüber hinaus wurden für Forschungszwecke Sprachen entwickelt, die mehrere Paradigmen unterstützen (vgl. [Spinellis u. a. 1994]).

<sup>57</sup> im Unterschied zu dem Begriff des Paradigmas, wie er in Kapitel 1, S. 1 („Definition“ des Begriffs nach KUHN) dargestellt wurde.

<sup>58</sup> Alle Objekte werden als Funktion  $f : D \rightarrow E$  aufgefasst, selbst Konstante sind konstante Funktionen. Funktionen können Argument einer weiteren Funktion sein und vice versa. Siehe z. B. [Staudte 1992].

<sup>59</sup> Mit Hilfe des Prädikatenkalküls wird versucht, über Aussageformen Mengen zu beschreiben. Damit kann der Zusammenhang zwischen der mengentheoretischen und der aussagenlogischen Beschreibung von Problemstellungen zum Zweck der Bestimmung von Lösungsmengen geleistet werden. Siehe z. B. [Baumann 1991, Schubert 1992].

<sup>60</sup> **imperativ:** Im Mittelpunkt steht die an [virtuellen] Maschinen orientierte Programmiersprache; Ziel ist das Erlernen einer speziellen Programmiersprache – die Problemstellungen werden dem einzuführenden „Befehl“ angepasst. Dieser Ansatz wurde im Laufe der Zeit zunehmend in Richtung auf Strukturen hin verbessert und optimiert und stellt bis heute an vielen Schulen im Sekundarstufen II-Bereich den Standard des Informatikunterrichts dar. Siehe z. B. [Balzert 1983].

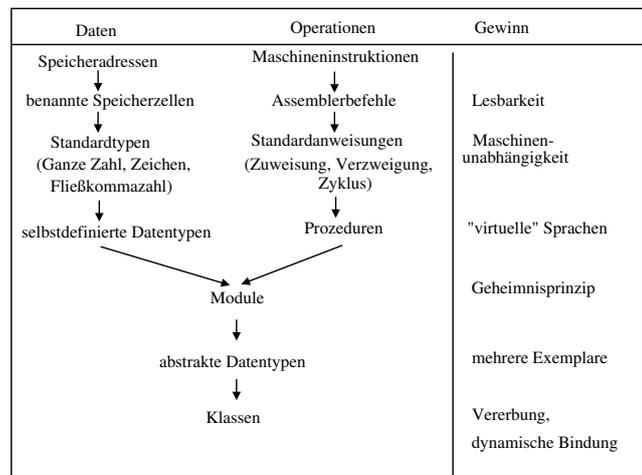
**objektorientiert:** Jedes System besteht aus Objekten. Jede Aktivität des Systems ist die Aktivität einzelner Objekte. Gleichartige Objekte werden zu Klassen zusammengefasst. Die Objekte stellen eine Einheit aus *Zustandsvariablen* und *Methoden* dar, d. h. im Objekt wird die imperative Trennung von Daten und Aktion auf (mit) den Daten [bei der Analyse] aufgehoben. Zentral ist die Analyse des Gegenstandes und der anschließende Entwurf, allerdings kommen zentrale imperative Konzepte bei der Erarbeitung von Methoden durchaus zum Einsatz. Siehe z. B. [Czischke u. a. 1999].

<sup>61</sup> 4<sup>th</sup> Generation Language (4GL) oder High Level Language (HLL)

Für die Informatikerin stellen die Auffassungen und die zugehörigen Sprachklassen Werkzeuge dar, die es ermöglichen, eine [teil-]problemangemessene Entscheidung zu treffen.<sup>62</sup> BALZERT fordert: „Jeder Software-Ingenieur muß wissen, welches Paradigma für welches Problem oder Teilproblem am besten geeignet ist. Es geht nicht um »Entweder – Oder« sondern um »Sowohl als auch«“ [Balzert 1996, S. 40f].

## Entwicklung von Abstraktionsmechanismen in Programmiersprachen

Abstraktionsstufen, die in der historischen Entwicklung von Programmiersprachen<sup>63</sup> eingeführt wurden, um die „semantische Lücke“ zwischen der Problemstellung (Problemwelt) und der Maschinenebene (dem Programm) zu schließen, werden von Hans-Peter MÖSSENBOCK beschrieben.



nach [Mössenböck 1992, Abb. 1.9, S. 10]

Abbildung 2.2: Entwicklung von Abstraktionsmechanismen in Programmiersprachen

Es sollte allerdings der Eindruck vermieden werden, dass mit der Objektorientierung ein Endpunkt der Entwicklung erreicht ist: „In der Tat wäre es unbefriedigend, wenn die Entwicklung von Blöcken über Prozeduren zu Modulen nun bei Klassen stehen bleiben sollte“ [Jähnichen und Herrmann 2002, S. 273].

## Von der schrittweisen Verfeinerung zur Objektorientierung

Der objektorientierte Entwurf unterscheidet sich von dem Verfahren der schrittweisen Verfeinerung<sup>64</sup>, das im Zusammenhang mit der strukturierten Programmierung [Dijkstra 1969, Dijkstra 1970, Wirth 1974, Wirth 1978] bedeutsam ist. Die Änderung der Sichtweise zur Bearbeitung von Problemstellungen von der aufgabenorientierten Sicht (Top-Down – vom Problem zum Programm) auf die objektorientierte Sicht (Bottom-Up – von den Objekten ausgehend) spielt insbesondere für die Gestaltung von Lehr-/Lernprozessen eine wichtige Rolle. Es ist notwendig, konsistente Strategien zu entwickeln, die methodisch so angelegt sind, dass Schülerinnen ein verständlicher Zugang zur Objektorientierung möglich wird. Dabei kann ein Verfahren berücksichtigt werden, das von Russell J. ABBOTT vorgeschlagen wurde.

<sup>62</sup> Ein weithin unbekanntes Beispiel für eine solche Entscheidung: „[...] in WindowsNT von Microsoft wird ein kleiner Prolog-Interpreter verwendet, um optimale Konfigurationen für Netzwerke zu erzeugen“ [Beckmann 1998, S. 83, Fußnote 2].

<sup>63</sup> Dabei werden ausschliesslich von Neumann Sprachen betrachtet.

<sup>64</sup> von Niklaus WIRTH in [Wirth 1971] vorgeschlagen

A technique [...] for developing programs from informal but precise English descriptions. The technique shows how to derive data types from common nouns, variables from direct references, operators from verbs and attributes, and control structures from their English equivalents. The primary contribution is the proposed relationships between common nouns and data types; the others follow directly.

[Abbott 1983, S. 882]

Das Verfahren ist auch für den objektorientierten Entwurf geeignet (vgl. [Abbott 1983, S. 885f], [Mössenböck 1992, S. 133]). Damit können Kandidaten für Klassen, Attribute und Methoden identifiziert werden. Darüber hinaus eignen sich die Verfahren der Anwendungsfallanalyse (Use-Cases) [Jacobson u. a. 1992] und die Benutzung von CRC-Karten [Cunningham und Beck 1986] um von Objekten aus dem zu modellierenden Problembereich zur Entwicklung von Klassenstrukturen zu gelangen.<sup>65</sup>

## Objektorientierung – Darstellung und Kritik

Die zentrale Aussage der Befürworterinnen der objektorientierten Modellierung: Die Welt ist einer objektorientierten Betrachtung zum Zwecke der informatischen Modellierung zugänglich.<sup>66</sup> Im Folgenden werden ausgewählte Argumente für die objektorientierte Modellierung dokumentiert.

Many people who have no idea how a computer works find the idea of object oriented systems quite natural. In contrast, many people who have experience with computers initially think there is something strange about object-oriented systems.

[Robson 1981]

Zu diesem Zeitpunkt (1981) kann die Objektorientierung noch nicht von dem Gros der Informatikerinnen als akzeptiert gelten. Die Hinweise von David ROBSON sind für Überlegungen im Zusammenhang mit der Gestaltung von Lehr-/Lernprozessen bedeutsam<sup>67</sup>, verdeutlichen aber gleichzeitig das Beharrungsvermögen derjenigen, die zu diesem Zeitpunkt nicht bereit sind, sich mit der Objektorientierung auseinander zu setzen.

The purpose of the Smalltalk project is to provide computer support for the creative spirit in everyone. Our work flows from a vision that includes a creative individual and the best computing hardware available. We have chosen to concentrate on two principle areas of research: a language of description (programming language) that serves as an interface between the models in the human mind and those in computing hardware, and a language of interaction (user interface) that matches the human communication system to that of the computer.

[... A] computer system should provide models that are compatible with those in the mind.

[Ingalls u. a. 1988]

Many research examples developed at PARC demonstrated that object-oriented design could produce an appealing, intuitive, and direct mapping between objects in the real world and objects in a software implementation. We saw this as a radical breakthrough in one of the most difficult and problem-prone steps in software development – identifying terms and relationships as understood by human participants of a particular situation with those understood by a computer.

[Deutsch und Goldberg 1991]

<sup>65</sup> Eine unterrichtliche Umsetzung findet sich in [Jochum 1998].

<sup>66</sup> Die Wurzeln der Objektorientierung sind in der Entwicklung der Programmiersprache Simula (ab 1962, angestossen durch Kristen NYGAARD und Ole-Johan DAHL, vgl. [Dahl 2002]) zu finden. Simula gehört zur ALGOL-Sprachfamilie. Die Ablösung der Objektorientierung von imperativen Wurzeln wurde in Smalltalk [Kay 1993], einer rein objektorientierten Sprache erreicht.

<sup>67</sup> David ROBSON, Adele GOLDBERG [Goldberg und Robson 1983] und Alan KAY haben die objektorientierte Sprache Smalltalk entwickelt. Zur Geschichte sei auf [Kay 1993, Kay 2002] verwiesen.

Die Welt ist objektorientiert, d. h. sie besteht aus Objekten und Relationen zwischen ihnen. Beides wird in/mit Klassen beschrieben und es gibt Vererbung und Polymorphie.

[Werner 2001]<sup>68</sup>

Um die Anforderungen an objektorientierte Programmiersprachen zu identifizieren, wurden 1987 Ergebnisse einer Untersuchung veröffentlicht: Peter WEGNER weist darin als „Dimensionen objektbasierter Sprachen“ objects, classes, inheritance, data abstraction, strong typing, concurrency und persistence aus [Wegner 1987, S. 168] und untersucht auf dieser Basis die zu diesem Zeitpunkt existierenden objektbasierte Sprachen. Um die Objektorientierung auf der sprachtheoretischer Ebene zu klassifizieren hat er den minimalen, orthogonalen Satz von Konzepten identifiziert:

- objects – modular computing agents
- types – expression classification mechanism
- delegation – resource sharing mechanism
- abstraction – interface specification mechanism

[Wegner 1987, S. 172]<sup>69</sup>

Bereits Wegner kommt zu dem Ergebnis, dass die orthogonalen Grundkonzepte nicht unbedingt das sind, was Programmierer direkt in die Hand bekommen sollten. Während die vier genannten Konzepte und Mechanismen für die sprachtheoretische Diskussion wichtig sind, ist beim Programmieren möglicherweise Vererbung interessanter, obwohl oder weil sie eine spezifische Mischung von Konzepten<sup>70</sup> darstellt.

[Jähnichen und Herrmann 2002, S. 268]

Dem Anspruch der Protagonistinnen der Objektorientierung, dass „die Welt“ objektorientiert ist, setzen die im Folgenden angegebenen Autorinnen einige bedenkenswerte Argumente entgegen.

1. Die Fokussierung auf die objektorientierten Softwareentwicklung vernachlässigt die Berücksichtigung der anderen Paradigmen (vor allem in der Ausbildung).

Es gibt auch eine Welt außerhalb der Objektorientierung. In diesem Sinne haben funktionale und logische Programmierung nach wie vor ihre Berechtigung. Interessanterweise gibt es für beide Fälle Systeme, die diese Paradigmen mit der Objektorientierung verbinden.

[Jähnichen und Herrmann 2002, S. 268]

2. Die [unterstellte] Simplifizierung wird als Hindernis für den notwendigen Verständigungsprozesse im Sinne der evolutionären Softwareentwicklung dargestellt.

In der objektorientierten Softwareentwicklung ist [... der] Erfolg [des evolutionären Vorgehens] wieder gefährdet. Das Programmieren wird auf das schematische Anwenden einfacher Techniken reduziert, die eine genauere Analyse unnötig erscheinen lassen. Die Objekte des Anwendungsbereichs lassen sich auf Papier oder Bildschirm so einfach darstellen, daß sie Anwendern und Informatikern gleich vertraut erscheinen; sie fühlen sich nicht auf Zusammenarbeit angewiesen. Die formal erzeugten Objekte machen die

<sup>68</sup> Sicherlich zeigt besonders die letzte Bemerkung über „die Welt“ eine zu simplifizierende Sicht.

<sup>69</sup> Deutsche Übersetzung:

- „Objekte: »modulare Berechnungsagenten«,
- Typen: »Mechanismus zur Klassifizierung von Ausdrücken«,
- Delegation: »Mechanismus zur gemeinsamen Nutzung von Ressourcen«,
- Abstraktion: »Mechanismus der Schnittstellenspezifikation«

[Jähnichen und Herrmann 2002, S. 268].

<sup>70</sup> „Vererbung ist ein Spezialfall von Delegation; Polymorphie ergibt sich in der Kombination mit Typisierung“ [Jähnichen und Herrmann 2002, S. 268].

Hybridisierung von Mensch- und Maschinenwelt vollkommen; die gemeinsam denkenden und handelnden Menschen verschwinden darin ebenso wie die ausführenden Maschinen und die beide verknüpfenden Formalismen. Die Objekte scheinen sich selbst zu entwickeln und als Agenten im Netz auch selber auszuführen.

[Siefkes 2001, S. 802 mit einem Verweis auf [Malsch 1998]]

3. Einige wichtige Konzepte sind in der bisherigen objektorientierten Sicht nur unzureichend berücksichtigt worden: insbesondere asynchrone Nebenläufigkeit [Broy und Siedersleben 2002, S. 10]. Die Objektorientierung löst trotz ihrer Beliebtheit längst nicht alle Probleme.

„Objektorientierung liefert keinen geeigneten Komponentenbegriff [...] kennt keine Komposition von Klassen, das OO-Ausführungsmodell ist [...] sequenziell [...] sagt uns nicht, wie wir das Verhalten [...] von Schnittstellen definieren sollen“ [Broy und Siedersleben 2002, S. 5].

Diesen Aussagen wird in [Jähnichen und Herrmann 2002] widersprochen. Exemplarisch seien hier einige Argumente angegeben: „Fakt ist, dass die Informatik sich allmählich mühsam auf einen Komponentenbegriff geeinigt hat“ [Jähnichen und Herrmann 2002, S. 273]. Zur Komposition merken die Autoren an „Alle diese Ansätze zeigen uns, dass neue Konzepte für Modularisierung und Komposition erforderlich sind, um komplexe Systeme langfristig wartbar zu halten. Die verschiedenen Entwicklungen verbinden sich unter dem Dach der aspektorientierten Softwareentwicklung“ [Jähnichen und Herrmann 2002, S. 274]. [Wegner 1987] „[...] führt zusätzlich noch Nebenläufigkeit und Persistenz an. Beide Konzepte stehen jedoch aus heutiger Perspektive an der Schwelle zwischen programmiersprachlichen Lösungen und separaten Techniken, die entweder im Programm explizit aufgerufen oder durch generative Techniken (z. B. Präprozessoren) eingeflochten werden“ [Jähnichen und Herrmann 2002, S. 270].

4. Fehlende Möglichkeit der statischen (Typen-)Prüfung objektorientierter Programme durch den Compiler stellen ein Problem der vollständigen Objektorientierung dar.

Im Oberon-System [Reiser 1991] wurde in der Sprache Oberon [Reiser und Wirth 1992] das Konzept der Objektorientierung durch Typen-Erweiterung [Wirth 1988] eingeführt. Mit Hilfe von Prozedurvariablen werden „allgemeine Methoden“ (sogen. Handler) eingeführt. Der Vorteil: „Statische Typenprüfung erlaubt es dem Compiler, Inkonsistenzen, d. h. Programmierfehler vor der Ausführung aufzuzeigen, und dem Programmierer, Datendefinitionen und -strukturen zu ändern ohne große Gefahr, daß bestimmte Konsequenzen der Änderung übersehen werden. Statische Typenprüfung beschleunigt nicht nur die Änderung: sehr oft würde man gar nicht wagen, sie ohne die Typenprüfung vorzunehmen“ [Wirth 1994, S. 9]. Größter Nachteil dieses Konzepts sind die dadurch notwendigen Verzweigungskaskaden, durch die die Programmiererin garantieren muss, dass alle Fälle behandelt werden.<sup>71</sup>

Konsequenzen der Diskussion:

Eine optimistische Einschätzung über die Durchsetzung der Objektorientierung von Adele GOLDBERG aus dem Jahr 1991.

We hope that in 2001, objects will be boring. In comparison, radical ideas of past decades – that system software should be written in higher-level languages or in languages with strong type systems, and that computers can and should be seamlessly networked – are thoroughly accepted today. Whether to implement them is almost never an issue now, even though there is still plenty of discussion about how to implement them well. In the same vein, we expect that 10 years from now, the object-oriented approach to software design and implementation will be an accepted, standard technique used in every language, library, database system, and operating system and will be taught in undergraduate computer science courses at every university. This is an issue of moving the technology further out into the world, and no major new thinking will be needed to accomplish it. One significant technological advance will be that we

<sup>71</sup> Trotz dieser kritischen Grundposition hat WIRTH das Vorwort ([Wirth 1992]) zu [Mössenböck 1992] geschrieben, in dem eine Erweiterung der Sprache Oberon um klassenbezogene Prozeduren (= Methoden) vorgenommen wird.

will free ourselves even further from equating objects with the nouns in the problem domain. Some of the most remarkable advances in the usability of computer systems have come from recognizing that processes, as well as things, can and should be described, modeled, and manipulated. Therefore, we will see software objects being used to model time, places, actions, and events.

[Deutsch und Goldberg 1991]

Viele der beschriebenen Erwartungen sind eingetroffen. Allerdings zeigen die oben dokumentierten Positionen, dass die mit der Durchsetzung der Objektorientierung verbundenen Probleme nicht ignoriert werden dürfen, sondern in der Lehre zu berücksichtigen sind, wie das folgende Zitat deutlich macht:

Um Studierende für künftige Entwicklungen zu wappnen, muss die Informatikausbildung darauf hinweisen, dass Objektorientierung ein Ding mit vielen Gesichtern ist und dass bestimmte Konflikte schwierige Abwägungen erfordern. Gerade die mächtigsten Techniken (wie z. B. das Template Hook-Muster<sup>72</sup>) erlauben eine sehr effektive Wiederverwendung (z. B. im Falle von Frameworks) und stellen gleichzeitig eine akute Gefahr für die Korrektheit des Programms dar. Nur Informatiker, die beide Seiten kennen, können sich ein eigenes Urteil bilden, in konkreten Projekten die geeignetsten Techniken auswählen und vielleicht an einer Verbesserung unserer Techniken und Methoden forschen. Aber vielleicht spricht dann schon keiner mehr von der Objektorientierung, so wie heute kaum noch vom strukturierten Programmieren die Rede ist gewisse Dinge sind irgendwann einfach selbstverständlich.

[Jähnichen und Herrmann 2002, S. 275]

Für die allgemeine Bildung muss die Frage nach dem Bildungswert der informatischen Modellierung insgesamt gestellt werden. Marco THOMAS widmet seine Dissertation der „Informatischen Modellbildung“ und führt aus:

Für die Informatik ist jedoch ein weitaus umfassenderes Modellieren von Modellen zu erwarten, da die Informationsverarbeitung (mit Modellen) und ihre Automatisierung wesentlicher Inhalt dieser Wissenschaft ist. Die Schwierigkeit der Einordnung der Informatik in den Wissenschaftskanon macht die Untersuchung des Modellverständnisses innerhalb dieser Fachwissenschaft besonders interessant.

[Thomas 2002, S. 25]

In der Bestimmung der „Konsequenzen für einen Informatikunterricht“ charakterisiert THOMAS informatische Modelle als kulturerschließend und -produzierend:

Informatische Modelle sind nicht nur Teil unserer Kultur, sondern mit ihnen wird Kultur erschlossen und produziert. Das Erfahren des (informatischen) Modellierens, das Bewusstmachen der Bedeutung von Modellbildung für den Menschen und der dabei verwendeten Modelloperationen dürfte für eine gezielte Auswahl von Modelltypen und deren Modellierung in anderen Bereichen recht nützlich sein. Wie der fallende Stein exemplarisch für das Fallgesetz stehen kann, so können (einzelne) informatische Modelle das Exemplarische für allgemeine Modelltypen darstellen.

[Thomas 2002, S. 68]<sup>73</sup>

In der Zusammenfassung wird deutlich zum Ausdruck gebracht, wie zentral der Modellbegriff für die allgemeine Bildung ist.

Informatische Modelle stellen ein Bildungsgut zur Enkulturation<sup>74</sup> des Modellierens von Modellen dar und lassen sich an interessanten und anspruchsvollen Themen konstruktiv und

<sup>72</sup> Schablonenmethode (Template Method) [Gamma u. a. 1996, S. 327ff]

<sup>73</sup> Diese Thesen müssen aus Ressourcengründen zur Zeit noch ohne empirische Prüfung bleiben, wie THOMAS ausführt.

<sup>74</sup> Enkulturation wird in der Soziologie als Begriff für den Prozess der Verinnerlichung von kulturellen Normen, Werten und Haltungen gebraucht.

explorierend erschließen. Die Merkmale der Allgemeinbildungsbegriffe von KLAFKI sowie von BUSSMANN und HEYMANN können auf informatische Modelle und auf das informatische Modellieren von Modellen mit Erfolg angewandt werden, so dass eine Leitlinie »Informatische Modellbildung« als allgemeinbildend gelten darf.

[Thomas 2002, S. 76]

Diese Ergebnisse bestätigen den Bildungswert der informatischen Modellierung als zentrales Element allgemeiner Bildung.

## Basiskonzepte

„Interessant ist, daß die meisten Basiskonzepte schon recht alt sind. Ihre Anzahl scheint sich in einer überschaubaren Größenordnung zu konsolidieren (< ca. 10)“ [Balzert 1996, S. 99]. Die Erstellung von Informatiksystemen wird mit Hilfe der Software-Technik vorgenommen. Um in diesem Zusammenhang Konzepte zuordnen zu können, ist es notwendig, Sichten auf das System zu charakterisieren und zu entscheiden, mit welchem konkreten Konzept die jeweilige Sicht modelliert werden kann. Dabei haben sich die folgenden Sichten als nützlich für die Konstruktion von Informatiksystemen erwiesen: Funktionen, Daten, Dynamik und Benutzungsoberfläche. Damit werden deutlich statische und die dynamische Sichtweisen ausgewiesen und zudem die Interaktion von Benutzerinnen mit Informatiksystemen, die zu einer erheblichen Komplexitätssteigerung bei der Modellierung führen, berücksichtigt.

Wie BALZERT zeigt, können die im Bereich der Software-Entwicklung entwickelten Konzepte auf Basiskonzepte zurückgeführt werden. Das obligatorische Kriterium, das ein Konzept erfüllen muss, um als Basiskonzept für die Software-Entwicklung zu gelten: Das Konzept ist nicht auf andere Basiskonzepte reduzierbar und wird damit als *atomar* bezeichnet. Neben diesem Ausschlusskriterium muss mindestens eins der folgenden Kriterien erfüllt sein: langlebig, phasen-, kontextübergreifend einsetzbar. Mit Hilfe dieser Kriterien werden in der Tabelle 2.2 bezüglich der verschiedenen Sichten die Basiskonzepte zugeordnet (vgl. [Balzert 1996, S. 98ff]).

Funktionale Hierarchie	Informationsfluß	Daten-Strukturen	Entitäten & Beziehungen	Klassen-Strukturen	Kontroll-Strukturen	wenn-dann-Strukturen	Endlicher Automat	Nebenläufige Strukturen	Interaktions-Strukturen
Funktionale Sicht		Datenorientierte Sicht		Objektorientierte Sicht	Algorithmische Sicht	Regelbasierte Sicht	Zustandsorientierte Sicht		Szenariobasierte Sicht

nach [Balzert 1996, S. 98]

Table 2.2: Basiskonzepte und Sichten der Software-Entwicklung

Basiskonzepte können auf verschiedene Art beschrieben werden. Dies reicht von vollständig formalisierten bis zu informalen Beschreibungen, die in den Ausprägungen textuell bis graphisch ihre jeweilige Darstellung finden. An den Beispielen der Konzepte Zustandsautomat und Kontrollstruktur wird dies in der Tabelle 2.3 verdeutlicht.

Für den Einsatz im Bereich der Software-Entwicklung werden mehrere Basiskonzepte zu Konzepten für die Systemanalyse kombiniert und als Methoden ausgewiesen.

Dabei kommt der Methode der objektorientierten Analyse (OOA) eine große Bedeutung zu, da sie im Unterschied zu anderen Methoden inzwischen häufig in der softwaretechnischen Praxis eingesetzt wird. Heide BALZERT weist in dem Lehrbuch zur Objektmodellierung [Balzert 1999] objektorientierte Grundkonzepte aus<sup>75</sup>. Die folgenden Konzeptelemente weisen dabei sowohl statische, wie auch dynamische Konzepte auf.

<sup>75</sup> Sie charakterisiert den Terminus Grundkonzept dadurch, dass es sich hierbei um Konzepte handelt, die in allen Phasen der Softwareentwicklung vorhanden sind (vgl. [Balzert 1999, S. 5ff]).

Basiskonzept	grafische Notation	textuelle Notation
Zustandsautomat	Zustandsgraph (Zustandsautomat)	Zustandsdiagramm Zustandstabelle
Kontrollstruktur	Struktogramm Programmablaufplan (PAP)	Pseudocode

nach [Balzert 1996, S. 103]

Table 2.3: Notationsmöglichkeiten von Basiskonzepten

- *Objekt, Attribut, Botschaft*<sup>76</sup>
- *Klasse, Operation, Vererbung*

Damit ein Fachkonzept modelliert werden kann, sind die genannten Grundkonzepte (nach [Balzert 1999, S. 5]) um Konzepte aus der semantischen Datenmodellierung zu erweitern:

- *Assoziation, Paket*
- *Geschäftsprozess, Szenario*
- *Zustandsautomat*

Für die Umsetzug der Fachkonzepte werden informatische Mittel eingesetzt, die in der Diskussion um die Möglichkeiten der objektorientierten Modellierung oft nicht mehr deutlich herausgestellt werden. Dabei handelt es sich um die Mittel, die zur Strukturierung der Abläufe eingesetzt werden, also um grundlegend algorithmische Strukturen. Darüberhinaus behalten die gefundenen Möglichkeiten zur Datenabstraktion ihre Gültigkeit.

Die unter dem Oberbegriff Algorithmen und Datenstrukturen bekannten Lösungsmuster der Informatik sind also mit der objektorientierten Modellierung nicht obsolet geworden, sie sind „an geeigneter Stelle“<sup>77</sup> als erfolgreiche Muster geeignet zu berücksichtigen.

## Explication von Arbeitshypothesen

Ausgehend von der Fachwissenschaft Informatik wurden die Bandbreite informatischen Denkens aufgezeigt und die Versuche gelungener Abgrenzungen der Fachwissenschaft von anderen Wissenschaften verdeutlicht. Notwendig erscheint angesichts der sich konsolidierenden Informatik die Herausarbeitung der langfristig gültigen Konzepte, die Verbindung mit Fachmethoden, die kognitiven Bedürfnissen (vor allem im Kontext einer allgemeinen Bildung) gerecht werden. Als Schlüsselbegriff kann der Modellierungsbegriff der Informatik identifiziert werden.

Unter dem Dach der Informatik finden sich heute verschiedene, widersprüchliche wissenschaftstheoretisch orientierte Sichtweisen. Begründet ist das einerseits in der erfolgreichen Ablösung von den Disziplinen Mathematik und Elektrotechnik, die die Informatik hervorbrachten, und andererseits in der sich abzeichnenden Brücke zu den Geisteswissenschaften. Erhalten blieb der Charakter einer Strukturwissenschaft wie Mathematik, die aber andere Denkweisen und Methoden hervorbrachte. Neu ist das Verständnis für soziotechnische Systeme und die Anwendung partizipativer Konzepte in der Softwareentwicklung. Darüber hinaus führt die Anwendung der informatischen Modellierung in anderen Wissenschaften, aber

<sup>76</sup> Die Zuordnung zu dynamischen Konzepten wird durch eine *hervorgehobene* Darstellung vorgenommen.

<sup>77</sup> Algorithmen zur Gestaltung von Methoden – Datenstrukturen als Teil von Klassenstrukturen

auch die Nutzung der Informatik als Methodologie zu überraschenden Ergebnissen<sup>78</sup> der wissenschaftstheoretischen Betrachtungen.

In Bezug auf die Fachwissenschaft Informatik können folgende Hypothesen formuliert werden:

### **Arbeitshypothesen**

- Die Fachwissenschaft Informatik unterscheidet sich hinsichtlich des Gegenstandsbereichs und ihrer Methoden von den tradierten Wissenschaften.<sup>79</sup>
- Die informatische Modellierung verändert durch autooperationale Form den modellierten „Welt-ausschnitt“.<sup>80</sup>
- Informatik als dritte Säule der wissenschaftlichen Arbeitsweise neben theoretischem und experimentellem Vorgehen erweitert die tradierten (Natur- und Ingenieur-) Wissenschaften um eine neue Qualität.

---

<sup>78</sup> vgl. Informatik als Methodologie, S. 14

<sup>79</sup> Die Gegenstände der Informatik sind wie bei Philosophie und Mathematik immateriell. Darüber hinaus ist erkennbar, dass sie mit ihren Methoden die Lücke zwischen Anwendungsbereich und Formalisierung schließt.

<sup>80</sup> Sie ist insofern nicht mit anderen Wissenschaften vergleichbar. Informatik bedarf der Anwendung als Bezugspunkt.



Pädagogik [...] muß die Umstände annehmen, wie sie tatsächlich sind. [...] Wie dies zu realisieren ist, hängt davon ab, wie] Pädagogik in die Lage versetzt, didaktische Prozesse auf die volle Wirklichkeit anzulegen, die Gesetzmäßigkeiten gesellschaftlicher Bewegungen und ökonomischer Veränderungen von der Erscheinungsebene auf ihr Wesen zurückzuführen und damit die zentrale Aufgabe von Wissenschaft wahrzunehmen.

[Gamm 1983, S. 195f]

## Kapitel 3

# Lerntheoretische Grundlagen

Um der Zielsetzung dieser Arbeit gerecht werden zu können, ist es unabdingbar, einen fokussierenden Blick auf wesentliche Positionen der Erziehungswissenschaft<sup>81</sup> zu werfen, um so für den Kontext dieser Forschungsarbeit Schlussfolgerungen zu ziehen.

## Funktionen der Schule – nach herrschender Meinung

Ausgehend vom gesellschaftlichen Steuerungsinteresse erfüllt die Institution Schule (nach [Fend 1974], [Hurrelmann 1975]) folgende Funktionen:

### Qualifikation

- Allgemeine und fachliche [Aus-]Bildung<sup>82</sup> für die Gesellschaft

### Allokation/Selektion

- Entscheidung über Sozial- und damit Lebenschancen in der Gesellschaft
- Auslese und Verteilung der jeweils Geeigneten

### Sozialisation/Integration/Legitimation

- Eingliederung in die jeweilige Gesellschaftsordnung
- Ermöglichung des gemeinsamen Lebens durch Anpassung in der Gesellschaft
- Rechtfertigung der Gesellschaftsordnung

Auch wenn diese Klassifikation nicht unumstritten ist,<sup>83</sup> wurde sie von anderen Erziehungswissenschaftlerinnen übernommen. Bis heute kann sie als brauchbare strukturelle Beschreibung institutionell gebundener [Aus-]Bildung betrachtet werden.

<sup>81</sup>

- Die Begriffe Erziehungswissenschaft und Pädagogik werden in der Literatur unterschiedlich definiert. Pädagogik ist nach dem griechischen Wortstamm die Wissenschaft von der Erziehung. Erziehungswissenschaft kann im Sinne von eher empirischer Wissenschaft von der Pädagogik abgegrenzt werden.
- Erziehungswissenschaft wird hier als Querschnittswissenschaft gesehen – daher wird der Plural "Erziehungswissenschaften" nicht verwendet.

<sup>82</sup> Der Bildungsbegriff wird in den letzten Jahrzehnten kontrovers diskutiert. Wolfgang KLAFKI rückt die diesem Begriff durchaus innewohnenden Potenziale mit einer emanzipatorischen Zielrichtung (vgl. [Klafki 1985a]) in den Mittelpunkt seiner kritisch-konstruktiven Didaktik [Klafki 1985b, S. 31–86]. Der Begriff der Ausbildung ist vom Begriff der Bildung insofern abzuheben, als Ausbildung immer auch das Element der Professionalisierung für eine berufliche Tätigkeit zu berücksichtigen hat. Der entscheidende Unterschied zwischen Bildung und Ausbildung liegt dabei in der Orientierung des Prozesses auf das Subjekt (Person, Kräfte der Lernenden) bzw. auf das Objekt (Inhalt, Sache, Wissenschaft).

<sup>83</sup> Beispielsweise wird bei der Klassifikation der historische Ansatz und damit der Aspekt eines möglichen Wandels ausgeklammert.

Die Grundlage der Art und Weise des Unterrichts wird durch subjektive Theorien oder (latente) Leitbilder vom Lehren und Lernen sowie durch die Position und die Funktion der Lehrenden und Lernenden im Unterricht<sup>84</sup> bestimmt. In diesem Zusammenhang wird auch von Paradigmen des Lehrens und Lernens<sup>85</sup> oder Unterrichtsphilosophien<sup>86</sup> gesprochen. Die Charakterisierung verschiedener Selbstbilder der Lehrerinnen sollte eher als Typen des Lehrens und Lernens bezeichnet werden. Im Unterrichtsalltag stellen sie die Grundlage dar, auf der praktische Entscheidungen über didaktische Prinzipien und Methoden des Unterrichts getroffen werden.

Das individuelle Interesse der Schülerinnen<sup>87</sup> kann durchaus gegen den auf sie ausgeübten Selektionsdruck gerichtet sein. Gerade in der Phase der Pubertät und der anschließenden Adoleszenz opponieren Schülerinnen vor allem gegen die integrative Tendenz der Erziehung. Darüber hinaus wurde gezeigt, dass Lehrerinnen sich gewissen institutionellen Funktionszuweisungen verschließen (vgl. [Combe und Buchen 1996]).<sup>88</sup>

Die gesellschaftliche Bedeutung spiegelt sich in dem Bemühen, die mit den Funktionen und Problemen zusammenhängenden Fragestellungen zu analysieren und über die Stufen Nachdenken (= pädagogische Weisheit), Theorie (= Philosophie der Erziehung und systematische Pädagogik) und Wissenschaft (= Bildungsökonomie und Erziehungstechnologie) planen zu können (wie Herwig BLANKERTZ ausführt – vgl. [Blankertz 1980b]). Im Folgenden wird davon ausgegangen, dass die Institution Schule den gesellschaftlichen Auftrag hat, die oben beschriebenen Funktionen unter den Zielsetzungen: Einweisung in Lebensperspektiven sowie der Befähigung zur Selbstständigkeit, Urteilsfähigkeit und sozialem Verhalten<sup>89</sup> einzulösen. Mit der Institution Schule wird Lernen aus dem allgemeinen Lebensvollzug ausdifferenziert und in das Subsystem Schule verlagert.

## Phasenunterteilung/Phasierung des Unterrichts

Die Darstellung der Geschichte folgt gängigen Didaktiken (vgl. exemplarisch [Meyer 1988, S. 155–240] – Stufen- und Phasenschemata des Unterrichts – Ordnungsversuche). Überlegungen zu allgemeinen Gesetzmäßigkeiten für das Lernen und Unterrichten haben eine lange Tradition.<sup>90</sup> Als Reaktion auf die Ende des 19. Jahrhunderts geforderte Standardisierung des Volksschulunterrichts und der Lehrerbildung wurde auf der Basis der lerntheoretischen Elemente der Pädagogik von Johann Friedrich HERBART<sup>91</sup> [Herbart 1913, S. 23f] das Konzept der Formalstufen für eine Lehrtheorie entwickelt. In der geisteswissenschaftlich orientierten Pädagogik wurde diese Theorie zu einer Unterrichtsstruktur ausdifferenziert [Rein 1893, S. 34], die für alle Unterrichtsinhalte gültig sein sollte.<sup>92</sup> In der Reformpädagogik wurden diese Überlegungen grundlegend kritisiert und sowohl schülerorientierte, handlungsbezogene und flexiblere Alternativen<sup>93</sup> propagiert, als auch die vollständige Ablehnung von Stufenkonzepten zum Ausdruck gebracht (vgl. Berthold OTTO mit dem Konzept des „Gesamtunterrichts“ [Otto 1913]).

Mit dem Begriff „Arbeitsschule“ werden im Laufe der Zeit verschiedene Ansätze bezeichnet. Mit diesen Konzeptionen wird die Kritik an den Stufenkonzepten konstruktiv formuliert.

<sup>84</sup> aber auch in der Schule und in der Gesellschaft

<sup>85</sup> vgl. [Dubs 1995] – hier handelt es sich nicht um Paradigmen, wie sie in Kapitel 1, S. 1 („Definition“ des Begriffs nach KUHN) dargestellt wurde.

<sup>86</sup> vgl. [Mandl und Reinmann-Rothmeier 1995]

<sup>87</sup> Im Folgenden wird neben dem Begriff *Schülerin(nen)* auch der Begriff *Lernende(n)* verwendet.

<sup>88</sup> Im Zusammenhang mit konstruktivistischen Grundorientierungen (siehe S. 39) werden diese Punkte zu berücksichtigen sein.

<sup>89</sup> Zielsetzungen, die aus der demokratischen Verfasstheit erwachsen

<sup>90</sup> Johann Amos COMENIUS prägte 1638 mit der »Didactica magna« den Begriff Didaktik und schuf damit die Grundlagen für das Bemühen, mit Hilfe eines gestuften Aufbaus des Unterrichtsprozesses durch die Lehrerin den Schülerinnen das Lernen zu erleichtern.

<sup>91</sup> Artikulationsschema – der Begriff wird bis heute verwendet, um die sinnvolle Gliederung (Artikulation) der bei den Schülerinnen vermuteten Lernphasen zum Ausdruck zu bringen. Andere Autoren verwenden auch die Begriffe Grundformen oder Choreographie (vgl. [Aebli 1998])

<sup>92</sup> Wie von Hilbert MEYER dokumentiert, wurde von den Herbartianern darüber hinaus eine Theorie der Kulturstufen und als dritte Säule ihrer Unterrichtstheorie das Konzentrationsprinzip zur Sicherung des Sittlichen im Bildungsprozess entwickelt (vgl. [Meyer 1988, S. 170]).

<sup>93</sup> Exemplarisch sind hier Konzepte der „Arbeitsschule“ zu nennen.

- Die Reformpädagogen Hugo GAUDIG und Georg KERSCHENSTEINER bezeichneten ihre unterschiedlichen Konzeptionen als „Arbeitsschule“ (vgl. [Gaudig 1921], [Kerscheneiner 1927f]). Als Kontrast zu den „Buchschulen“, die ausschließlich abstraktes Wissen vermitteln, forderten sie übereinstimmend, Wissen und Fertigkeiten vor allem durch Handeln in authentischen Kontexten zu erwerben. Damit strebten sie an, möglichst viel geistige Arbeit mit manueller Arbeit zu verknüpfen und auf diese Weise authentische Aktivitäten im Rahmen der Schule zu fördern. Damit kann das Konzept der „Arbeitsschule“ als historischer Vorläufer für eine konstruktivistische Unterrichtsgestaltung (vgl. S. 39ff) betrachtet werden.
- Im nachrevolutionären Russland wird von Pavel Petrovič BLONSKIJ 1918 eine „Arbeitsschule“ konzipiert, die sich an dem Arbeitsbegriff von Karl MARX orientiert. Das Vorbildhafte an BLONSKIJS „Arbeitsschule“ sind die Idee des (werk-)tätigen Menschen als Ausgangspunkt für curriculare Überlegungen (Vereinigung von Kultur und Produktion) und die Vorschläge für eine Schule, die Wohn-, Arbeits-, Vergnügungs- und damit Lebensraum ist (vgl. [Frey 1998, S. 54]).

Aus heutiger Sicht kann das Konzept eines fächer-, inhalts- und institutionsneutralen, allgemein gültigen Phasenschemas nicht aufrechterhalten werden, da (laut [Meyer 1988, S. 178])

- keine für ein solches Vorhaben notwendig vorauszusetzende allgemeine Lern- und Lehrtheorie existiert<sup>94</sup> und
- zwischen Zielen, Inhalten und Methoden des Unterrichts komplexe Wechselwirkungen bestehen.<sup>95</sup>

Diese Feststellungen machen deutlich, warum inzwischen führende allgemeine Didaktiker in der Bundesrepublik auf die Vorstellung eigener Stufen- oder Phasenmodelle verzichten. Unterricht ist ein überaus komplexes Geschehen, in dem jede Theoriebildung bestimmte, ausgewählte Schwerpunkte setzt (vgl. „Primat der ...“ – Fußnote 95). Damit erweisen sich theoriegeleitete Phasenmodelle für die Planung und Gestaltung des Unterrichts unter bestimmten Bedingungen als nützlich. So sind bis heute Phasenschemata zur Unterrichtsplanung verbreitet.<sup>96</sup> Als Gliederungshilfen stellen sie im Zusammenhang mit der Entwicklung professionellen Lehrerinnenhandelns ein nicht zu unterschätzendes Hilfsmittel dar. Sie liefern ein Gerüst, das als anfängliche Hilfe für komplexe Planungsprozesse eine handhabbare Unterstützung bietet. Planungsstrukturhilfen dürfen dabei nicht zu einer Zwangsstruktur degenerieren, nach der Lehrerinnenhandeln stattzufinden hat.

### Exkurs: Lernobjekte

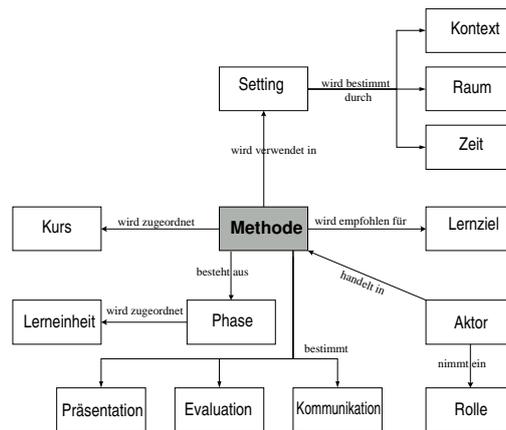
Zur Gestaltung von Lernumgebungen mit Informatiksystemen sind in den letzten Jahren konkurrierende Standards entwickelt worden. Die bekanntesten sind Sharable Content Object Reference Model (SCORM) [Dodds 2001] und Learning Object Metadata (LOM) [IEEE LTSC 2002]. Von diesen Standards wird erwartet, dass mit ihrer Hilfe „Lernobjekte“ systemunabhängig erstellt, verteilt und genutzt werden können. Das Ziel der Standardisierungsbemühungen besteht darin, die Interoperabilität von Inhalten zwischen verschiedenen Informatiksystemen zu gewährleisten, die als „Lernplattformen“ eingesetzt werden sollen. In diesem Kontext finden sich Vorschläge zur Erweiterung der Standards um Elemente, die als „Modellierung didaktischer Konzepte“ bezeichnet werden (vgl. den gleichnamigen Titel [Pawlowski 2002]).

Wie in Abbildung 3.1 herausgestellt wird, geht mit diesen modellierten Lernobjekten immer auch eine Strukturüberlegung einher: „Die Beschreibung einer Methode besteht aus den Hauptelementen Setting, Phase, Präsentation, Kommunikation und Evaluation“ [Pawlowski 2002, S. 372]. Die Darstellung macht deutlich, dass mit dieser Modellierung eine normierende Wirkung verbunden sein kann. Standards, denen eine spezifische Vorstellung des Lernens und Lehrens zu Grunde liegen, dienen als Grundlage für technisch unterstützte Lernprozesse. In der Anwendung werden die Gestaltungsmöglichkeiten dieser Prozesse durch solche Modelle reglementiert. Bemerkenswert erscheint darüber hinaus, dass nicht die Schülerin (hier als „Aktor“, der eine Rolle einnimmt, bezeichnet), sondern die Methode im Zentrum der Überlegungen steht.

<sup>94</sup> Die Problemlage ist bis heute unverändert: „Es gibt zur Zeit keine Theorie des Lehrens!“ [Loser und Terhart 1977, S. 11]

<sup>95</sup> Allerdings unterscheiden sich die Didaktiken bei allem Festhalten an der Interdependenz in der Frage, wem das Primat einzuräumen ist. Der klassischen Position, dem Primat der Ziele, stellt Hilbert MEYER das Primat der Methode (vgl. [Meyer 1997, S. 159]) entgegen.

<sup>96</sup> exemplarisch: „Rezept für die Ausführung einer Unterrichtsstunde“ – Kapitelüberschrift in [Grell und Grell 1996, S. 103ff]



aus [Pawlowski 2002, S. 373]

Abbildung 3.1: Datenschema Methode

### Exkurs: Schulfächer – Arbeitsformen

Über den gesellschaftlichen Aushandlungsprozess, ein Schulfach zu etablieren, soll an dieser Stelle nicht reflektiert (und lamentiert) werden (vgl. [Meyer 1988, S. 78ff] – Die Herkunft der Unterrichtsinhalte). Vielmehr soll der Blick auf eine begriffliche Vielfalt gelenkt werden, die im Folgenden an einigen Stellen aufscheint und nicht ohne eine Vereinbarung über die Terminologie verständlich wird (vor allem, wenn Abgrenzungen vorgenommen werden). Zur Klärung der Begriffsvielfalt im Zusammenhang mit dem Präfix *fächer* sind in Tabelle 3.1 weiter verbreitete Formen des fächerübergreifenden Unterrichts angegeben.

Begriff	Charakterisierung	Voraussetzung
fächerüberschreitend	von der Fachlehrerin wird über die Grenzen des Faches auf übergreifende Themen verwiesen	weiter Horizont der Lehrerin, keine unterrichtsorganisatorischen Vorkehrungen
fächerverknüpfend	Verweise wechselseitig in Kenntnis dessen, was im jeweils anderen Fach wann im Unterricht behandelt wird	Unterrichtsplanung, die eine Übersicht über alle Fächer einer Klasse oder Stufe einschließt und Kontakt zwischen den Lehrerinnen
fächerkoordinierend	Unterricht mehrerer Fächer in der Planung aufeinander bezogen, aber weiterhin getrennt durchgeführt durch Fachlehrerin	Teilnahme derselben Schülerinnen an diesen Kursen (mindestens zum größten Teil), gemeinsame Planung

nach [Hildebrandt 2002, Folie 6]<sup>97</sup>

Tabelle 3.1: Formen fächerübergreifenden Unterrichts

<sup>97</sup> Die Begriffe fächerergänzend und fächeraussetzend aus [Hildebrandt 2002, Folie 6] werden hier nicht aufgeführt, da sie nicht etabliert sind und in den weiteren Ausführungen nicht benötigt werden.

In der Tabelle fehlt der Begriff *fächerverbindend*, dem (in Nordrhein-Westfalen) ein eigener Abschnitt der Richtlinien für die gymnasiale Oberstufe gewidmet ist: „So wichtig es ist, durch systematische fachliche Arbeit fachliche Kompetenzen zu fördern, so bedeutsam ist es, die Fachperspektive zu überschreiten. Durch fachübergreifendes und fächerverbindendes Lernen wird eine mehrperspektivische Betrachtung der Wirklichkeit gefördert, und es werden damit auch übergreifende Einsichten, Fähigkeiten, Arbeitsmethoden und Lernstrategien entwickelt, die unterschiedliche fachliche Perspektiven für gemeinsame Klärungen und Problemlösungsstrategien verbinden und so zur Kenntnis der komplexen und interdependenten Probleme der Gegenwart beitragen. Deshalb gehört das Überschreiten der Fächergrenzen, das Einüben in die Verständigung über Differenzen und über Differenzen hinweg neben dem Fachunterricht zu den tragenden Prinzipien der gymnasialen Oberstufe. Wissenschaftspropädeutisches Lernen erfordert beides: das fachliche Arbeiten, seine Reflexion und das Denken und Handeln in fachübergreifenden Zusammenhängen“ [MSWWF 1999, S. XVIII].

## Von der Kritik am Primat der Instruktion<sup>98</sup>

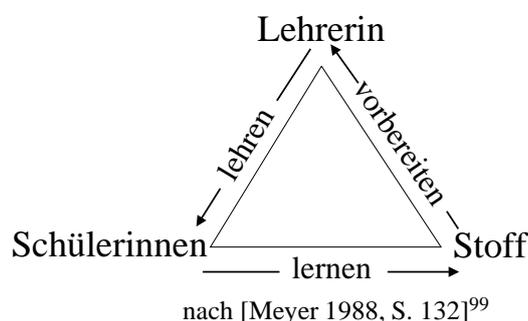


Abbildung 3.2: »Didaktisches Dreieck«

Schulisches Lehren und Lernen ist heutzutage typischerweise in unterrichtlichen Kontexten organisiert. In diesem Unterricht wird den Lehrerinnen der aktive und den Schülerinnen eher der passive Teil zugeschrieben. Bei dieser Art des Unterrichts wird<sup>100</sup> vom Primat der Instruktion ausgegangen: Die Anstrengungen der Lehrerin bestehen darin, zu entscheiden, wie der Unterricht geplant, organisiert und gesteuert werden soll, damit die Schülerinnen die präsentierten Inhalte in ihrer Systematik verstehen, sich diese Inhalte entsprechend zu eigen machen und somit Lernerfolg im Sinne vorher definierter Lehr-/Lernziele erlangen. Die Lehrerin präsentiert und erklärt die Inhalte, leitet die Schülerinnen an und stellt ihre Lernfortschritte sicher. Die Schülerin befindet sich in einer eher passiven Position.

Wie stark die Planungshoheit der Lehrerin Eingang in die lerntheoretisch orientierten Didaktiken<sup>101</sup> gefunden hat, spiegelt sich in der folgenden Aussage: „Dabei ist grundsätzlich die Totalerfassung aller im Unterrichtsgeschehen wirksamen Faktoren angestrebt“ [Heimann u. a. 1970, S. 9]. Zentrale Strukturierungseinheit in diesem Modell stellen Hierarchiestufen dar, die zwar als interdependent bezeichnet werden, aber dennoch in Hierarchieform präsentiert werden (vgl. Abbildung 3.3).

„Unterrichtsplanung ist immer Reduktion von Komplexität und so qua Definition nicht total planbar“ [Görllich, persönliche Mitteilung vom 17. November 2002]. Da die vollständige Planbarkeit des Unterrichts nicht erreichbar ist, werden einige Elemente in dem »Berliner Modell« speziell gestaltet. So soll der „Forderung nachzukommen versucht [werden] alle den Unterricht konstituierenden Momente als im Verhältnis wechselseitiger Abhängigkeit stehend zu behandeln (Prinzip der Interdependenz) [und] wegen der prinzipiellen Unvorhersehbarkeit von Schülerreaktionen in der Planung mehrere Verlaufsmöglichkeiten vorzusehen (Prinzip der Variabilität) [. Die Unterrichtsplanung ist] so zu gestalten, daß das Maß seiner Erfüllung überprüft werden kann (Prinzip der Kontrollierbarkeit)“ [Heimann u. a. 1970, S. 11]. Die grundlegenden Annahmen der Instruktion als traditioneller Unterrichtsphilosophie lassen sich thesenartig zusammenfassen (nach [Mandl und Reinmann-Rothmeier 1995, S. 11], vergleiche auch [Schulmeister 2002, S. 125]):

<sup>98</sup> Eine pointierte Erklärung des Begriffs Instruktion: „den Schülern etwas beibringen und dafür zu sorgen, daß sie möglichst viel Schulstoff lernen“ [Grell und Grell 1996, S. 49].

<sup>99</sup> Bei Meinert MEYER wurde der Begriff Stoff inzwischen durch Erfahrung ausgetauscht (persönliche Mitteilung von Christian GÖRLICH, 17. November 2002).

<sup>100</sup> trotz verschiedener theoretischer Überlegungen und Unterrichtsmodelle, z. B.

- lernzielorientierte Didaktik
  - lern- und lehrtheoretische Didaktik
  - bildungstheoretische und kritisch-konstruktive Didaktik
- siehe [Meyer 1988, S. 24]

<sup>101</sup> lerntheoretische Didaktik (= »Berliner Modell«) nach HEIMANN, OTTO und SCHULZ (vgl. [Heimann u. a. 1970]) – durch den Wechsel von Wolfgang SCHULZ nach Hamburg in seiner weiteren Entwicklung später als lehrtheoretische Didaktik (= »Hamburger Modell«) bezeichnet.

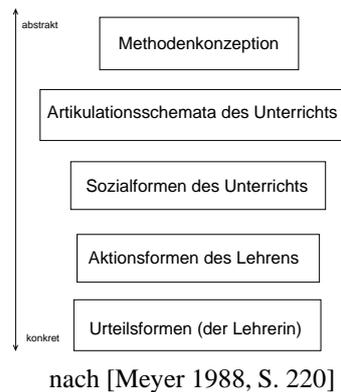


Abbildung 3.3: Hierarchiemodell nach Wolfgang SCHULZ

- Lerninhalte sind Wissenssysteme, die in ihrer Entwicklung abgeschlossen und klar strukturierbar sind.
- Lernen erfolgt systematisch und vorwiegend linear.
- Lernen ist vor allem ein rezeptiver Prozess.
- Bei der Unterrichtsgestaltung stehen Fragen der Instruktion im Vordergrund.
- Die Schülerinnen müssen von außen stark angeleitet und kontrolliert werden.
- Die Lehrende fungiert als Mediator und präsentiert neue Inhalte.
- Bewährte Unterrichtsformen können unabhängig von Inhalt, Kontext, Zeitpunkt und Personenmerkmalen wiederholt angewendet werden.
- Die Unterrichtsergebnisse werden als vorhersagbar angenommen.<sup>102</sup>
- Ziel des Unterrichts: Die Schülerinnen erfüllen die gesetzten Leistungskriterien.

Das daraus abgeleitete instruktionale Vorgehen weist eine klare Struktur auf: Wissen ist systematisch und damit in einer für die Schülerinnen übersichtlichen und nachvollziehbaren Art zu vermitteln. Diese Systematik und Übersichtlichkeit der traditionellen Unterrichtsphilosophie wirft allerdings Probleme auf. Eine ausführliche Darstellung mit Belegen für die hier stichwortartig beschriebenen Probleme findet sich in [Mandl und Reinmann-Rothmeier 1995, S. 10f].

### Empirische Probleme

Es fehlt an empirischen Belegen dafür, dass die Effekte einzelner Instruktionketten wiederholbar sind. Im Rahmen traditioneller Unterrichtsmodelle werden isolierte Lernmechanismen postuliert, die in dieser Form in der Praxis nicht analysierbar sind.

### Theoretische Probleme

Ganzheiten werden in elementare Teile zerlegt und dann getrennt voneinander vermittelt. Verstehen ist von der gesamten Wissensstruktur und nicht von isolierten Teilen dieser Struktur abhängig. Traditionelle Instruktionstheorien bieten konkrete Verfahrensvorschriften für die Auswahl einzelner Unterrichtsmethoden an. Diese Methodenwahl baut auf der Annahme auf, die Wirkung einzelner Methoden könne vorhergesagt werden. Diese Annahme zur Vorhersagbarkeit der Wirkung ist nicht haltbar.

<sup>102</sup> Seiteneffekte werden dabei nicht beachtet.

### Praktische Probleme

Das Primat der Instruktion bedingt eine weitgehend rezeptive Haltung der Schülerinnen. Durch den daraus folgenden Mangel an Aktivität und Eigenverantwortung für den Prozess und Erfolg des Lernens bleiben die Schülerinnen passiv und sind – wenn überhaupt – extrinsisch motiviert. Das hat entsprechend ungünstige Folgen für das Lernen, das ja vor allem dann erfolgreich ist, wenn es auf intrinsischer Motivation beruht. Ein weiteres Kennzeichen besteht darin, dass das Lernen losgelöst von einem relevanten Kontext stattfindet. Das sachlogisch aufbereitete Wissen in der Lehr-/Lernsituation hat mit den komplexen und wenig strukturierten Anforderungen und Erfahrungen in Alltagssituationen wenig gemein. Damit produziert Unterricht nur sogenanntes „träges“ Wissen – ein Wissen, das zwar erworben, aber in realen Situationen nicht angewendet wird.

## 3.1 Didaktische Grundorientierungen

Ausgehend vom Unterricht als organisiertem Lernprozess stellt sich u. a. die Frage nach Theorien, die dem Lernen zu Grunde liegen.<sup>103</sup> Jeder der entwickelten theoretischen Ansätze beleuchtet bestimmte ausgewählte Aspekte des Lernens. Überlegungen zu solchen Theorien haben Konsequenzen für die Gestaltung von Lehrprozessen.<sup>104</sup> Daher werden im Folgenden die behavioristische, die kognitivistische und die konstruktivistische didaktische Grundorientierung hinsichtlich ihrer zentralen Aussagen vorgestellt. Die Darstellung fokussiert auf zentrale, kennzeichnende Elemente und soll damit deutlich machen, dass neben der didaktischen Grundorientierung des Konstruktivismus andere grundsätzliche Überlegungen bedeutsame Beiträge für das Forschungsfeld liefern.

### Behaviorismus

Prominentester radikaler Vertreter der Übertragung behavioristischer Annahmen auf die Unterrichtspraxis ist Burrhus Frederic SKINNER [Skinner 1974], der in der Theorie des operanten Konditionierens die entscheidende Erklärung allen Verhaltens sieht. Zur Vervollkommnung der Strategie, jeden einzelnen Lernschritt systematisch zu verstärken, schlug er den programmierten Unterricht vor, in dem die Lehrende [streckenweise] durch Lehrprogramme und „Lehrmaschinen“ ersetzt wird.<sup>105</sup> Die Behavioristen gehen davon aus, dass jedes Lernen an die Konsequenzen von Verhalten gebunden ist.<sup>106/107</sup>

### Kognitivismus

Die moderne kognitive Psychologie konturierte sich in dem Zeitraum von 1950–1970. John R. ANDERSON [Anderson 2001, S. 10ff] nennt drei Einflussfaktoren für die Wiederbelebung der kognitiven Psychologie.

1. Forschungen zur Leistungsfähigkeit und Leistungsausführung von Menschen in Verbindung mit Untersuchungen der Ideen zur Informationstheorie

<sup>103</sup> Mit dieser Sicht ist eine Einschränkung der beschriebenen Gesamtsicht unterrichtlicher Prozesse verbunden, da so ein Aspekt besonders hervorgehoben und untersucht wird.

<sup>104</sup> Bestätigt wird diese Einschätzung durch das Zitat in Fußnote 94. Auf ein Problem in diesem Kontext muss allerdings hingewiesen werden: aus einer deskriptiven (Lern-)Theorie kann keine präskriptive (Lehr-)Theorie abgeleitet werden (vgl. [Schulmeister 2002, S. 137]), d. h. selbst wenn es gelänge, eine allgemein akzeptierte Lerntheorie zu finden, so bliebe der Schritt zu einer Lehrtheorie noch zu leisten.

<sup>105</sup> Besondere Beachtung hat die von CHOMSKY vehement bestrittene Übertragungsmöglichkeit von im Labor durchgeführten Tierexperimenten auf menschliches und hier insbesondere auf verbales Verhalten gefunden (vgl. [Chomsky 1959], [Chomsky 1967]).

<sup>106</sup> Die wissenschaftstheoretische Grundposition steht den von dem Biologen Benjamin S. BLOOM entwickelten Taxonomien des Lernens [Bloom 1956] nahe und hat eine große Affinität zum Kritischen Rationalismus (Sir Karl R. POPPER [Popper 1972]). Seine Einflüsse auf die Pädagogik reichen im deutschsprachigen Raum von der Kybernetischen Pädagogik [Frank 1969] über die Lernzielorientierte Didaktik [Möller 1973] bis zur Lerntheoretischen Didaktik (Berliner Modell) [Heimann u. a. 1970].

<sup>107</sup> Darüber hinaus soll nicht vergessen werden, dass Ludwig WITTGENSTEIN (der „frühe“ WITTGENSTEIN: »Tractatus logico-philosophicus«) und der Wiener Kreis die wissenschaftstheoretischen Grundpositionen und damit eine philosophische Grundlage dieser Überlegungen stützen.

2. Entwicklungen der Künstlichen Intelligenz (KI)
3. Einfluss des Linguisten Noam CHOMSKY mit seinen Untersuchungen zur Komplexität der Sprache, die mit den behavioristischen Ansätzen nicht erklärt werden kann (vgl. Fußnote 105)

Jean PIAGET<sup>108</sup> und Jerome S. BRUNER<sup>109</sup> verfolgten für die moderne kognitive Psychologie wichtige Ideen. Lernen beruht danach auf kognitiven Strukturen und wird durch kognitive Konzepte des Individuums repräsentiert. Der Lernprozess wird als permanente Anpassungsleistung interpretiert, bei der erworbene Konzepte an veränderte Gegebenheiten angepasst werden, um damit ein dynamisches Gleichgewicht<sup>110/111</sup> herzustellen. Damit stellen diese Theorien der kognitiven Psychologie sowohl die Grundlage für den Kognitivismus, als auch für den Konstruktivismus dar.

**Exkurs: Fundamentale Ideen → Spiralprinzip, Repräsentationsstufen (BRUNER)**

Wie bereits in Kapitel 1 (Abschnitt Feststellungen zum Problemfeld – Motivation, S. 3) dargestellt wurde, fordert BRUNER, dass fundamentale Ideen jeder Schülerin auf jeder Entwicklungsstufe in einer angemessenen Form näher gebracht werden. Daraus leitet BRUNER zwei Forderungen ab:

**Spiralprinzip** Im Laufe der [Schul-]Zeit sollte immer wieder auf die fundamentalen Ideen zurückgekommen werden – unter verschiedenen Gesichtspunkten und auf verschiedenen Niveaus.

**Repräsentationsmodell** Bei der Vermittlung der fundamentalen Ideen, vor allem aber bei der Erstbegegnung mit einem Sachverhalt, ist das intuitive Denken und Verstehen der Schülerin zu berücksichtigen.

Der Mathematikdidaktiker Erich WITTMANN leitet aus dem spiralförmigen Curriculaufbau zwei Prinzipien ab:

„Prinzip des vorwegnehmenden Lernens

Die Behandlung eines Wissensgebietes soll nicht aufgeschoben werden, bis eine endgültig-abschließende Behandlung möglich erscheint, sondern ist bereits auf früheren Stufen in einfacher Form einzuleiten [ . . . ].

Prinzip der Fortsetzbarkeit

Die Auswahl und Behandlung eines Themas an einer bestimmten Stelle des Curriculums soll nicht ad hoc, sondern so erfolgen, daß auf höherem Niveau ein Ausbau möglich wird. Zu vermeiden sind vordergründige didaktische Lösungen, die später ein Umdenken erforderlich machen.“

[Wittmann 1981, S. 86]

Im Repräsentationsmodell werden die Informationsaufnahme, -verarbeitung und -speicherung klassifiziert: BRUNER unterscheidet die Stufe der Handlung (*enaktiv*), der bildhaften Wahrnehmung (*ikonisch*) und der Sprache (*symbolisch*) (vgl. [Bruner 1974, S. 48–52], [Klafki 1985b, S. 103]). Dabei ist zu berücksichtigen, dass diese Stufen aufeinander aufbauen. Damit wird auf kognitionspsychologischer Basis Kritik an der „Buchschule“ geübt, die mit der abstrakten Stufe beginnt.

<sup>108</sup> „PIAGET sollte weder als Entwicklungspsychologe noch als Lernpsychologe missverstanden werden. Es ging ihm um die Aufklärung der Entwicklung menschlicher Erkenntnis [ . . . ]“ [Häußler u. a. 1998, S. 183]. Er wurde durch die Beschreibung eines differenzierten Modells der Entwicklung und des Lernens bei Kindern bekannt (Entwicklungsstufenmodell [Piaget 1948]). Seine grundlegenden Überlegungen wurden von den Konstruktivisten aufgenommen und wird von ihnen als Bestätigung gegen die traditionellen Lerntheorien der Behavioristen und der Kognitivisten angesehen (vgl. S. 39f).

<sup>109</sup> Populär wurde BRUNER mit dem Konzept des »Entdeckenden Lernens« (vgl. Abschnitt 3.2, Fußnote 139f, S. 47). Seine Rezeption innerhalb der Informatikdidaktik verdankt er den Arbeiten von Andreas SCHWILL, der die »Fundamentalen Ideen« BRUNERS zu einer wichtigen Grundlage seiner Überlegungen machte (vgl. Abschnitt 4.2.1, S. 63).

<sup>110</sup> In dem Wechselspiel zwischen äußeren Ereignissen sowie Erfahrungen und den individuell entwickelten Schemata kommt es immer wieder zu einem Ungleichgewicht.

<sup>111</sup> Die von Piaget verwendete Begrifflichkeit wird in der Fußnote 117 (S. 40) erläutert.

In der kognitivistischen Grundorientierung werden die Schülerinnen als Individuen betrachtet, die äußere Reize aktiv und selbstständig verarbeiten und (im Unterschied zum Behaviorismus) nicht passiv durch sie gesteuert werden. Lernen wird damit als aktiver Prozess des Individuums verstanden, der zu einer Repräsentation des Wissens beim Individuum führt. Der Lernprozess besteht in der Bedeutung, durch die diese symbolische Darstellung an das Gedächtnis übergeben wird. Damit wird der individuellen „Verarbeitung“ durch die Schülerin beim Lernen eine hohe Bedeutung zugemessen. Dennoch gehen die Ansätze von starken Wechselwirkungen zwischen internen Verarbeitungsprozessen und externen Präsentationen aus. Es handelt sich damit um ein Ein-/Ausgabe-Modell im Sinne der Verarbeitung von Symbolen.<sup>112</sup> Hieraus wird abgeleitet, dass das Lernen durch Instruktion und Lernhilfen im Sinne des Kognitivismus nicht nur angeregt und unterstützt, sondern auch begrenzt gesteuert werden kann.

Prominente Vertreter: der Schweizer Biologe und Psychologe Jean PIAGET, der Russe Lev S. VYGOTSKY und der US-Amerikaner Robert M. GAGNÉ.<sup>113</sup>

## Konstruktivismus

Als Hauptvertreter des »radikalen Konstruktivismus«<sup>114</sup> gilt Ernst VON GLASERSFELD (vgl. [Fischer 1996, S. 7], [von Glasersfeld 1997a]). Der Konstruktivismus ist primär eine Erkenntnistheorie, die davon ausgeht, dass „Wirklichkeit“ nicht von sich aus vorhanden und damit zugänglich ist, sondern vom Individuum konstruiert wird. Seine Entstehung ist im Kontext der Forschungen der Biologen Francisco J. VARELA und Humberto Maturana anzusiedeln (vgl. [Maturana und Varela 1992]). Als Konsequenz des Konstruktivismus muss „die Suche nach unwandelbaren, objektiven Wahrheiten“ [von Glasersfeld 1989, S. 173] aufgegeben werden. Für organisierte Lernprozesse wird „der Glaube an objektives, naturwissenschaftliches Wissen, ein Glaube, der bislang als unzweifelbare Grundlage fast aller Ausbildung in Schulen und Hochschulen diente, [...] erschüttert“ [von Glasersfeld 1989, S. 172]. In dem hier thematisierten Zusammenhang wird der Begriff Wissen auf konstruktivistischer Grundlage näher beleuchtet. Von den Konstruktivisten „wird verlangt, daß die begrifflichen Konstrukte, die wir Wissen nennen, sich in der Erfahrungswelt des erkennenden Subjekts als viabel<sup>115</sup> erweisen“ [von Glasersfeld 1989, S. 173].<sup>116</sup>

Die für Lehr-/Lernprozesse bedeutsamen Elemente stellt VON GLASERSFELD in einer Zusammenfassung vor:

- der Erwerb von Fertigkeiten, d.h. von Handlungsmustern [ist] klar von der aktiven Konstruktion viabler begrifflicher Netzwerke, also vom Verstehen [zu unterscheiden. . . .]
- die alten methodischen Hilfsmittel des Auswendiglernens und des Wiederholens im Training [behalten] ihren Wert, es wäre jedoch naiv zu erwarten, daß sie auch das Verstehen befördern. [ . . . ]
- [die] verbale Erklärung eines Problems führt nicht zum Verstehen [ . . . ]

<sup>112</sup> Francisco J. VARELA stellt in seinem Überblick der kognitionswissenschaftlichen Forschung zusammenfassend dar: „Die wesentliche Behauptung der Kognitivisten ist, daß intelligentes Verhalten die Fähigkeit voraussetzt, die Welt als in bestimmter Weise seiend zu repräsentieren oder abzubilden“ [Varela 1990, S. 39]. Darüber hinaus behaupten die Kognitivisten, „daß wir Intelligenz und Intentionalität nur erklären können, wenn wir annehmen, Kognition sei Handeln auf der Grundlage von Repräsentationen, die physikalisch in Form eines symbolischen Kodes im Gehirn oder in einer Maschine verwirklicht sind“ [a. a. O.].

<sup>113</sup> Die Reihenfolge der Vertreter ist chronologisch.

VYGOTSKY kannte PIAGET's Arbeiten und kommentierte diese. Umgekehrt lernte PIAGET die Arbeiten von VYGOTSKY (weitere, ebenfalls gebräuchliche Schreibweisen des Namens: *Wygotski*, *Vygotski*) erst gegen Ende seines Lebens kennen. Die Arbeiten von Lev S. VYGOTSKY enden 1934 durch sein frühes Ableben abrupt. Aus politischen Gründen blieben seine Arbeiten (exemplarisch: [Vygotski 1978]) bis vor Kurzem unbekannt. Die Veröffentlichung [Gagné 1962] ist in besonderer Weise auf militärische Einsatzbereiche fokussiert – ein Aspekt, der in der deutschsprachigen Rezeption kaum Erwähnung findet.

Direkte Umsetzungsversuche der Ideen von PIAGET und GAGNÉ in Lehrstufen werden von Hilbert MEYER in [Meyer 1988, S. 160] als gescheitert charakterisiert. Die Ideen von PIAGET werden im Kontext des Konstruktivismus unter anderer Blickrichtung wieder neu diskutiert (vgl. die ausführliche Darstellung: „Piaget und Lernen im naturwissenschaftlichen Unterricht“ [Häußler u. a. 1998, S. 183–190]).

<sup>114</sup> „Der radikale Konstruktivismus ist eine Wissenstheorie“ [von Glasersfeld 1995, S. 165]

<sup>115</sup> wegbar, gangbar – von „via“ (lat. Weg)

<sup>116</sup> Dieser Wissensbegriff unterscheidet sich deutlich von der traditionellen Auffassung, die davon ausgeht, dass Wissen außerhalb des Menschen existiert (und damit dann auch transportiert werden kann).

- [es] ist daher von wesentlicher Bedeutung, daß der Lehrer über ein adäquates Modell des begrifflichen Netzwerkes verfügt, innerhalb dessen der Schüler assimiliert<sup>117</sup> [ . . . ]
- Lernen [ist] das Produkt von Selbstorganisation.  
[von Glasersfeld 1989, S. 190f]

Die Kognitionstheorie PIAGETS wird von Konstruktivistinnen als Beleg für ihre Sicht auf Lernprozesse herangezogen. Danach stellt sich die Frage „wie der rationale Geist Erfahrung organisiert“. PIAGETS „Schematheorie [ . . . ] ist der Versuch, einen Teil dieser Frage zu beantworten. [ . . . ] Wissen wird niemals [nur] passiv erworben, denn Neues kann nur durch Assimilation an eine kognitive Struktur bewältigt werden, die das erfahrende Subjekt bereits besitzt. Das Subjekt kann in der Tat keine Erfahrung als neuartig wahrnehmen, bevor diese nicht mit Bezug auf ein erwartetes Ergebnis eine Perturbation<sup>118</sup> erzeugt. Erst an diesem Punkt kann die Erfahrung zu einer Akkomodation und somit zu einer neuartigen begrifflichen Struktur führen, die relatives Gleichgewicht wieder herstellt. Es ist notwendig [ . . . ] zu unterstreichen, daß die häufigste Quelle von Perturbationen für das sich entwickelnde kognitive Subjekt die Interaktion mit anderen ist“ [von Glasersfeld 1989, S. 191f].

Darin liegt in der Tat der Grund, warum konstruktivistische Lehrer der naturwissenschaftlichen Fächer und der Mathematik Gruppenlernen gefördert und befürwortet haben, eine Praxis, die zwei oder drei Schüler gemeinsam Lösungswege für eine bestimmtes Problem diskutieren läßt, ohne daß der Lehrer eingreift. [ . . . ] Nicht länger wäre es dann möglich, an der Idee festzuhalten, daß es für eine bestimmte Aufgabe nur einen einzigen Lösungsweg gibt. [ . . . ] Somit kann daher ein Schüler durchaus eine vernünftige Lösung anbieten, die dem Lehrer unverständlich bleibt. [ . . . ] Daher würden konstruktivistische Lehrer sich bemühen zu erkunden, wie ihre Schüler ein Problem sehen und warum ihnen der eingeschlagene Lösungsweg aussichtsreich erschien. Dies wiederum ermöglicht es, ein [ . . . ] Modell des begrifflichen Netzwerkes der Schüler aufzubauen und die unterrichtlichen Tätigkeiten so [zu gestalten . . . ], daß sich daraus Gelegenheiten für Akkomodationen ergeben, die in der Reichweite der Schüler liegen.

[von Glasersfeld 1989, S. 192f]

## Stellenwert des Konstruktivismus im Vergleich

Die Wende vom Instrukionalismus (dessen Basis sowohl behavioristische als auch kognitivistische Elemente umfasst) zum Konstruktivismus wird von Rolf SCHULMEISTER als Paradigmenwechsel bezeichnet (vgl. [Schulmeister 2002, S. 165, 169]). Daher wird hier auf eine detaillierte Darstellung von Ausprägungen verzichtet, die das instruktionale Design um ausgewählte „konstruktivistische Elemente“ anreichern, um der Kritik von Konstruktivistinnen Rechnung zu tragen.<sup>119</sup>

Es geht dem Konstruktivismus nicht nur darum, dem Lernenden »etwas mehr« Verantwortlichkeit zuzuschreiben, sondern den fundamentalen Wechsel von der Instruktion zum Lernen zu vollziehen. Der Lehrer ist nicht mehr der Steuermann und der Polizist des Unterrichts, sondern Ressource und Facilitator<sup>120</sup> für den Lernprozeß. Der Lernende ist seine eigene Kontrolle. Die Abgabe der Verantwortung an den Lernenden ist kein bloßer motivationspsychologischer Trick des allmächtigen Lehrers, um den Lernenden besser manipulieren zu können. Es geht um den Lernenden selbst, als autopoietischem<sup>121</sup> Wesen, das fähig ist, selbsttätig zu lernen.

[Schulmeister 2002, S. 169]

<sup>117</sup> „Durch die Assimilation versucht das Individuum, die außenweltlichen Ereignisse, die neuen »Erfahrungen«, seinen bereits vorhandenen kognitiven Strukturen, seinen verfügbaren Schemata, anzugleichen. Gelingt dies nicht mehr, müssen die vorhandenen Schemata modifiziert, oder es muß ein völlig neues Schema entwickelt werden. Diesen Prozeß nennt Piaget Akkomodation“ [Häußler u. a. 1998, S. 184]. Der gesamte Prozess wird von PIAGET als Äquilibration bezeichnet.

<sup>118</sup> Nach [Maturana und Varela 1992, S. 108]: „alle Interaktionen, die Zustandsveränderungen auslösen.“

<sup>119</sup> Rolf SCHULMEISTER stellt die Auseinandersetzungen zwischen Instrukionalisten und Konstruktivistinnen ausführlich dar (Kapitelüberschrift: „Die verzweifelte Gegenwehr des Instrukionalismus“) [Schulmeister 2002, S. 166].

<sup>120</sup> Facilitator (engl.) := die Unterstützerin | der Unterstützer | die Vermittlerin | der Vermittler

<sup>121</sup> Autopoiese (lt. [Maturana 1994]): „(nach dem griechischen autos und poiein für Selbstgestaltung: autopoiesis), womit ich ausdrücken wollte, wie sich Systeme als Produkte ihrer eigenen Operationen realisieren.“ – vgl. <http://beat.doebe.li/bibliothek/w00025.html>

Es interessiert weniger das Problem, wie Wissen vermittelt wird, als vielmehr die Frage, wie Wissen konstruiert wird und in welcher Verbindung Wissen und Handeln stehen. Jeder Mensch gestaltet das Wissen so, dass es in seinen eigenen Bezugsrahmen eingepasst werden kann. In einer entsprechend aktiven Position befindet sich die Lernende, während der Lehrenden die Aufgabe zukommt, Problemsituationen und „Werkzeuge“ zur Problembearbeitung zur Verfügung zu stellen und bei Bedarf auf Bedürfnisse der Lernenden zu reagieren. In der konstruktivistischen Unterrichtsphilosophie wird Lernen als aktiv-konstruktiver Prozess betrachtet, der situativ gebunden erfolgt.

Zusammenfassend kann festgestellt werden, dass im Forschungsbereich (bis auf Randbereiche) der Behaviorismus als überwunden betrachtet werden kann. In vielen Bereichen der pädagogisch relevanten Verhaltenssteuerung spielt er nach wie vor eine wichtige Rolle: z. B. Lob und Tadel.<sup>122</sup> Weder behavioristisch noch kognitivistisch geprägte Lehr-/Lernsysteme<sup>123</sup> haben sich (lt. [Schulmeister 2002, S. 127]) in der schulischen Praxis breiter durchgesetzt; andererseits ist festzustellen, dass die vom Paradigmenwechsel zum Konstruktivismus ausgehenden grundlegenden Überlegungen bisher im schulischen Umfeld ebenfalls (noch) nicht breiter umgesetzt werden.

Im folgenden Abschnitt werden Elemente einer vom Lernenden ausgehenden konzeptionellen Grundlage für unterrichtliches, schulisches Lernen dargestellt. Damit werden ausgewählte, pragmatische Ansätze präsentiert, die Wege zu einem stärker subjektorientierten Verständnis für die Praxis schulisch organisierter Lernprozesse deutlich machen sollen.

## 3.2 Unterrichtskonzepte – Prinzipien methodischen Handelns



vgl. [Meyer 1988, S, 209]

Abbildung 3.4: Methoden/Konzepte des Unterrichts<sup>124</sup>

Konzepte des Unterrichts (Handlungsvorschläge) stellen Orientierungen dar, die eine wichtige Rolle innerhalb der Bereichs- und der Fachdidaktiken besitzen. Sie unterscheiden sich von den allgemeindidaktischen

<sup>122</sup> Bei der Arbeit mit vielen Informatikprodukten können Elemente des „negativen Feedbacks“ nachgewiesen werden. Dies entspricht im behavioristischen (Reiz-Reaktions-Schema) dem „Bestrafen“. In der Argumentation der Behavioristen wird dadurch das „Fehlverhalten“ zwar kurzfristig vermieden, langfristig ist die Strategie erfolgreicher, bei „Fehlern“ keine Reaktion zu zeigen. (Der Begriff Fehler wird dabei im Sinne von „unerwünschtem Verhalten“ gebraucht.)

<sup>123</sup> mit Unterstützung durch Informatiksysteme

<sup>124</sup> Die hervorgehobenen Methoden werden im Folgenden näher beleuchtet.

Darüber hinaus wird hier zusammenfassend auf den genetischen Unterricht eingegangen: das genetische Prinzip des Unterrichts wird von Martin WAGENSCHNEIN durch die Merkmale genetisch, sokratisch und exemplarisch beschrieben (vgl. [Wagenschein 1982]). Er hebt folgende Merkmale hervor:

- die pädagogische Dimension des Unterrichts (genetisch),
- nicht belehren oder unterweisen, sondern die Schülerin beim eigenen Zurechtfinden und Urteilen unterstützen (sokratisch),
- die Schülerin soll selbst entdecken; die Entdeckungen sollen paradigmatischen Charakter haben, so dass eine Stoffauswahl notwendig und möglich ist (exemplarisch).

„Gerade der exemplarische Charakter des genetischen Lehrens scheint ein Hinweis dafür zu sein, dass sich sehr gute Verknüpfungen zu den fundamentalen Ideen herstellen lassen, die ebenfalls paradigmatischen Charakter haben sollen. [...] Ein Gegensatz zwischen der Orientierung an fundamentalen Ideen und dem Spiralprinzip einerseits sowie dem genetischen bzw. entdeckenlassenden Lehren andererseits kann unter diesem akzentuierten Blickwinkel nicht gesehen werden“ [Vohns 2000, S. 40].

Theorien, da sie keinen Anspruch auf allgemeine Gültigkeit erheben und häufig nicht in einen größeren wissenschaftstheoretischen Kontext eingebunden sind. Deshalb kann auf dieser Ebene das Problem konfligierender Normen nicht gelöst werden. Unterrichtskonzepte können als konstruktive Reaktion auf aktuelle fachdidaktische und/oder unterrichtsmethodische Probleme betrachtet werden. Sie formulieren Prinzipien methodischen Handelns und stellen einen Rahmen für konkrete methodische Entscheidungen zur Verfügung.

### **Problemorientierung im Unterricht – Potenziale**

In Übereinstimmung mit Hilbert MEYER können *Probleme* als „nicht routinemäßig lösbare Aufgaben“ charakterisiert werden [Meyer 1988, S. 211]. Ein auf Probleme bezogenes Stufenschema entwickelte Heinrich ROTH [Roth 1976]. Es weist drei Arten von Problemen aus:

- (I) Probleme, die sich der Lernenden in realen Problemsituationen stellen und die sie löst, ohne dabei daran zu denken, dass sie zugleich etwas lernt,
- (II) Probleme, die die Lernende selbsttätig und selbstständig, aber mit bewusster Lernabsicht zu lösen versucht, und
- (III) Probleme, vor die die Lehrerin ihre Schülerinnen zum Zwecke der Belehrung stellt  
vgl. [Meyer 1988, S. 183]

Diese Überlegungen stellen einen geeigneten Ausgangspunkt für eine schülerorientierte Interpretation der Kategorisierung von Problemklassen dar, verweisen sie doch darauf, dass (I) und (II) für unterrichtliche Situationen ausgezeichnete Voraussetzungen bieten. Indem ROTH die genannten Problemarten analysiert, um sie sodann in einem Phasenschema so miteinander zu verschränken/parallelisieren, dass für die Schülerinnen der Eindruck entstehen soll, dass ein Problem der Kategorie (I) und/oder (II) vorliegt, aber „eigentlich“ ein Problem der Kategorie (III) bearbeitet werden soll. Hilbert MEYER charakterisiert diese Methode, in dem er ausführt: Der Lehrer „wirft die Mütze[n] der Kinder [...] über den Bach und erwartet, dass sie sie auf der Stelle wieder holen wollen ...“ [Meyer 1988, S. 184]. Die unterrichtliche Umsetzung soll nach ROTH in sechs Stufen erarbeitet werden (vgl. [Hilger 2002, S. 5f], [Meyer 1988, S. 184–186]).

Aktuellere Überlegungen gehen nicht ganz so offen mit dieser Klassifizierung um, wie die Definition von MANDL und REINMANN-ROTHMEIER deutlich macht:

- Unter Problemorientierung verstehen wir, daß Lehrende Probleme in den Mittelpunkt ihres Unterrichts stellen, die
- entweder authentisch sind oder Bezug zu authentischen Situationen oder Ereignissen haben,
  - für die Lernenden relevant sind,
  - aktuell sind und allgemeine oder persönliche Brisanz besitzen,
  - neugierig und betroffen machen.

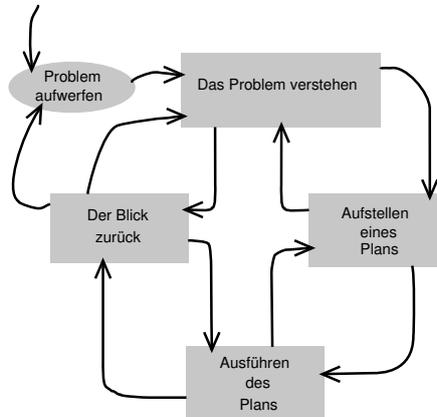
[Mandl und Reinmann-Rothmeier 1995, S. 27]

Unzweifelhaft stellen Probleme der von ROTH vorgestellten Arten (I) und (II) eine Möglichkeit da, Unterricht am selbstständigen Denken der Schülerinnen zu orientieren und damit Probleme nicht nur zum Ausgangspunkt des Unterrichts zu machen, sondern sie in den Mittelpunkt des Unterrichts zu stellen.

Probleme ermutigen dazu, persönliche Erfahrungen zu nutzen und konstruktive Leistungen zu erbringen. Sie konstituieren sich nur in einem spezifischen Kontext und verdeutlichen damit, dass Interpretationen, Standpunkte und Lösungen immer situative Merkmale haben. Probleme eignen sich schließlich auch besonders gut dazu, in sozialen Austauschprozessen gemeinsam Lösungen zu erarbeiten und dabei unterschiedliche Perspektiven zu berücksichtigen. Damit ist Problemorientierung im Unterricht das Gestaltungsprinzip, das einer [...] konstruktivistischen Auffassung vom Lernen in hohem Maße Rechnung trägt.

[Mandl und Reinmann-Rothmeier 1995, S. 28]

Im Zusammenhang mit Überlegungen zur Beschreibung von Heuristiken zur Lösung mathematischer Probleme im schulischen Kontext hat Georg PÓLYA einen Rahmen (vgl. [PÓLYA 1945], [PÓLYA 1967]) entwickelt.<sup>125</sup>



nach [PÓLYA 1967]

Abbildung 3.5: Schema zum Problemlöseprozess<sup>126</sup>

Schemata zur Problemlösung dienen der Strukturierung unübersichtlicher nicht unmittelbar zugänglicher Strategien zur Bearbeitung [komplexer] Problemstellungen. Sie legen einen Ablauf (in Form einer Reihung/Sequenz) nahe und sollen die Entscheidungsfindung unterstützen. Das Schema aus Abbildung 3.5 bietet auch für den Kontext informatischer Problemstellungen eine tragfähige Lösungsstrategie, wie sie in [Balzert 1976, Klappentext (innen)] benutzt wird, um ein „Schema zur Problemlösung“ zur Bearbeitung von Problemen auszuweisen. Ein vereinfachtes Schema, das den Prozess der imperativen/prozeduralen Problemlösung (Top-Down-Analyse) unterstützt, findet sich im Anhang G.1.<sup>127</sup> Die Verzahnung mit dem Konzept der Problemorientierung liegt nahe, wenn auch zu bedenken ist, dass solche Schemata keine Schülerorientierung ersetzt. Sie stellen vielmehr Werkzeuge zur Verfügung, um bei der Bearbeitung von Problemen bestimmte Schritte voneinander abzugrenzen und die Kommunikation über Problemlösungen und Problemlösestrategien zu erleichtern.

## Die Projektmethode – projektorientierter Unterricht

Kaum ein Begriff der pädagogischen Theorie und Praxis weist derartig viele Facetten auf, wie der Projektbegriff (vgl. [Wolters 1994]). Dies ist der Tatsache geschuldet, dass dieser Begriff sowohl im Zusammenhang mit der (allgemein-)pädagogischen Praxis, darüber hinaus in der berufsbezogenen Ausbildung eine lange Tradition hat. Hier ist nicht beabsichtigt, die aktuell geführte Diskussion um die (Nicht-)Berücksichtigung der historischen Dimension des Projektgedankens in der Pädagogik aufzuarbeiten. Durch eine kurze Darstellung der Geschichte kann deutlich werden, wie der Projektgedanke – von der beruflichen Bildung

<sup>125</sup> Im Abschnitt 4.3 wird das Konzept des Problemlösens auf den Informatikunterricht bezogen dargestellt.

<sup>126</sup> Die Abbildung wurde aus dem Amerikanischen <http://jwilson.coe.uga.edu/emt725/Intro/image3.gif> ins Deutsche übertragen. Zur Erläuterung der Skizze führt der Autor James W. WILSON aus: „Any of the arrows could describe student activity (thought) in the process of solving mathematics problems. Clearly, genuine problem experience [...] can not be captured by the outer, one-directional arrows alone. It is not a theoretical model. Rather, it is a framework for discussing various pedagogical, curricular, instructional, and learning issues involved with the goals of mathematical problem solving in our schools“ <http://jwilson.coe.uga.edu/emt725/Intro/Intro.html> vgl. auch [Wilson u. a. 1993].

<sup>127</sup> Zur Unterstützung des Problemlöseprozesses bei der objektorientierten Modellierung wird ebenfalls ein Schema vorgeschlagen (vgl. Anhang G.2).

ausgehend – in der allgemeinen Pädagogik aufgenommen wurde. Darüber hinaus kommt dem Projektgedanken im Kontext der informatischen Bildung, aber auch der Informatik in der beruflichen Praxis, eine wichtige Rolle zu (vgl. 4.1).

Die Geschichte der Projektmethode kann in fünf Phasen gegliedert werden<sup>128</sup> (siehe Tabelle 3.2).

<b>1590–1765</b>	die Anfänge der Projektarbeit an den Schulen für Architektur in Italien und Frankreich
<b>1765–1880</b>	das Projekt als reguläre Unterrichtsmethode an den kontinentaleuropäischen Bauakademien und Hochschulen für Ingenieurwissenschaft und die Übertragung des Projektgedankens nach Amerika
<b>1880–1915</b>	die Durchführung von Projektarbeit im Werk- und Arbeitsunterricht der amerikanischen High und Elementary School
<b>1915–1965</b>	die Neudefinition der Projektmethode durch KILPATRICK und ihre Rückübertragung von Amerika nach Europa
<b>1965–heute</b>	die Wiederentdeckung der Projektidee in Westeuropa und die dritte Welle ihrer internationalen Verbreitung.

vgl. [Knoll 1999]

Tabelle 3.2: Geschichtliche Phasen der Projektmethode

Michael KNOLL hat festgestellt, dass „aus einem Begriff des Bau- und Ingenieurwesens ein Begriff der Pädagogik und der Didaktik wurde. [...] Auf Schule und Hochschule wurden die berufsspezifischen Methoden übertragen [... z. B.] das Experiment der Naturwissenschaftler, die Fallstudie der Juristen, das Planspiel der Offiziere: angepaßt an die Bedingungen des institutionalisierten Lernens gehören sie heute zu den Standardmethoden des Unterrichts“ [Knoll 1999]. Dies erfolgte nicht für das Projekt im allgemein bildenden Schulwesen. Für die ersten drei der o. g. Phasen (bis 1915) wird als Ziel des Projektunterrichts ausgewiesen: Professionalisierung in der Berufsausbildung. Der Projektunterricht wurde „in die Akademien und Schulen übernommen, damit die Studenten und Schüler bereits während ihrer Ausbildung Gelegenheit hatten, die [...] erworbenen Kenntnisse und Fertigkeiten selbständig auf lebens- und wirklichkeitsnahe Situationen anzuwenden“ [Knoll 1999].<sup>129</sup>

Ab ca. 1900 finden sich in den USA vermehrt Belege für die Verwendung des Begriffs Projekt im praktisch orientierten Unterrichtszusammenhang (vgl. [Pütt 1982, S. 13ff]). Die Ideen des Projektunterrichts wurden um 1900 von Charles R. RICHARDS in die allgemeine Bildung der Elementary School übertragen. Dazu änderte RICHARDS die Schwerpunktsetzung und rückt das Projekt in den Mittelpunkt des Werkunterrichts. Nicht „Instruktion vor Konstruktion“, sondern „Instruktion durch Konstruktion“ lautet das Motto.

Eine breite, allgemeinpädagogisch orientierte, theoretische Begriffsbestimmung wird 1918 von den amerikanischen Reformpädagogen John DEWEY und William Heard KILPATRICK [Kilpatrick 1918] vorgenommen. DEWEY vertrat die Auffassung, dass Sinn und Wert entstehen, wenn Probleme gelöst werden. Die Problemlösungen werden nach dem Muster der Naturwissenschaften<sup>130</sup> erarbeitet (vgl. [Frey 1998, S. 62]). KILPATRICK hingegen fasst den pädagogischen Projektbegriff [sehr] weit und legt eine Typologie vor, in der er Produktions-, Konsumtions-, Problem- und Lernprojekte unterscheidet (vgl. [Kilpatrick 1918, S. 530]). „Damit hatte [...] KILPATRICK], unter neuem Namen und unter neuem Gesichtspunkt, die ganze Palette der herkömmlichen Methoden in seinem Konzept vereint – vom Lehrgang und Training über das Plan- und Rollenspiel bis zum Experiment und Praktikum“ [Knoll 2000]. In der Ausgestaltung findet

<sup>128</sup> vgl. [Knoll 1997] deutsche Fassung lt. [Knoll 1999]

<sup>129</sup> Die ersten aus den Vereinigten Staaten dokumentierten Beispiele für Projektunterricht (nach WOODWARD ~1880) bestehen in der Umsetzung der klaren Abfolge der Abschnitte Lehrgang, Übung und Projekt. Die Struktur wird von KNOLL als „Lineares Modell“ charakterisiert, da alle der drei aufeinander folgenden Phasen der Reihe nach durchlaufen werden müssen. Innerhalb des Projekts gibt es wiederum eine klare zeitliche Reihung, die als Projektphasen Planen, Entwerfen und Verwirklichen ausgewiesen werden (vgl. [Knoll 2000]).

<sup>130</sup> Methodologie: Hypothesen bilden, Lösungen entwerfen, ausprobieren

sich keine Abfolge von durchzuführenden Schritten, es wird nicht angegeben, ob ein Ergebnis produziert werden soll/muss. Der Projektbegriff wird so vom Methodenbegriff gelöst und zum Prinzip des Unterrichts erklärt. Dies führt (bis heute) dazu, dass häufig unklar bleibt, was gemeint ist, wenn der Begriff Projekt benutzt wird.

Darüber hinaus trugen sowohl in Deutschland, wie auch in den Vereinigten Staaten von Amerika eine Anzahl dokumentierter Beispiele dazu bei, die Projektidee immer wieder neu zu beleben und als fruchtbares Moment einer gelungenen pädagogischen Praxis auszuweisen/darzustellen. Heinz PÜTT stellt 1982 fest: „Der nach wie vor anhaltende Drang nach der Durchführung von Projektunterricht und sogenannten Projektwochen ist wohl damit zu erklären, dass Schüler und Lehrer gleichermaßen in dieser dem schülerorientierten Unterricht adäquatesten Form einen Weg aus der Routine suchen und damit Möglichkeiten zur Öffnung der Schule ergreifen“ [Pütt 1982, S. 63]. Dem Projektunterricht ordnet PÜTT bestimmte Merkmale zu (siehe Tabelle 3.3). Solche Merkmalslisten zeigen, dass es offenbar nicht einfach ist, anzugeben was unter einem Projekt zu verstehen ist (im Sinne einer Definition). Eine andere Möglichkeit besteht darin, ein Planungs- oder Verlaufsschema anzugeben. Dadurch soll verdeutlicht werden, wie ein Projekt geplant und durchgeführt wird/werden kann (vgl. [Pütt 1982, S. 63-98 (mit Beispielen)]).

	problemhaltige Aufgabe	Motivation	Zielorientierung	planvolles, selbstständiges und selbstbestimmtes Lernen	Verbindung von schulischem und außerschulischem Tun	Hingabe und ernstes Engagement	individuelles und kooperatives Handeln	Verbindung von Theorie und Praxis	Ausdauer	Abbau von Lehrerdominanz	typische Verlaufsstruktur	fachübergreifender Charakter	Methodenvielfalt	Abschluss und Aufgabenbeurteilung	gesellschaftliche Relevanz
Projektunterricht	✓ <sup>131</sup>	✓	✓	✓++	✓	✓	✓	✓++	✓	✓	✓	✓++	✓	✓	✓

vgl. [Pütt 1982, S. 101]

Tabelle 3.3: Merkmale des Projektunterrichts

Karl FREY sieht sich in der 8. Auflage des Buches zur Projektmethode veranlasst, Merkmale [Frey 1998, S. 16f] der Projektmethode anzugeben. Darüber hinaus erläutert er seine Zugangsweise zur Projektmethode [Frey 1998, S. 61-70] und stellt ein Grundmuster des Projektunterrichts [Frey 1998, S. 77] vor. Ablaufbeispiele [Frey 1998, S. 18-20] und Komponenten der Projektmethode [Frey 1998, S. 86ff] vervollständigen das Bild. Durch die Darstellung wird deutlich: in der Präsentation von anregenden Ideen, zu berücksichtigenden Rahmenbedingungen und Seiteneffekten, gelungenen und misslungenen Projekten entsteht ein Eindruck der ungebrochenen pädagogischen Wirksamkeit der Projektmethode.

Anfang der siebziger Jahre des letzten Jahrhunderts stellt Bernhard SUIN DE BOUTEMARD ein Projektmodell vor, das einen ähnlich umfassenden Anspruch, wie das Konzept von KILPATRICK einlösen soll. Als Grundlage wird ebenfalls auf die Ideen von John DEWEY abgehoben. Das vorgestellte Projektmodell wird als höchste und umfassendste der drei Unterrichtsformen: Informations-, Problem- und Projektunterricht eingestuft.<sup>132</sup>

<sup>131</sup> Legende:

✓ mögliches Merkmal des Projektunterrichts;

✓ ++ Muss-Kriterium, damit von einem Projekt gesprochen werden kann – ist eines der Muss-Kriterien nicht erfüllt, sollte (lt. PÜTT) von projektorientiertem Unterricht gesprochen werden.

Das wichtigste Merkmal ist unzweifelhaft die „Autonomie der Lernenden im Lernprozeß“, wie [Pütt 1982, S. 101] herausstellt.

<sup>132</sup> In der Konsequenz mit KILPATRICKS Anspruch zu vergleichen.

Ein praxistaugliches, für die Umsetzung in der allgemein bildenden Schule praktikables Modell wird von Herbert GUDJONS (vgl. Tabelle 3.4) vorgestellt.

Projektschritt	Merkmale
1 Eine für den Erwerb von Erfahrungen geeignete, problemhaltige Sachlage auswählen	Situationsbezug Orientierung an den Interessen der Beteiligten Gesellschaftliche Praxisrelevanz
2 Gemeinsam einen Plan zur Problemlösung entwickeln	Zielgerichtete Projektplanung Selbstorganisation und Selbstverantwortung
3 Sich mit dem Problem handlungsorientiert auseinandersetzen	Einbeziehen vieler Sinne Soziales Lernen
4 Die erarbeitete Problemlösung an der Wirklichkeit überprüfen	Produktorientierung Interdisziplinarität Grenzen des Projektunterrichts

vgl. [Gudjons 2001, S.81–94]

Tabelle 3.4: Schritte und Merkmale eines Projektes

Ungeachtet der Diskussion des Projektbegriffs im allgemeinpädagogischen Kontext muss konstatiert werden, dass Projekte seit vielen Jahren erfolgreiche Bestandteile der beruflichen Ausbildung von Informatikerinnen darstellen. Die Komplexität und der häufig fächerübergreifend Charakter informatischer Fragestellungen und die Art der Bearbeitung im Kontext der Fachwissenschaft kann als Ursache für den Erfolg dieser im schulischen Bereich nach wie vor nicht zum Regelunterricht gehörenden „Methode“ angesehen werden.<sup>133</sup> In Informatikprojekten – ob schulisch oder „hoch“schulisch umgesetzt – findet eine [besondere] professionelle Arbeitsweise ihren Ausdruck.

Interessant ist – auf dem Hintergrund der pädagogischen Diskussion – die Reflexion der informatikbezogenen Auseinandersetzungen um Vorgehens- und Prozessmodelle – allgemein um die adäquate Abbildung eines Prozesses in Phasen. Die Diskussionen verlaufen in ähnlicher Weise: häufig wird zu Beginn der Entwicklung offenbar davon ausgegangen, dass sich derartig komplexe Planungsprozesse durch ein Phasenmodell (mehr oder weniger restriktiv) handhaben lassen. In der tieferen Auseinandersetzung (und durch das „Scheitern“ von Projekten bedingt) werden Alternativen geprüft: die Phasenabfolge wird nicht mehr als Muss-Kriterium angesehen, ein Projekt kann nach jeder Phase ein Ergebnis liefern oder auch beendet werden, es werden spiralförmige und/oder evolutionäre Modelle auf ihre Praxistauglichkeit geprüft. Die Erkenntnis, dass es Nutznießer, Betroffene, etc. gibt, also Menschen, die für das Gelingen oder Mislingen des Projekts mittelbar oder unmittelbar bedeutend sind, und damit berücksichtigt werden müssen, hat sowohl in der pädagogischen Diskussion, wie auch in der Softwaretechnik über die Zeit zu Konsequenzen geführt. Als Belege können – im allgemein-pädagogischen Bereich [Frey 1998, S. 86ff] mit pragmatisch orientierten Tipps zur Projektdurchführung und Beispielen, was warum gescheitert ist – in der Informatik die „partizipative Softwareentwicklung“ ([Floyd u. a. 1987], [Reisin 1992]), die Möglichkeiten zur Strukturierung des Prozesses der Softwareerstellung ([DeMarco 1978], [DeMarco 2002]), die Probleme beim Bau des Betriebssystems für die IBM /360 (mit 1.000 Personen vgl. [Brooks u. a. 1966], [Brooks 2002]), aber auch ökonomische Fragen im Kontext der Softwareentwicklung ([Boehm 1984], [Boehm 2002]) herangezogen werden. Für Informatikerinnen ist die Granularität der Begriffe Problem und Projekt relativ

<sup>133</sup> Die Ergebnisse der Arbeiten von Michael KNOLL legen nahe, dass dem Projektunterricht in der allgemeinpädagogischen Diskussion von einigen Autoren „zuviel zugemutet“ wird, d.h. durch die Anwendung/Änderung einer Methode soll die Schule als Ganzes verändert werden. Dieser Anspruch kann nicht eingelöst werden, darum kann ein mit derartigen Hoffnungen verbundenes Konzept nicht adäquat umgesetzt werden.

klar: Ein Problem ist eher Teil eines Projekts.<sup>134</sup>

Für den Informatikunterricht werden nach Empfehlungen der Gesellschaft für Informatik (GI) (vgl. Abschnitt 4.1, ab 1976) in den Lehrplänen und Richtlinien der Bundesländer Projektphasen curricular verankert. Dabei sollte berücksichtigt werden, dass durch verschiedene, unscharfe Begrifflichkeiten die Trennschärfe nicht garantiert werden kann. Dies äußert sich darin, dass der jeweils zur Anwendung gebrachte, intendierte Projektbegriff nicht expliziert wird. Eher wird die Form beschrieben, in der ein „Projekt“ stattfindet.<sup>135</sup>

## Handlungsorientiertes Lernen – Handlungsorientierter Unterricht

Häufig wird mit Handlungsorientierung der Begriff Produktorientierung assoziiert. Dabei wird eine Vorstellung von „Handeln“ unterstellt, die [Handlungs-]Produkte als [konkrete] Ergebnisse des Lernprozesses begreift. In der allgemeinpädagogischen Literatur finden sich Hinweise und Definitionen, die diese Erklärung der Handlungsorientierung zur Grundlage machen.<sup>136</sup> Diese Einschränkung ist nicht notwendig, wie die Definition von Herbert GUDJONS zeigt: „Handlungsorientierter Unterricht ist [...] der [...] Versuch, tätige Aneignung von Kultur in Form von pädagogisch organisierten Handlungsprozessen zu unterstützen. Über die ikonische<sup>137</sup> Aneignungsweise hinaus bietet [...] der handlungsorientierte Unterricht] die Möglichkeit, handelnd Denkstrukturen aufzubauen und den Zugang zur Welt nicht über ihre Abbilder, sondern durch vielfältige sinnliche Erfahrungen zu schaffen. Kompensatorisch zur tendenziellen »Entwicklung der Wirklichkeit« dient er dem Aufbau einer umfassenden Handlungskompetenz [...] Er bezieht sich auf Handeln als tätigen Umgang mit Gegenständen, Handeln in sozialen Rollen und Handeln auf symbolisch-geistiger Ebene“ [Gudjons 1986, S. 49], [Gudjons 2001, S. 67].

Vorläufer der Überlegungen zu modernen handlungsorientierten Prinzipien des Lernens und Unterrichtens können in dem geflügelten Wort des Pädagogen Johann Heinrich PESTALOZZI vom „Lernen mit Kopf, Herz und Hand“, in den Überlegungen der Reformpädagogin Maria MONTESSORI [Montessori 1926]<sup>138</sup> und nicht zuletzt in der Arbeitsschulbewegung aufgezeigt werden (siehe Fußnote 93, S. 32). Die kognitionspsychologischen Überlegungen von Jerome S. BRUNER (vgl. [Bruner 1975]) werden zur Begründung der Handlungsorientierung herangezogen. Auf Grundlage<sup>139</sup> der Überlegungen BRUNERS werden den Schülerinnen durch die Lehrerin Selbstständigkeit und eigene aktive Erfahrungsmöglichkeiten zugestanden. Lernen ist kein linearer, additiver Prozess in kleinen Schritten, sondern ein vernetzter Vorgang, der das gesamte Denk- und Handlungssystem betrifft.<sup>140</sup> Ziel des handlungsorientierten Lernens ist es, die Schülerinnen mit umfassender Handlungskompetenz auszustatten, indem aufbauend auf den Interessen der Beteiligten<sup>141</sup> selbstverantwortlich der Umgang mit Gegenständen des wirklichen Lebens oder das Handeln in sozialen Rollen geplant, ganzheitlich, d. h. fächerübergreifend und mit allen Sinnen zielgerichtet und gemeinsam<sup>142</sup>, ausgeführt wird und die Produkte der Handlungen sowie die ablaufenden Prozesse<sup>143</sup>

<sup>134</sup> Da die epochalen Schlüsselfragen (-probleme) [Klafki 1985b] der Gesellschaft nicht durch informatische Problemlösungen gelöst werden (vgl. [Weizenbaum 1977]), ist diese Teilmengebeziehung m. E. realistisch.

<sup>135</sup>

- Projektarbeit in der Informatik in Unternehmen – [Bothe 1998]
- Projektgruppen (PG) am Fachbereich Informatik an der Universität Dortmund – <http://ls4-www.cs.uni-dortmund.de/PGB/alles/node13.html>
- Projekte im Informatikunterricht – [Lehmann 1985], [Koerber 1986]
- Unterrichtsprojekte allgemein, Projektwochen – [Pütt 1982], [Gudjons 2001]

<sup>136</sup> Bei Hilbert MEYER sind die Ergebnisse des handlungsorientierten Unterrichts die „vereinbarten Handlungsprodukte“ [Meyer 1987, S. 402].

<sup>137</sup> vgl. Exkurs: Fundamentale Ideen, S. 38

<sup>138</sup> Maria MONTESSORI sah in der Erziehung des Kindes zu einem selbstbestimmten, verantwortlichen Mitglied der Gesellschaft den einzigen Weg, das gestörte Gleichgewicht zwischen Mensch und Umgebung wiederherzustellen und leitete daraus die Ideen eines spielerischen, ganzheitlichen, enaktiven und schülerorientierten Unterrichts ab.

<sup>139</sup> Diese Überlegungen werden auch mit dem Terminus „Entdeckendes Lernen“ bezeichnet.

<sup>140</sup> Die Überlegungen BRUNERS weisen damit über den Kognitivismus hinaus auf den Konstruktivismus (vgl. [Schulmeister 2002, S. 71f]).

<sup>141</sup> vor allem der Schülerinnen

<sup>142</sup> d. h. in Kooperation und Interaktion mit anderen

<sup>143</sup> Kommunikation, Kooperation, Planung, Ausführung

an Hand der gemeinsam gesteckten Ziele kontrolliert werden (vgl. [Korbmacher 1992]). Handlungsorientiertes Lernen wird dadurch gefördert, dass die Schülerinnen möglichst in realen Situationen Lerngelegenheiten wahrnehmen können, die zur Exploration einladen und in denen die Schülerinnen neues Wissen selbstständig erwerben.

Handlungsorientiertes Lernen kann damit als Bündelung bewährter, didaktischer Prinzipien charakterisiert werden.<sup>144</sup> Damit wird deutlich, dass projektunterrichtliche Ansätze (z. B. in [Gudjons 1986]) als Mittelpunkt der Handlungsorientierung aufgeführt werden.

Die Schülerinnen setzen sich aktiv mit Problemen auseinander, sie sammeln selbstständig eigene Erfahrungen, sie führen Experimente durch und finden so Einsichten in komplexe Inhalte, Konzepte und Prinzipien. Das eigenständige Entdecken ist eine notwendige Bedingung für das Erwerben von Problemlösestrategien und heuristische Methoden durch die Schülerinnen.

In diesem Kontext sollte darauf hingewiesen werden, dass es einen Widerspruch zwischen dem Konzept der fundamentalen Ideen<sup>145</sup> und dem Konzept der Handlungsorientierung in dem hier vorgestellten Sinn gibt. Nicht die durch die Wissenschaft, durch die Fachdidaktik oder durch die Lehrerin vorgenommene Strukturierung des Fachgegenstandes und des Aufweises des Fundamentalen stellt für den individuellen Lernprozess die entscheidende Grundlage dar, sondern die (gemäß konstruktivistischer Auffassung) Konstruktion des Wissens beim Individuum. Fundamentale Ideen haben üblicherweise objektiven Charakter. Diese Sicht aber verbietet sich bei konstruktivistischer Betrachtungsweise. Aus diesem Widerspruch haben GALLIN und RUF das Konzept der relativistischen Pädagogik [Gallin und Ruf 1998]<sup>146</sup> entwickelt, das Kernideen<sup>147</sup> in den Mittelpunkt des Unterrichts stellt. Sie weisen deutlich auf die Bedeutung der Gewinnung der Kernideen hin, und zeigen, dass diese im Prozess eines rückschauenden verstehenden Verständnisses gewonnen werden. Damit eignen sie sich nicht „vor der Hand“ für den Lehrprozess: „Aus seiner Rückschau auf sein geordnetes und gegliedertes Wissensgebiet entwickelt der Lehrer Kernideen, die das ganze in vagen Umrissen andeuten. [...] Sie sind so knapp und prägnant, daß sie das Gedächtnis der Lernenden nicht belasten“ [Gallin und Ruf 1998, S. 75]. Um zu einer ernsthaften Berücksichtigung der Lernenden zu gelangen, muss der Blick in die andere Richtung gelenkt werden. Dabei kommt dem von GALLIN und RUF formulierten Prinzip des „Lernens auf eigenen Wegen“ als Ausdruck der Verpflichtung gegenüber konstruktivistischen Ideen eine zentrale Funktion zu. Dieses Prinzip soll zu individueller Aneignung aus der „Vorschau“-Perspektive der Schülerin anregen. Dies kann eine Möglichkeit darstellen, der „Objektivitätsfalle“ zu entkommen, bedarf allerdings noch eingehender Prüfung bzgl. realistischer Ansätze zur Umsetzung.

## Unterrichtsformen – Sozialformen des Unterrichts

Eng verbunden mit dem Unterricht sind die konkreten Organisationsformen der Arbeit der Schülerinnen. Diese werden als Sozialformen oder Unterrichtsformen bezeichnet. In der Abbildung 3.6 sind gängige Unterrichtsformen/Sozialformen von Hilbert MEYER kollagenartig zusammengestellt.

Als Hinweis für einen in sehr großen Teilen lehrgangsartig organisierten Unterricht in der Bundesrepublik kann die Aussage von Hilbert MEYER auf die Dominanz des Frontalunterrichts mit ca.  $\frac{4}{5}$  des Anteils an der gesamten Unterrichtszeit gesehen werden ( [Meyer 1987, mit Verweisen auf verschiedene empirische

<sup>144</sup> [Korbmacher 1992] führt die folgenden Prinzipien auf: Schülerorientierung, selbstständiges Lernen, soziales Lernen, ganzheitliches Lernen, exemplarisches Lernen, problemorientiertes Lernen, Arbeitsablauforientierung.

<sup>145</sup> Zum Konzept der fundamentalen Ideen für den Informatikunterricht siehe 4.2.1 – 1993, S. 63.

<sup>146</sup> Für die Ebene der Unterrichtsplanungsprozesse als „Didaktik der Kernideen“ [Gallin und Ruf 1990] bezeichnet, wie [Krainer u. a. 2002] ausführen.

<sup>147</sup> „Kernideen« [charakterisieren] vor allem begriffliche Vorstellungen von SchülerInnen [...] Sie ] enthalten als wichtiges Merkmal auch den Antrieb des Lernenden, Fragen zu formulieren, und stellt insofern eine Brücke zwischen dem Lernenden und dem Stoff dar. Mit "Kernidee" können daher nicht nur Dinge gemeint sein, die das Wesentliche einer fachlichen Idee betreffen, sondern auch Ideen, die sich am Rande eines Themas oder sogar von ihm weggeben, oder möglicherweise – zumindest in der Rückschau von ExpertInnen – sogar »falsch« sind [...] Der Kernidee-Ansatz erfordert von den Lehrenden nicht nur, dass sie die Antworten kennen, die für ein Fachgebiet wichtig sind, sondern auch die Fragen, die zu den Antworten geführt haben. Sie sollten auch Wege und Irrwege, denen diese Antworten zu verdanken sind, aus eigenen Erfahrungen kennen“ [Krainer u. a. 2002].



vgl. [Meyer 1988, S. 209]

Abbildung 3.6: Unterrichtsformen/Sozialformen des Unterrichts

Untersuchungen – S. 61ff], Details siehe Tabelle 3.5). Eine aktuellere Studie, die im Auftrage des *Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie* (BMBF) zum „Bild der Schule aus der Sicht von Schülern und Lehrern“ 1995 durchgeführt wurde<sup>148</sup> bestätigt die weiterhin dominante lehrerzentrierte Art des praktizierten Unterrichts.

Die aktuellere Studie dokumentiert differenziert die Situation in bundesdeutschen Schulen aus der Sicht von Schülerinnen und Lehrerinnen. Damit wird der Ist-Zustand aufgenommen, es werden Defizite aus der Sicht der konkreten Akteure dargestellt. Ausgewählte Ergebnisse der Befragungen werden in den folgenden Punkten zusammenfassend dokumentiert.

- Selbständig an selbstgewählten Aufgaben arbeiten bundesdeutsche Schülerinnen und Schüler nur selten [...]
- Noch seltener kommen Schülerinnen und Schüler dazu, eigene Untersuchungen im Unterricht durchzuführen [...]
- Die klassischen, eher lehrerzentrierten Unterrichtsformen dominieren nach wie vor Deutschlands Klassenzimmer, Schülerzentriertheit und entdeckendes, forschendes Lernen ist eher seltener zu finden [...]
- Gänzlich uneinheitlich schließlich ist das Meinungsbild, wenn es um die Einschätzung des lehrer gelenkten Unterrichtsgesprächs als beste Methode geht. 35% der Befragten vertritt die Meinung, daß eine Mehrheit in ihrem Kollegium dieser Aussage zustimmen würden, 26% lehnen dies ab [<sup>149</sup> ...]
- Ein gravierendes Manko in der heutigen bundesdeutschen Schule ist nach Ansicht der befragten Lehrerinnen und Lehrer, daß in der Schule zu wenig auf die Fähigkeit zur Zusammenarbeit geachtet wird [...]
- Sehr zurückhaltend fällt [...] die Antwort der Schülerinnen und Schüler auf die Frage aus, ob ihrer Meinung nach die Lehrer schwierige Sachverhalte gut erklären können; immerhin 21% geben an, daß dies nur für sehr wenige oder gar keinen Lehrer gelte [...]
- Nur selten übernehmen nach eigener Aussage die Lehrerinnen und Lehrer Vorstellungen der Schüler in die eigene Unterrichtsplanung, und der Wunsch danach ist auch nicht besonders stark ausgeprägt [...]

<sup>148</sup> Diese Studie wurde vor der Durchführung und Diskussion der Ergebnisse der internationalen Vergleichsstudien

- *Third International Mathematics and Science Study* (TIMSS) im Auftrag der *International Association for the Evaluation of Educational Achievement* (IEA) [Baumert u. a. 1998] und
- *Programme for International Student Assessment* (PISA) durch die *Organisation for Economic Co-operation and Development* (OECD) [Baumert u. a. 2001]

durchgeführt.

<sup>149</sup> Gefragt wurden die Lehrerinnen:

„Ist die folgende Aussage für Ihre Schule zutreffend? [...] Mehrheit im Kollegium meint, das lehrergelenkte Unterrichtsgespräch ist die beste Methode“ [Kanders u. a. 1997, S. 36].

- Dürfen Schüler mitbestimmen, *wie* im Unterrichts vorgegangen wird? Wiederum klaffen Wunsch und Wirklichkeit weit auseinander: Nicht einmal jeder zehnte Befragte gibt an, daß die meisten Lehrer diese Form der Schülermitwirkung zulassen, bei Gymnasiasten sind es nur 5%. [...] Offensichtlich wünschen Schüler ein Unterrichtsklima, in dem sie das Gefühl haben, auf die Vorgehensweisen bei der Vermittlung von Lerninhalten substanziell einwirken zu können.

[Kanders u. a. 1997, Zitatenkollage (S. 19-99)]

Sozialform/Unterrichtsform	Quelle	[Kanders u. a. 1997]	
	[Meyer 1987, S. 61] empirische Studien	Selbstausskunft <sup>150</sup>	
	Fremdbeobachtung	Lehrerinnen	Schülerinnen
„Die Schüler sitzen und hören zu, der Lehrer redet.“	–	35% <sup>151</sup>	48%
„Der Lehrer redet und stellt Fragen, einzelne Schüler antworten.“	–	48%	66%
„Der Lehrer und die Klasse diskutieren gemeinsam.“	–	42%	29%
Frontalunterricht	76,86 %	–	–
Gruppenunterricht	7,43 %	26%	14% <sup>152</sup>
Partnerarbeit	2,88 %	–	–
Einzelarbeit	10,24 %	41%	52% <sup>153</sup>
„Die Schüler arbeiten selbständig an selbstgewählten Aufgaben.“	–	10%	9%
„Die Schüler führen eigene Untersuchungen durch.“	–	6%	7%
„Die Schüler bearbeiten Arbeitsblätter.“	–	45%	45%
„Die Schüler gucken Filme und Videos.“	–	6%	10%
„Die Schüler arbeiten am Computer.“	–	5%	4% <sup>154</sup>

Tabelle 3.5: Praktizierte Unterrichtsformen

„Nur ein Fünftel aller Schüler ist derzeit mit den pädagogischen Fähigkeiten ihrer Lehrer zufrieden. Der Frontalunterricht mit aktivem Lehrer und passiven Schülern ist nach wie vor Unterrichtsmethode Nummer 1. Damit werden Selbständigkeit, Motivation, Kreativität und Teamfähigkeit der Schüler nicht gefördert“ [Kanders u. a. 1997, S. III, Vorwort von Dr. Jürgen Rüttgers]. Besonders auffällig ist, dass der Frontalunterricht insbesondere von jüngeren Lehrkräften häufiger praktiziert wird, als von älteren (vgl. Tabelle 3.5, Fußnote 151).

Die in diesen Studien erhobene Situation an den bundesdeutschen Schulen ist erheblich von den pädagogischen Vorstellungen entfernt, die als lernförderlich ausgewiesen werden: Selbstständigkeit und eigenständiges Arbeiten der Schülerinnen im Unterricht, offene Unterrichtsformen, in denen die Schülerinnen im Mittelpunkt der Planungsprozesse ernsthaft Berücksichtigung finden. Inzwischen kann (nicht zuletzt als Ergebnis der Diskussion um die Leistungsfähigkeit des bundesdeutschen Schulwesens im internationalen Vergleich – siehe Fußnote 148) festgestellt werden, dass zunehmend Initiativen zur Erhöhung der Berücksichtigung schülerorientierter Unterrichtsmethoden im tatsächlich praktizierten Unterricht angeregt werden.<sup>155</sup>

<sup>150</sup> Ausgewiesene Antworten(in Prozent): „So ist es ... sehr oft“ – die weiteren Antwortmöglichkeiten: „... manchmal, ... niemals oder ganz selten“. Die Angaben in den Spalten sind den Tabellen [Kanders u. a. 1997, S. 14–23] entnommen.

<sup>151</sup> „Hierbei praktizieren jüngere Lehrkräfte diese Form signifikant häufiger als ältere“ [Kanders u. a. 1997, S. 14].

<sup>152</sup> S. 17: „Die Schüler bearbeiten in Gruppen Aufgaben.“ – prozentual ausgewiesene Antwort: „So ist es ... sehr oft“.

„38% der Gymnasiasten sagen sogar, niemals oder ganz selten in Gruppen zu arbeiten. Ein deutlich häufigeres Vorkommen dieser Arbeitsweise wünschen Lehrer wie Schüler, etwas defensiv verhalten sich dabei nur die Gymnasiallehrer“ (vgl. [Kanders u. a. 1997, S. 17]).

<sup>153</sup> S. 18: „Die Schüler arbeiten jeder für sich an den gleichen Aufgaben.“ – prozentual ausgewiesene Antwort: „So ist es ... sehr oft“.

<sup>154</sup> Ausgewiesene Antworten (in Prozent): „So ist es ... regelmässig“ – die weiteren Antwortmöglichkeiten: „... ab und zu, ... selten oder nie“. In der Quelle wurde keine schülerbezogene kumulative Angabe veröffentlicht, so dass die Tabellen „Computernutzung im Unterricht:“ in sprachlichen Fächern (S. 25), in naturwissenschaftlichen Fächern (S. 27), mit jeweils 4% herangezogen wurden. Allerdings ist darauf hinzuweisen, dass in den Fächern Arbeitslehre, Wirtschaft und Technik der Einsatz nach Schülerangaben zu 14% regelmässig erfolgt (S. 32).

<sup>155</sup> In dieser Arbeit kann nicht differenziert auf die nationale Diskussion (Strukturreformdebatte, Ressourcenfrage[n]) des Bildungssystems der Bundesrepublik eingegangen werden.

### 3.3 Anwendung der Erkenntnisse der Lerntheorie

Die Darstellung der geschichtlichen Entwicklung soll zu pädagogischer Reflexion im Kontext gesellschaftspolitischer und lerntheoretisch begründeter [Neu-]Orientierung beitragen. Aus der Diskussion zwischen Instrukionalisten und Konstruktivisten (vgl. Stellenwert des Konstruktivismus im Vergleich, S. 40) folgert Rolf SCHULMEISTER: „Für Konstruktivisten wird Wissen aktuell generiert und konstruiert. Diese Vorstellung zwingt zur Aufgabe der Orientierung der Instruktion an Lernzielen. Lernen als Prozeß, Lernen in Wissensgemeinschaften und kontextbezogenen Lernumwelten steht im Mittelpunkt des Konstruktivismus. Es wird daher verständlich, daß das Augenmerk der Konstruktivisten auf gerade jenen höheren Lern- und Denkprozessen wie Interpretieren und Verstehen liegt, die Instrukionalisten bewußt ausgespart haben. Konstruktivisten ersetzen Instruktion durch Lernen“ [Schulmeister 2002, S. 166]. Die Diskussion um die Bedeutung der fundamentalen Ideen versus Kernideen (vgl. 3.2, S. 48) kann ebenfalls dazu beitragen, die Perspektiven auf das Lernen zu ändern.

Die Überlegungen der Konstruktivisten sollen nicht darüber hinwegtäuschen, dass mit ihnen keine Methodik verbunden werden kann. Diese Erkenntnis führt zur Ernüchterung in Bezug auf die Hoffnung, theoriegeleitet »erfolgreiche« Unterrichtsmethoden entwickeln zu können.

Die Auswahl der vorgestellten Elemente tritt besonders in zwei Aspekten für schulisch veranstaltetes Lernen hervor. Es wird deutlich, dass

- ein Wechsel von Modellen des Lernens bei Menschen, die vom „objektiv Gegebenen“ als Objekt des Lernens ausgehen, zur Lernenden als Subjekt des Lernens nahegelegt wird (konstruktivistische Perspektive) und
- ein Wechsel von Modellen des Unterrichts – verstanden als Instruktion – zu Unterrichtsformen, in denen die Schülerinnen in den Mittelpunkt des Unterrichts rücken, angeraten erscheint.

In der Betrachtung historischer Konzeptentwicklungen wird deutlich, dass die Prozesse der pädagogischen Reflexion und Neuorientierung von theoretischen Erkenntnissen eine lange Tradition besitzen. Viele der vorgestellten Ideen wurden jedoch nicht umgesetzt.

In Anbetracht [der] überwältigenden Komplexität pädagogischer Situationen ist es erstaunlich, daß Unterricht überhaupt stattfindet. Die unvermeidliche situative Unvorhersagbarkeit im Unterricht ist eine nahezu grenzenlose Herausforderung für die Lehrer. Lehrer sind mit einem Paradox konfrontiert. Sie sollen tun, was im Prinzip nicht möglich ist.

[Opp und Freytag 1997, S. 275]

Die für die Umsetzung wichtige Frage der Ressourcen wird in dieser Diskussion ausgespart. Wieviel Lernzeit steht zur Verfügung? Sollen nachhaltige Ergebnisse erzielt werden? Sollen kurzfristig „Standards“ erfüllt werden?

#### Was ist guter Unterricht?

Die Fragestellung kann ohne die vorgängige Verständigung über Bezugsnormen und Gütekriterien nicht beantwortet werden. Zur Beantwortung wird in der aktuellen Diskussion nach Beispielen gefragt, die als geglückt gelten und allgemeine Akzeptanz finden. Das Ergebnis ist eine Interpretationsleistung und keine Methode, mit der Unterricht gemessen<sup>156</sup> werden kann.

Zur Charakterisierung „guten Unterrichts“ wird eine Beschränkung auf konkreten Unterricht vorgenommen, auch wenn dadurch Elemente aus dem Blick geraten, die für die Gesamtsituation wichtig sind (exemplarisch: die sogenannte „15.000-Stunden-Studie“ [Maughan u. a. 1980], die bereits erwähnte Untersuchung zum Bild der Schule [Kanders u. a. 1997]). Im Kontext der Veröffentlichung der Ergebnisse aktueller

<sup>156</sup> Messen ist noch nicht Beurteilen.

internationaler Vergleichsuntersuchungen ist in den letzten Jahren die Diskussion um die Evaluation der in den Schulen geleisteten Arbeit in das Licht der Öffentlichkeit gerückt worden.<sup>157</sup> Die Diskussion in der Öffentlichkeit wird sehr stark von dem [internationalen] Ranking bestimmt, so dass Fragen der Qualität m. E. auf „das Zählbare“ reduziert werden.

Zielstellung und Konzeption der PISA-Studien werden im Folgenden kurz dargestellt, um Elemente zu identifizieren, die auch für die Evaluation des Informatikunterrichts bedeutsam sind. „Die allgemeinen Zielsetzungen von PISA [...] lassen sich knapp zusammenfassen. Primäre Aufgabe des Programms ist es, [...] Prozess- und Ertragsindikatoren zur Verfügung zu stellen, die [...] zur Verbesserung der nationalen Bildungssysteme brauchbar sind. [...] Die Indikatoren beziehen sich auf die Bereiche Lesekompetenz (Reading Literacy), mathematische Grundbildung (Mathematical Literacy), naturwissenschaftliche Grundbildung (Scientific Literacy) und fächerübergreifende Kompetenzen (Cross-Curricular Competencies). Zu den fächerübergreifenden Kompetenzen gehören im ersten Zyklus, wenn man einmal vom Leseverständnis als fächerübergreifender Basiskompetenz absieht, Merkmale selbstregulierten Lernens und Vertrautheit mit Computern. [...] Darüber hinaus wurden auch Aspekte von Kooperation und Kommunikation und von Problemlösefähigkeit verstanden als Planungskompetenz in komplexen Alltagssituationen zum Gegenstand der Untersuchung gemacht“ [Baumert u. a. 2002, S. 11]. Im Zusammenhang mit der Zielsetzung dieser Arbeit ist bemerkenswert, dass der Begriff Informatik nicht benutzt wird. Ebenso wenig wird der (geläufige englische) Begriff Computer Literacy verwendet. Offenbar wurde in diesem Feld in den Beratungen der OECD kein fachlicher Rat der Informatikdidaktik eingeholt.

In der konkreten Planung für weitere Studien [vgl. PISA 2003 – <http://www.ipn.uni-kiel.de/projekte/pisa/index.html>] soll der Faktor Cross-Curricular Competencies (CCC) im Sinne von Problemlösen berücksichtigt werden. „Man muss sich darüber im Klaren sein, dass die PISA-Tests mit ihrem Verzicht auf transnationale curriculare Validität und der Konzentration auf die Erfassung von Basiskompetenzen durchaus ein [...] bildungstheoretisches Konzept mit sich führen, das normativ ist. Dieses inhaltliche Benchmarking (Vergleichsnormierung) ist auch bei der Darstellung und Interpretation der Ergebnisse bewusst zu halten“ [Baumert u. a. 2002, S. 15].

„Die PISA zu Grunde liegende Philosophie richtet sich auf die Funktionalität der bis zum Ende der Pflichtschulzeit erworbenen Kompetenzen für die Lebensbewältigung im jungen Erwachsenenalter und deren Anschlussfähigkeit für kontinuierliches Weiterlernen in der Lebensspanne. Kennzeichnend hierfür ist die funktionale Sicht auf Kompetenzen als basale Kulturwerkzeuge“ [Baumert u. a. 2002, S. 58].

„Der erweiterte Lern- und Leistungsbegriff umfasst nicht mehr nur Speicherung oder Reproduktion; er ist wesentlich ausgerichtet auf die Einrichtung einer Metaebene, von der aus sowohl die Intentionen und Ansprüche als auch die Prozeduren und ihre Produkte einschätzbar werden und der mitverantwortlichen Verbesserung zugeführt werden können. Damit ist gute Unterrichtung eine Veranstaltung, die aufklärerisch im ursprünglichen Sinn ist und sinnstiftend wird. Dies wird auch den materiellen Unterrichtsergebnissen sowie der Motiviertheit zum Lernen zugute kommen (für uns war wichtig, das kann ich anderen jetzt erklären, davon wissen wir im Moment noch zu wenig, das haben wir ganz gut herausgearbeitet)“ [Bönsch 1998, S. 29].

## Unterrichtsbeobachtung – Evaluation

Bereits im Abschnitt 3.2 wurden Ergebnisse dargestellt, die nicht durch analytische, sondern durch empirische Untersuchungsmethoden gewonnen wurden. Dort kamen Verfahren der Unterrichtsbeobachtung und der Introspektion zum Einsatz, um den Anteil verschiedener Unterrichtsformen im „Regelunterricht“ zu quantifizieren. Im Zusammenhang mit internationalen Vergleichsstudien<sup>158</sup> der letzten Jahren wurden Unterrichtsbeobachtungsinstrumente eingesetzt, um Unterrichtsmuster<sup>159</sup> zu identifizieren.<sup>160</sup> Zu Beginn der Untersuchungen wurden dabei einzelne quantitative Indikatoren geschätzt (vgl. [Klieme und Siegle

<sup>157</sup> Zur Evaluation des Informatikunterrichts wird in diesen Studien kein expliziter Beitrag geleistet.

<sup>158</sup> Die folgenden Hinweise beziehen sich auf TIMSS (vgl. [Baumert u. a. 1998]).

<sup>159</sup> durch Beobachtung in internationalen Videostudien

<sup>160</sup> Ausgewertet wurden Videomitschnitte nordamerikanischer, deutscher und japanischer Mathematikunterrichtsstunden.

2001, S. 84]). Durch die Auswertung dieser Indikatoren konnten die folgenden vier Cluster<sup>161</sup> gebildet werden:

- (1) Stark direkter Unterricht,
- (2) wenig strukturierter und anregungsarmer Unterricht,
- (3) ausgewogenes Unterrichtsprofil (mittlere Direktivität und kognitive Aktivierung bei hoher Schülerorientierung), hoher Anteil an Einzel- oder Gruppenarbeit und häufige Schülerfragen
- (4) anspruchsvoller (direktiver, schülerorientierter und zugleich kognitiv stark aktivierender) Unterricht.

Die Cluster können im Rückgriff auf methodisch-didaktische Merkmale und Interaktionsmerkmale näher charakterisiert werden. Beispielsweise zeichnet sich Cluster (3) durch einen hohen Anteil an Einzel- oder Gruppenarbeit und häufige Schülerfragen aus. [...] Die Cluster stellen [primär] Unterrichtsmuster dar, die durch den Lehr-Lern-Prozess bestimmt sind [...]. Das Unterrichtsmuster ist aber vom Ausgangszustand der Klasse (Fachleistung und Fachinteresse der Schüler) abhängig und erklärt seinerseits einen Teil der Leistungs- und Interessensentwicklung über ein Schuljahr hinweg: Sogenannte Optimalklassen, die hinsichtlich Leistung und Interesse besonders positive Zuwächse verzeichnen, finden sich vorzugsweise im Cluster (3).

[Klieme und Siegle 2001]

Um die Unterrichtsmuster zu konkretisieren, werden Begriffe wie Unterrichtsskript oder Choreographie des Unterrichts eingeführt, die zur Beschreibung von Unterrichtssituationen verwendet werden. Zu den Choreographien des Unterrichts in der Bundesrepublik werden von Jürgen BAUMERT Ergebnisse empirisch-quantitativer Studien vorgestellt.

Die Unterrichtsstunde beginnt mit einem komplexen Problem, das in seiner Grundstruktur im nachfolgenden Gespräch entfaltet und mit dem Abschluss der Stunde zu einer konzisen Lösung geführt werden soll. [...] Unsere Videoaufzeichnungen [...] bestätigen in vielen Fällen das Ergebnis mathematikdidaktischer Fallstudien, in denen diese Unterrichtsform als schrittweise Trivialisierung eines komplexen Ausgangsproblems beschrieben wird. Die Lehrkraft beginnt in der Einführungsphase mit einer anspruchsvollen Aufgabenstellung, die Schüler allerdings nicht in dieser Komplexität zu bearbeiten haben – wie man dies etwa in mathematischen Musterstunden in Japan beobachten kann –, sondern die sukzessiv in Teilleistungen und elementare Fragen zerlegt wird, die zu beantworten Schülern manchmal geradezu peinlich sein kann [...] Die Folge: Man [<sup>162</sup>] tastet sich mit assoziativen Beiträgen an die Vorstellung der Lehrkraft heran, bis die Glieder des Reißverschlusses einrasten und Teilschritte abgearbeitet werden können. Die beiden Ziele, Offenheit während des Gespräches und konvergente Steuerung auf ein Ziel hin, sind also nur schwer zu vereinbaren. Erschwerend kommt hinzu, das der Zeittakt einer Unterrichtsstunde unerbittlich ist. [...] Die] eigentliche Problematik des fragend-entwickelnden Unterrichts [...] kann folgendermaßen beschrieben werden]: Schüler würden nicht auf der Ebene des komplexen Ausgangsproblems kognitiv aktiviert, sondern auf der Ebene von Teilprozessen im Sinne von Reproduktion, Assoziation und einfachen Operationen. Weder Lehrer noch Schüler haben in dem teilweise sehr schnellen Dialog eine Chance, wirklich nachzudenken, obwohl der Idee nach von allen, faktisch jedenfalls von der Lehrkraft ständige Konzentration verlangt wird. Stockt das Gespräch, entsteht Peinlichkeit. [...] Zur Abhilfe wird empfohlen: ] Das Unterrichtsgespräch hat seinen festen Platz, aber es wechselt mit anderen Phasen ab, die Schüler zum selbstständigen Nachdenken, aber auch zum Gespräch miteinander bringen [...] In diesen Phasen gewinnt die Lehrkraft jenen Spielraum, der für eine gute Diagnostik, für die Sichtung alternativer Lösungen, die Entdeckung produktiver Fehler und besonders intelligenter Beiträge sowie die Vorbereitung eines Unterrichtsgesprächs, das von geordneten Schülerbeiträgen ausgeht, notwendig ist.

<sup>161</sup> cluster (engl.) := Anhäufung; Gruppen mit ähnlichen Merkmalsausprägungen – vgl. [Bortz und Döring 1995, S. 354f]

<sup>162</sup> gemeint sind die Schülerinnen

[Baumert 2002, S. 143ff]

Diese Ergebnisse finden zunehmend Berücksichtigung in Überlegungen zur inneren Reform des Unterrichts. Sie zeigen, in welche Richtung Unterricht gestaltet werden kann, damit sowohl den Interessen der Schülerinnen für die Lerngegenstände aber auch den berechtigten Leistungsanforderungen Rechnung getragen werden kann.

Aus den oben angeführten Darstellungen wird die folgende Hypothese abgeleitet.

#### **Arbeitshypothese**

- Die Bedeutung des Subjekts für erfolgreiche Lernprozesse im Sinne institutionell-politischer Vorgaben führt in der Konsequenz zu Schülerorientierung und projektorientierten Unterrichtsformen – die durch informatische Methoden bereits vorausgesetzt werden.<sup>163</sup>

---

<sup>163</sup> Problemlösen, Projektarbeit, Spiral- und Stufenmodelle (hierarchische [De-]Konstruktion) sind wichtige Elemente der Didaktik der Informatik.

Ein jeder solcher Versuch fördert nicht,  
sondern entfernt den Schüler von dem ins  
Auge gefaßten Ziel, wie die rohe Hand eines  
Menschen, die eine Knospe entfalten wollte,  
nur alles zerdrücken würde, wenn sie jedes  
Blatt einzeln öffnete.

---

[Tolstoj 1862, S. 72]

## Kapitel 4

# Stand der Schulinformatik

Um die chronologische Linie in den folgenden Abschnitten zu verdeutlichen, werden Jahreszahlen als Marginalien angegeben.

### 4.1 Fachdidaktische Empfehlungen der Informatik

Da bis 1996 noch keine Forschungsgruppen zur Didaktik der Informatik existierten, stammen die ersten wissenschaftlichen Studien zur Lehrdisziplin der Informatik von Fachwissenschaftlerinnen. Es wird nicht der gesamte historische Weg nachgezeichnet – vielmehr werden solche Arbeiten ausgewählt, die für den Fortgang der Entwicklung unter den Aspekten überdauernder Fragestellungen und Nachhaltigkeit bedeutsam sind.

Die erste Empfehlung der GI zum Informatikunterricht an allgemein bildenden Schulen wird veröffentlicht. 1976  
Besonders wichtig ist die Forderung nach einer verpflichtenden Informatikbildung<sup>164</sup> für alle Schulabgänger. Die Schwerpunkte dieser Informatikbildung liegen auf algorithmischen Lösungen, auf der Entmystifizierung von Informatiksystemen und deren Einbettung in die Gesellschaft.

„Ein wesentlicher Platz im Informatik-Unterricht kommt einer Projektphase zu, in der das selbständige Entwickeln von Algorithmen in der Form von spielerischem Lernen eingeübt werden sollte“ [Brauer u. a. 1976].

Mit diesem Vorschlag der Gesellschaft für Informatik wird der Stand der Fachwissenschaft hinsichtlich der inhaltlichen Orientierung des Informatikunterrichts an algorithmischen Strukturen für eine Umsetzung in der Schule aufbereitet. Die Verflechtung der unterrichtlichen Umsetzung mit Fragestellungen aus dem Anwendungskontext und den Auswirkungen der Informatik auf die Gesellschaft zeigen, dass Informatikerinnen die Verantwortung für die Gestaltung von Lösungen als Bestandteil des Prozesses zur Problemlösung berücksichtigt wissen wollen. Dies muss nach Meinung der Autoren dieser Empfehlungen auch im Schulfach Informatik beachtet und damit Gegenstand unterrichtlicher Praxis werden.

Der Vorschlag, eine Projektphase als Bestandteil des Informatikunterrichts vorzusehen, ist bemerkenswert, da er eine methodisch-didaktische Qualität auf der Ebene dieser Empfehlungen erzielt.<sup>165</sup> Zu dem Erfolg dieser Unterrichtsempfehlungen haben die in den Anhängen beigefügten Beispiele in besonderer Weise beigetragen. Trotzdem blieben noch viele informatikdidaktische Fragen offen – z. B. solche zum Identifizieren von kognitiven Problembereichen einer Einführung von Schülerinnen in die Informatik, die Frage nach dem „Verhältnis von Theorie und Praxis“ im Informatikunterricht.

Aussagen dazu, worauf eine Lehrerin im Detail zu achten hat, finden sich in den Lehrwerken von Helmut BALZERT ab 1976: „Um das "Klebenbleiben" an gerätespezifischen Kenntnissen zu Ungunsten des 1977

---

<sup>164</sup> Informatikbildung wird in dieser Arbeit als Informatische Bildung bezeichnet

<sup>165</sup> vgl. Abschnitt 3.2, S. 43

weiteren Verständnisses zu vermeiden, sollte unbedingt zunächst die Problemanalyse, Modellbildung und algorithmische Problemlösung an einer Vielzahl von Beispielen aus unterschiedlichen Problembereichen geübt werden. [...] Es kommt nicht darauf an, daß eine Programmiersprache in kürzester Zeit erlernbar ist, sondern daß sie problemadäquate Sprachstrukturen zur Verfügung stellt, die der menschlichen Denkweise und dem menschlichen Problemlöseverhalten angepaßt sind“ [Balzert 1977b].<sup>166</sup>

„Als didaktisch schwierige Gebiete der Informatik erweisen sich das Variablen-, Feld- und Prozedurkonzept. Auf diese grundlegenden Programmierprinzipien wird daher besonders ausführlich eingegangen“ [Balzert 1979, S. 8].

Damit wurde eine klare Orientierung an dem informatischen Problemlösungsprozess als didaktische Gestaltung für den Informatikunterricht vorgelegt. Besonders hervorzuheben ist die Beispielauswahl, die nicht aus dem zu diesem Zeitpunkt in den meisten Schulen noch üblichen mathematisch-technischen Aufgabebereichen stammen. Es werden Fragen der Aufwandsabschätzung der dynamischen und statischen Komplexität thematisiert. Auf Fragen des Persönlichkeitsschutzes wird am Rande eingegangen.

Während die vorherigen Arbeiten sehr knapp aus der Bedeutung der aufstrebenden Wissenschaftsdisziplin Informatik auf die Existenz eines gleichnamigen Schulfaches schlossen<sup>167</sup>, führt Volker CLAUS eine umfassendere Diskussion. Dazu er schlägt vor, als Begründungen für das Schulfach Informatik nur solche informatikspezifische Argumente zu akzeptieren, die eine Abgrenzung gegenüber anderen Disziplinen ermöglichen.

„Diese Denkweise in dynamischen Abläufen und im Prozess [...] und das systematische, möglichst fehlervermeidende Entwickeln von Programmen mit Hilfe der Methode der schrittweisen Verfeinerung (strukturiertes Programmieren) sind grundlegend für die neue Wissenschaft »Informatik«. [...] Durch diese Denkweise grenzt sich die Informatik zugleich deutlich von der Mathematik ab. Die Mathematik trainiert das Denken in statischen abstrakten Strukturen (”Denken in abstrakten Räumen”). Dies ist jedoch für dynamische Prozesse unzureichend“ [Claus 1977, S. 22f].

CLAUS geht davon aus und erwartet, dass Informatik als eigenständiges allgemein bildendes Fach in der Schule eingeführt werden wird. Seine Ausführungen stellen ein klares Plädoyer für die schülerorientierte Ausarbeitung und Ausgestaltung konkreter Vorschläge für den Informatikunterricht für alle Schülerinnen dar.

1979 Die von Gerhard GOOS formulierte Qualität der Informatik als konstruktive Wissenschaft eignet sich sehr gut zur weiteren Ausgestaltung der GI-Empfehlungen von 1976 [Brauer u. a. 1976]. „Der Aspekt konstruktive Wissenschaft scheint mir das überzeugendste Argument für die Einführung der Informatik als Schulfach“ [Goos 1979, S. 2f].

1982 Erstaunlicherweise dauerte es 17 Jahre, bis die Bedeutung der Informatik bei der Entstehung einer 4. Kulturtechnik [Arlt 1982, S. 47] einen breiten Konsens unter Bildungsexperten fand<sup>168</sup>. Ein Ziel der vorliegenden Arbeit besteht darin, für diese neue Kulturtechnik Konzepte zur Umsetzung und Gestaltung zu entwickeln.

1985 Während die oben dokumentierten Arbeiten der Fachwissenschaft die Leistungsfähigkeit der Informatik nicht in Frage stellten, setzt sich David L. PARNAS kritisch mit den Grenzen von Software-Engineering-Methoden auseinander.

<sup>166</sup> Für die Zitate wurde die 2. Auflage [Balzert 1979, S. 7] herangezogen

<sup>167</sup> „Seit Anfang der 70er Jahre werden an Universitäten und Fachhochschulen in der Bundesrepublik Informatiker ausgebildet. [...] wird in den meisten Bundesländern Informatik als Schulfach in der Sekundarstufe II eingeführt bzw. erprobt. Überlegungen werden angestellt, Informatik auch in der Sekundarstufe I und in den Primarstufe zu unterrichten“ [Balzert 1977a, S. 49].

„Die wachsende Bedeutung der Datenverarbeitung in Wirtschaft und Verwaltung sowie der Einsatz von Datenverarbeitungsanlagen in fast allen Lebensbereichen machen es erforderlich, daß Schüler an allgemein- und berufsbildenden Schulen eine Grundausbildung in Informatik erhalten“ [Balzert 1977c, S. 1].

„Der zunehmende Einsatz von Rechenanlagen verlangt von jedem Schulabgänger, daß er die Fähigkeit besitzt, einfache Probleme zu analysieren, deren Lösungsabläufe zu entwickeln und diese so zu beschreiben, daß sie schließlich auf einer Rechenanlage ausgeführt werden können. [...] Grundlegend für alle Bereiche der Informatik und ihrer Anwendungen ist das algorithmische Denken“ [Brauer u. a. 1976, S. 35].

<sup>168</sup> siehe: Erfurter Resolution in [Thoma 2001] – vgl. auch <http://www.gki.at/resolut.htm> und <http://www.i-12.org/resolut.htm>

„Software wird nicht dann zur Benutzung freigegeben, wenn sie nachweisbar korrekt ist, sondern wenn die Häufigkeit, mit der neue Fehler entdeckt werden, auf ein für die Geschäftsleitung akzeptables Niveau gesunken ist. Benutzer lernen, Fehler zu erwarten, und ihnen wird oft erklärt, wie sie bis zur Verbesserung des Programms die Fehler umgehen können“ [Parnas 1986, S. 59].

Im Folgenden schildert PARNAS sehr ausführlich die Vorteile und Grenzen der strukturierten Programmierung, die Verwendung formal spezifizierter abstrakter Schnittstellen zum Verbergen von Information über die Lösung einer Teilaufgabe und die Verwendung kooperierender, sequenzieller Prozesse als Hilfsmittel zur Bewältigung der Komplexität. Bis zum heutigen Tag besteht ein deutliches Ausbildungsdefizit zur Methodenkritik in der Informatikausbildung.<sup>169</sup> Mit der vorliegenden Arbeit soll ein Beitrag geleistet werden, dieses Defizit zu verringern.

Mit den GI-Empfehlungen von 1976 [Brauer u. a. 1976] wurden Struktogramme zur statischen Veranschaulichung von dynamischen Abläufen in Informatiksystemen eingesetzt. Darauf gründete die Annahme, dass man für alle schwierigen Bildungsgegenstände der Informatik die passende Veranschaulichung entwickeln könnte. Fredrick BROOKS widerlegte das mit seiner Analyse der Eigenschaften von Software:

1986

„Invisibility: Software is invisible and unvisualizable. [...] In spite of progress in restricting and simplifying the structures of software, they remain inherently unvisualizable, and thus do not permit the mind to use some of its most powerful conceptual tools. This lack not only impedes the process of design within one mind, it severely hinders communication among minds“ [Brooks 1986, Brooks 1987].

Die Schlussfolgerung aus dieser Erkenntnis kann nicht sein, dass wissenschaftsorientierter Informatikunterricht in allgemein bildenden Schulen unmöglich ist. Vielmehr gilt es, die Mittel der Informatik zu nutzen, die eine Kommunikation über „Software“ erlauben, d. h. mit den Mitteln der Abstraktion und der Modularisierung den Fokus auf konkrete Strukturen sowie auf einzelne Elemente zu richten. Damit fällt der Didaktik die Aufgabe zu, solche Mittel<sup>170</sup> für den „Anwendungsbereich“ der allgemeinen Bildung zu gestalten und ihre Umsetzungsmöglichkeit im Lernprozess zu begleiten. Darüber hinaus sollte die Modellierung exemplarisch bis zur Implementierung auf Quellcodebasis vorangebracht werden.

Für den Bereich der Datenbanktechnik begründet Peter C. LOCKEMANN [Lockemann 1986] fundamentale Ideen – Konsistenz, Konkurrenz und Persistenz, die in der Hochschulbildung enthalten sein sollen. Der Ansatz ist für die Didaktik der Informatik bedeutsam, da 1989 beginnend, eine breite Diskussion zur Bedeutung von fundamentalen Ideen der Informatik in der Allgemeinbildung einsetzte, der bis heute die informatikdidaktische Forschung beeinflusst.

Joseph WEIZENBAUM: „Auf welcher Ebene wollen wir den Kindern den Computer erklären? [...] Dann kann man vielleicht etwas sagen über [...] höhere Computersprachen. Wie funktioniert der Computer, daß er diese Sprache versteht? Dann gibt es das, was wir [...] Maschinensprache nennen – ja, und wie funktioniert das? Da muß man die Schalter des elektrischen Gerätes, seine Architektur, erklären. [...] Wenn man [das] sagt [...] dann dauert es nicht lange, bis wir in der Quantenmechanik [...], bis wir in der Physik sind“ [Weizenbaum 1990, S. 81f].

1990

Die von WEIZENBAUM aufgeworfenen Fragen machen deutlich, dass unter den Informatikerinnen zu diesem Zeitpunkt keine klare Vorstellung darüber entwickelt ist, was die Grundlage für das allgemein bildende Schulfach Informatik inhaltlich thematisiert werden soll. Es wird deutlich, dass sich aus der Beantwortung der Frage nach dem »Was« immer auch die Frage nach Vollständigkeit stellt. Da die fachwissenschaftliche Grundlagen der Informatik sowohl auf Wurzeln der Physik, wie auch der Mathematik basieren, kommt WEIZENBAUM in diesem Fall zu einer Fragestellung, die von dem Schulfach Physik zu klären ist – nicht von dem Schulfach Informatik, wie ich meine. Andererseits weist WEIZENBAUM mit dieser Antwort deutlich den fächerverknüpfenden<sup>171</sup> Charakter von Fragestellungen auf, die die technischen Wirkprinzipien von Informatiksystemen betreffen.

<sup>169</sup> „I was told to teach whatever I thought was important. [...] I fear that the same thing happens in many computer science departments today“ [Parnas 2002, S. 401].

<sup>170</sup> Werkzeuge – Informatiksysteme, die den Lernprozess in konstruktiver Weise unterstützen

<sup>171</sup> zur Begrifflichkeit (fächer\*) vgl. Tabelle 3.1, S. 34

## Zwischenresümee 1976–1991

1991

In den Jahren von 1976 bis 1990 entstand eine Ansammlung von Grundkonzepten der Informatik, die alle im Informatikunterricht allgemein bildender Schulen gelehrt werden sollten. Die einzelnen „Perlen der Informatik“ konnten nicht zu einer tragfähigen Struktur für den allgemein bildenden Informatikunterricht verdichtet werden: „Ein Potpourri von unzusammenhängenden Grundideen, auch wenn jede treffend vorgestellt wird, empfehle ich keineswegs als Einführung in die Informatik“ [Nievergelt 1991, 5. Einige klassische Beispiele].<sup>172</sup>

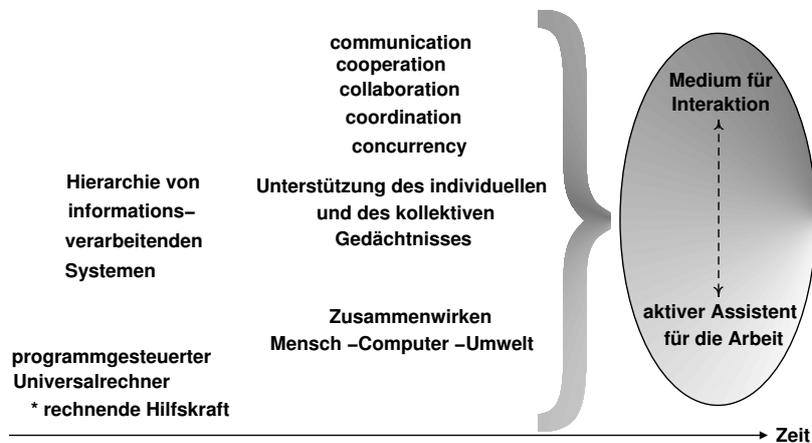
Die Motivation für der vorliegenden Forschungsarbeit gründet genau auf diesem bis heute existierenden Defizit. Zwar wurde inzwischen die GI-Empfehlung eines Gesamtkonzepts der informatischen Bildung von der Primar- bis zur Sekundarstufe veröffentlicht<sup>173</sup>, aber von diesen bildungspolitischen Forderungen bis zur Umsetzung in der Unterrichtspraxis sollten die folgenden drei Schritte realisiert sein:

- ein konzeptioneller Rahmen, in dem die Grundideen der Informatik flexibel (Varianten sind möglich) und zugleich logisch verknüpft (Beliebigkeit ist ausgeschlossen) eingebunden werden,
- qualitativ hochwertige, evaluierte Beispiele und
- entsprechende Richtlinien und Lehrpläne.

1992

Da die Informatik eine junge, sich stürmisch entwickelnde Wissenschaft ist, tritt der Zwang auf, Erkenntnissprünge angemessen in die Lehrdisziplin zu transformieren. Unter dem Titel „Informatik – das neue Paradigma“ wird von Ute und Wilfried BRAUER ein Beitrag vorgelegt, der zu einer grundsätzlichen Neubewertung der Fachstruktur auffordert:

„Als neues Paradigma der Informatik stellt sich ein Computersystem somit dar als Gruppe gleichrangiger, selbständiger, einigermaßen intelligenter Akteure, die bestimmte Aufgaben erledigen und dazu miteinander und mit der Umgebung interagieren“ [Brauer und Brauer 1992].<sup>174</sup>



nach [Brauer und Brauer 1992]

Abbildung 4.1: Neubewertung der Fachstruktur

Meine Schlussfolgerung besteht darin, kollaborative<sup>175</sup> Systeme als Gegenstand und Mittel in der Informa-

<sup>172</sup> Diese Aussage wird in [Nievergelt 1993, S. 7] wiederholt.

<sup>173</sup> siehe [Gesellschaft für Informatik 2000]

<sup>174</sup> Für das Zitat wurde [Brauer und Brauer 1995, S. 28] herangezogen.

<sup>175</sup> Der Begriff „kollaborativ“ ist bis heute im bundesdeutschen Sprachgebrauch negativ belegt (wg. Kollaboration  $\cong$  Zusammenarbeit mit dem „Feind“).

tikbildung einzusetzen. Stellvertretend sei hier als Beispiel Basic System Cooperative Workspace (BSCW) [Appelt und Busbach 1996] genannt. Wobei darauf hingewiesen werden muss, dass nicht das konkrete System, sondern dessen Einbindung in den konzeptionellen Unterrichtsrahmen zur Kompetenzentwicklung bei den Schülerinnen beiträgt. Dabei kommt es wesentlich auf die didaktische Gestaltung (z. B. Schreibtischmetapher für gemeinsames Arbeiten) und die Abbildung von Gruppenstrukturen in „virtuelle Räume“ an.

Vielfach dominieren im Informatikunterricht nicht die zentralen Fachkonzepte, sondern Produkte und ihre Bedienung/Benutzung als Missverständnis einer anwendungsorientierten Bildung. Hier kann die Informatik von anderen Wissenschaften mit Ingenieurbezug lernen, die sehr wohl zwischen den technischen Artefakten und den Basiskonzepten, die deren Konstruktion zugrunde liegen in unterschiedliche Bildungsgänge verzweigen.

Mary SHAW fordert: „Let’s organize our courses around ideas rather than around artifacts. This helps make the objectives of the course clear to both students and faculty. Engineering schools don’t teach boiler design – they teach thermodynamics. Yet two of the mainstay software courses – “compiler construction” and “operating systems” – are system-artifact dinosaurs“ [Shaw 1992].

Eine solche Trennung der Kompetenzen brächte Transparenz in die Rolle des Unterrichtsfachs Informatik, in dem die Basiskonzepte der Softwaretechnik zu erlernen sind – ohne Softwareentwickler ausbilden zu wollen.

WEIZENBAUM : „Nur sehr selten stimmen die Ergebnisse eines Experiments mit den in der Theorie errechneten überein. Das ist eine sehr, sehr wichtige Einsicht – ich kann sie gar nicht stark genug betonen. Vor allem die Lehrer sollten dies wissen und darüber mit den Schülern sprechen. Das ist nur ein ganz einfaches Beispiel der größeren Sache »Modell«. Was wir mit Computern machen, sind fast alles Simulationen, Modelle. Da sollte man schon etwas über Modelle wissen. Ich frage mich, wie viele Lehrer und Lehrerinnen in der ganzen Welt überhaupt etwas über Modelltheorien wissen“ [Weizenbaum 1992, S. 76].

Informatische Modellbildung ist eine der zentralen Strategien der Fachwissenschaft (vgl. Abschnitt 2.3, S. 16). Dies muss m. E. bei der curricularen Gestaltung des Informatikunterrichts berücksichtigt werden.<sup>176</sup>

Im Kontext der Diskussion um die Folgen der ersten Softwarekrise [Naur und Randell 1969] verdienen Entwicklungen um die Aufkündigung des Phasenmodells (Wasserfallmodell) im Zusammenhang evolutionärer Systementwicklung eine besondere Beachtung (vgl. [Floyd 1994, S. 35]). Interessant ist, dass in den von Christiane FLOYD beschriebenen Elementen der Design-Sicht bzgl. der Produktionssicht die Verschränkung von Softwareentwicklerinnen und Anwenderinnen durch gemeinsames Lernen gefordert wird. „Maßgeblich für das Gelingen sind Exploration, Experimente, Kommunikation und Reflexion zwischen den Beteiligten“ [Floyd 1994, S. 33]. „Die technische Unterstützung von Lernprozessen bei der Softwareentwicklung erfolgt durch exploratives, experimentelles und evolutionäres Prototyping ...“ [Floyd 1994, S. 35].

Für das Gelingen dieser Prozesse sind nach meiner Einschätzung Voraussetzungen zu schaffen, die zur allgemeinen Bildung zu zählen sind, d. h. die in der Schule für alle Schülerinnen Bestandteil des allgemein bildenden Unterrichts werden müssen. Um eine aktive Rolle in der partizipativen Softwareentwicklung einnehmen zu können, bedarf es der Kenntnis grundlegender Möglichkeiten zur Gestaltung von Informatiksystemen. Diese bestehen u. a. darin, prinzipielle Konzepte, ihre Möglichkeiten, aber auch ihre Grenzen zu kennen. Dann kann der oben beschriebene Prozess erfolgreich umgesetzt werden.

Niklaus WIRTH: „As teachers, however, we recognize another value in programming: it is in essence the construction of abstractions, the engineering of (abstract) machines“ [Wirth 1999, S. 3]. Im Kontext der Informatik und ihrer Didaktik wird häufig die Frage nach dem Stellenwert von entwickelten Problemlösungen in Form von Programmen gestellt. Gerade auf dem Hintergrund moderner Informatiksysteme mit ihren schier unerschöpflichen Ressourcen wird nach WIRTH zu wenig Kraft in die Lehre zur Entwicklung effizienter, ingenieurmässiger Lösungen investiert. Nicht nur WIRTH<sup>177</sup> sieht genau hier ein zentrales

<sup>176</sup> Es soll nicht unterschlagen werden, dass WEIZENBAUM sich gegen „Computer in der Schule“ ausspricht.

<sup>177</sup> vgl. auch PARNAS [Parnas 1989]

Handlungsfeld der Informatik.

Eine Konsequenz für die Gestaltung didaktisch nützlicher, unterrichtlich umsetzbarer Zielvorgaben besteht nach meiner Überzeugung darin, die konzeptionellen Ergebnisse zum Einen in möglichst verständlicher und klarer Art und Weise umzusetzen, andererseits durchaus grundlegende Überlegungen zur Komplexität von Problemlösungen anzustellen, die am Beispiel unterrichtlich umgesetzt werden.

2002

Uwe SCHÖNING: „Die Informatik hält eine Menge von Konzepten, Modellen, Algorithmen, Beschreibungsmethoden (oder schlichtweg: *Ideen*) bereit, die zum einen dazu dienen, komplexe Sachverhalte und Wirkungsmechanismen zu veranschaulichen und zu visualisieren. Zum anderen dienen sie dazu, die hierdurch modellierten Strukturen oder Sachverhalte im Computer weiter bearbeiten, analysieren und transformieren zu können. [...] Bei manchen der behandelten Themen [...] mag sich mancher fragen, warum sie unter dem Oberbegriff „Informatik“ verstanden werden. [...] Was die erwähnten Konzepte jedoch „informatisch“ macht, ist die algorithmische Behandlung derselben. [...] Diese Sichtweise finde ich ganz fundamental und bin davon überzeugt, dass Algorithmik und formale Konzepte der Informatik genauso zum allgemein Bildungsgut gehören (und dementsprechend an den Schulen gelehrt werden) sollten wie beispielsweise das Prozent- oder Bruchrechnen im Rahmen der Mathematikausbildung.“

[Schöning 2002, S. Vff (Zitatenkollage)].

Eine Verbindung zu der bereits diskutierten wissenschaftstheoretisch bedeutsamen Frage nach „Informatik als dritte Säule der wissenschaftlichen Arbeitsweise“ (vgl. Abschnitt 2.2, S. 14) wird durch diese Sicht konkretisiert. Zudem wird von SCHÖNING mit einer inhaltlich-fachmethodisch begründeten Argumentation die Forderung nach dem allgemein bildenden Schulfach Informatik in der Sekundarstufe I erhoben.

### **Prüfung/Schärfen/Verdichten der Arbeitshypothesen zu Wissenschaftstheorie und Informatik**

Die Arbeitshypothesen (vgl. Kapitel 2, S. 29) werden wiederholt, um den Prozess zur Überarbeitung nachvollziehbar zu verdeutlichen.

1. Die Fachwissenschaft Informatik unterscheidet sich hinsichtlich des Gegenstandsbereichs und ihrer Methoden von den tradierten Wissenschaften.<sup>178</sup>
2. Die informatische Modellierung verändert durch autooperationale Form den modellierten „Weltausschnitt“.
3. Informatik als dritte Säule der wissenschaftlichen Arbeitsweise neben theoretischem und experimentellem Vorgehen erweitert die tradierten (Natur- und Ingenieur-) Wissenschaften um eine neue Qualität.

Zu 1.: Der Begriff „Gegenstandsbereich“ ist unscharf und bedarf nach den vorgestellten Überlegungen der Konkretisierung.

Dabei ist im Zusammenhang mit dem Schulfach Informatik insbesondere die allgemeine Bildung zu berücksichtigten. Diese Anforderung hat inhaltlich-curriculare Konsequenzen.

Für die „Besonderheit“ informatischer Methoden wurden Belege dokumentiert. Darüber hinaus ist das Kriterium „Informatik als unverzichtbarer Teil der allgemeinen Bildung“ heranzuziehen.

Zu 2.: Das Problem der Vermittlung ist offensichtlich:

Allgemeine Bildung als Ziel organisierter Lernprozesse findet nicht in einem Umfeld statt, in dem die Entwicklung von Arbeitsprozessen Bestandteil der Erfahrungswelt der Schülerinnen ist.

Daher muss die These sehr genau und konkret auf die Lerngruppe bezogen werden. [Mögliche] Erfahrungen der Schülerinnen sind in den Kontext des konkreten Unterrichts einzubeziehen.

<sup>178</sup> Die Gegenstände der Informatik sind wie bei Philosophie und Mathematik immateriell. Darüber hinaus ist erkennbar, dass sie mit ihren Methoden die Lücke zwischen Anwendungsbereich und Formalisierung schließt.

Jahr	Autorin	Quelle	Charakterisierung	Bemerkungen
1976	GI	[Brauer u. a. 1976]	GI-Empfehlungen	Projektmethode, imperativ, Beispiele, Struktogramme
1977	Helmut BALZERT	[Balzert 1977b]	Schulbuch	Problemanalyse (Schema), Modellbildung
	Volker CLAUS	[Claus 1977]	Allgemeine Einführung	Abgrenzung gegenüber anderen Disziplinen
1979	Gerhard GOOS	[Goos 1979]	Editorial IF-Spektrum	konstruktives Element als Begründung für die Schulinformatik
1982	Wolfgang ARLT	[Arlt 1982]	Kulturtechnik	Informatik für Lehrer – zweibändiges Werk
1985	David L. PARNAS	[Parnas 1986]	Fehlerhafte Software	Grenzen der Software-Engineering-Methoden
1986	Fredrick P. BROOKS	[Brooks 1986]	Eigenschaften von Software	Software ist unsichtbar und kann nicht veranschaulicht werden
	Peter C. LOCKEMANN	[Lockemann 1986]	Fundamentale Ideen	Datenbanktechnik: Konsistenz, Konkurrenz, Persistenz
1990	Joseph WEIZENBAUM	[Weizenbaum 1990]	Wirkprinzipien – wie tief?	Große Unsicherheit, was für die allgemeine Bildung Informatik nötig ist
1991	Jörg NIEVERGELT	[Nievergelt 1991]	Kritik an „Sammlung von Ideen“	Forderung nach Strukturüberlegungen und -entscheidungen
1992	Ute und Wilfried BRAUER	[Brauer und Brauer 1992]	Paradigmenwechsel	5 × „C“
	Mary SHAW	[Shaw 1992]	Kritik an den Inhalten der Lehre	Weg von den Anwendungen – hin zu den Prinzipien
	Joseph WEIZENBAUM	[Weizenbaum 1992]	Modelle	Forderung, dass Lehrerinnen etwas über Modellbildung wissen müssen.
1994	Christiane FLOYD	[Floyd 1994]	Softwareentwicklung – Prototyping	Exploration, Experimente, Kommunikation unter den Beteiligten ist ein Lernprozess und muss (technisch?) unterstützt werden
1999	Niklaus WIRTH	[Wirth 1999]	Ingenieurmässiges Erstellen abstrakter Maschinen	In der Konsequenz muss die Implementierung auch im Detail verstanden werden
2002	Uwe SCHÖNING	[Schöning 2002]	Informatisches Interesse	Es geht um die Sichtweise – diese macht die Informatik aus

Tabelle 4.2: Empfehlungen der Fachwissenschaft zur Didaktik – Quellen und Einordnung

Seit einigen Jahren findet in der fachdidaktischen Forschung eine verstärkte Auseinandersetzung mit der informatischen Modellierung statt. Erste Bestätigungen des allgemein bildenden Charakters informatischer Modellierung werden im [folgenden] Abschnitt 4.2 dargestellt.

Zu 3.: Für diese Arbeitshypothese wurden Belege in den vorgestellten Positionen dargestellt. Es bleibt zu diskutieren, welche Elemente für den Bildungsprozess konkret ausgestaltet werden können, um dieser weitgehenden Position Rechnung zu tragen.

## 4.2 Didaktik der Informatik für Schulen

### 4.2.1 Entwicklung in der Bundesrepublik Deutschland

Um für die Bundesrepublik vereinheitlichende Bildungsplanungsmaßnahmen einzuleiten, die den [Bundes-]Ländern ein gewisses Maß an Gestaltungsmöglichkeiten eröffnet, entwickelt die Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung (BLK) Rahmenkonzepte. Die Kultusministerkonferenz (KMK) hingegen legt durch ihre Vereinbarungen normierende und damit vergleichbare Abschlussbedingungen für die Sekundarstufe I und das Abitur fest. In der Vereinbarung zur Gestaltung der gymnasialen Oberstufe in der Sekundarstufe II der KMK wird unter dem Punkt Zielsetzung festgelegt, dass in Grundkursen wissenschaftlich orientierte Arbeitsweisen zu berücksichtigen sind:

Die in der gymnasialen Oberstufe zu erwerbenden Kenntnisse, Methoden, Lernstrategien und Einstellungen werden über eine fachlich fundierte, vertiefte allgemeine und wissenschaftspropädeutische Bildung und eine an den Werten des Grundgesetzes und der Länderverfassungen orientierte Erziehung vermittelt, die zur Persönlichkeitsentwicklung und -stärkung, zur Gestaltung des eigenen Lebens in sozialer Verantwortung sowie zur Mitwirkung in der demokratischen Gesellschaft befähigen.

[Kultusministerkonferenz 1999, S. 4, 5]

Wissenschaftspropädeutische Bildung fordert den Einsatz fachspezifischer Methoden, die exemplarisch eingeführt werden. Deshalb ist es notwendig, fundierte Studien zu dem Beitrag des Schulfachs Informatik zur wissenschaftspropädeutischen Bildung durchzuführen.<sup>179</sup> Damit sollen die Möglichkeiten zur Umsetzung dieser Anforderungen für das Schulfach Informatik untersucht werden.

Der Autor stützt sich in seinen Untersuchungen auf allgemein verwendete curriculare Prinzipien wie spiralem Aufbau (Spiralprinzip, vgl. Abschnitt 3.1, S. 38 und 3.3, S. 51) und didaktische Reduktion. Obwohl das Spiralprinzip allgemein anerkannt ist, kann die Lehrerin bei der Umsetzung in Konflikt geraten mit der Binnenstruktur des Lerngegenstandes und mit einer scheinbar zwingenden Fachsystematik. Der Autor ist sich bewusst, dass es für jede Lerngruppe eine Vielzahl guter Varianten zur Umsetzung spiraler Curricula gibt; jedoch fehlt im Bereich der Informatischen Bildung eine ausreichende Anzahl empirischer Vergleiche, die die Bewertung solcher Varianten erleichtert. Deshalb führte der Autor exemplarisch empirische Studien durch, die in Kapitel 6 zu dem Modulkonzept verdichtet werden (Beispiele siehe Anhang F).

Didaktische Reduktion bezeichnet die Identifizierung der „elementaren Ideen“ eines Sachgegenstandes und die lerngruppenbezogene „Vereinfachung“, ohne in die Gefahr zu geraten, eine fachlich falsche Vorstellung zu entwickeln (vgl. [Häußler u. a. 1998, S. 201f]). Um einer möglichen falschen Interpretation<sup>180</sup> vorzubeugen, hält der Autor den Begriff „Didaktische Gestaltung“ für geeigneter, diesen Prozess zu bezeichnen.<sup>181</sup>

<sup>179</sup> Solche Studien existieren in unzureichendem Maße.

<sup>180</sup> Reduktion wird im allgemeinen Sprachgebrauch auch zur Bezeichnung für „unzulässige Vereinfachung“ benutzt und ist damit negativ besetzt.

<sup>181</sup> Der Begriff „Didaktische Gestaltung“ wird bisher in der Didaktik nicht in Abgrenzung zur „Didaktischen Reduktion“ verwendet. Allerdings wird er in einigen Veröffentlichungen benutzt, so z. B. im Kontext der Gestaltung von Lehr-/Lernmaterialien, die didaktischen Anforderungen genügen (sollen).

In der Bundesrepublik Deutschland bieten die Fachtagungen „Informatik und Schule“ (ab 1995 mit dem Kürzel INFOS), die seit 1984 in der Regel alle zwei Jahre stattfindet, Gelegenheit, die Entwicklung der Didaktik der Informatik für Schulen zu identifizieren. Aus diesem Grund werden Beiträge ausgewählt, die veränderte konzeptionelle Orientierungen deutlich machen. In die Diskussion werden darüber hinaus bedeutsame Ergebnisse einbezogen, die nicht im (direkten) Zusammenhang mit den Fachtagungen veröffentlicht wurden.

Im Erscheinungsjahr relativ unbeachtet, aber rückblickend sehr bedeutend ist die Dissertationsschrift von Petra KNÖSS „Fundamentale Ideen der Informatik im Mathematikunterricht“ [Knöß 1989]. Diese Studie begründet die informatische Allgemeinbildung aus der mathematischen Bildung im Primarbereich. Obwohl dieser Ansatz kritisch zu sehen ist, bildet er jedoch die Anregung für weitergehende Forschungen (z. B. [Schwill 1993]). KNÖSS meint, dass der Mathematikunterricht in der Primarstufe die Darstellung, Realisierung und Qualität von Algorithmen und Datenstrukturen den Lernenden nahebringt. Die Verkürzung fundamentaler Ideen der Informatik auf Algorithmen und Datenstrukturen durch KNÖSS bleibt hinter den Bildungsanforderungen der Schulinformatik in dieser Zeit zurück.

1989

Es bedurfte einiger Zeit der Praxis des Informatikunterrichts, bis ein Zielkonflikt erkannt wurde. Viele Lehrerinnen gestalteten ihren Unterricht nach den Empfehlungen der GI von 1976<sup>182</sup>, indem sie am Beispiel einer prozeduralen Programmiersprache Algorithmen und Datenstrukturen thematisierten. Gleichzeitig verloren die befehlsorientierten Unterrichtsbeispiele immer mehr an an motivierender Bedeutung im Vergleich zu Informatikprodukten, die Schülerinnen im Alltag begegneten. Deshalb wird der Vortrag von PESCHKE im nachhinein als bedeutender Versuch einer Neuorientierung der Informatischen Bildung gesehen, der aber 1989 heftig umstritten war.

„Probleme wie vernetzte Gesellschaft und informationelle Selbstbestimmung, die automatisierte Fabrik und soziale Gestaltbarkeit, fehlerhafte informationstechnische Systeme und Verantwortlichkeit können über den algorithmischen Zugang nicht befriedigend behandelt werden“ [Peschke 1989, S. 91ff].

PESCHKE berücksichtigt Ergebnisse der wissenschaftstheoretischen Fundierung der Informatik (vgl. Abschnitt 2.4), wie in dem Beitrag ausgewiesen wird. Aus heutiger Sicht treten sehr ungünstige Lösungsversuche ein, da PESCHKE sich gegen die Stärkung des Unterrichtsfaches Informatik ausspricht, anstatt eine Veränderung des Schulfaches zu fordern. So wurde in der Folge z. B. einer Pseudo-Integration informatischer Fachinhalte über Projekte, die an ein Leitfach gebunden wurden, propagiert und führten zu einer deutlichen Verschlechterung der Stellung und der Notwendigkeit des Schulfachs Informatik. Die Didaktik der Informatik war zu diesem Zeitpunkt noch kein etabliertes Teilgebiet der Informatik. Es wurden aber für die wenigen Standorte der Lehrerbildung Informatik bereits Lehrveranstaltungen angeboten, die von Forscherinnen des computerunterstützten Unterrichts (CUU) getragen wurden. Einer engagierten Lehrerschaft und der Fachzeitschrift LOG IN, die immer um gute Beispiele für den Informatikunterricht bemüht war, gelang es, den Fortbestand der Schulinformatik zu sichern. Die Didaktik der Informatik existierte so in Form dokumentierter Unterrichtserfahrungen. Es fehlten ihr zu diesem Zeitpunkt jedoch tragfähige Konzepte, die dem berechtigten Anspruch von PESCHKE gerecht wurden.

Sigrid SCHUBERT fordert die Einbeziehung von theoretischen Elementen in das Schulfach Informatik. Sie macht deutlich, dass der Nichtdeterminismus der Wissenschaft Informatik bislang unterrichtlich kaum berücksichtigt wird, obwohl ihm für die Weiterentwicklung der Informatik in der Schule eine wichtige Rolle zufallen kann. Des weiteren gibt sie an, dass für das Schulfach Informatik geeignete Systemumgebungen notwendig sind, die als Basis für die Erarbeitung inhaltlicher Fragen nützlich sind (vgl. [Schubert 1991]). Interessant ist hierbei der Vergleich mit der im gleichen Jahr veröffentlichten Stellungnahme von NIEVERGELT, der als Fachwissenschaftler argumentierend ähnliche Schlussfolgerungen zieht (vgl. Abschnitt 4.1 – 1991, S. 58). Sowohl NIEVERGELT wie auch SCHUBERT begründen Vorgehensweisen, die besonders für die informatikdidaktische Ausbildung von Lehramtsstudierenden angewendet werden können. Damit wird die Didaktik der Informatik für Schulen als verbindendes Element von Hochschule und Unterrichtspraxis stärker sichtbar.

1991

<sup>182</sup> siehe [Brauer u. a. 1976] – Abschnitt 4.1, S. 55

1993

Aus der Untersuchung der Bereiche der Kerninformatik leitet Andreas SCHWILL drei fundamentale Ideen der Informatik für die allgemeine Bildung ab:

- Algorithmisierung,
- strukturierte Zerlegung und
- Sprache.

[Schwill 1993, S. 25ff]

In Baumstrukturen werden diese drei fundamentalen Ideen mit weiteren fundamentalen Ideen ausdifferenziert. In den Studien findet sich keine Diskussion von Ideen, die zwar geprüft, dann aber verworfen wurden. Daraus läßt sich schließen, dass die 55 fundamentalen Ideen eher eine pragmatisch zusammengestellte Auflistung darstellen. Die Vermutung, dass der subjektive Erfahrungshintergrund des Forschers die Begründungsintensität stark beeinflusste, liegt nahe. Der weitergehende Anspruch einer „vollständigen Kollektion aller fundamentalen Ideen der Wissenschaft“ [Schwill 1993, S. 23] wird mit einer theoretischen Methodik verbunden, die fragwürdig erscheint. Es soll „von den Inhalten einer Wissenschaft zu ihren Ideen abstrahiert“ [Schwill 1993, S. 23] werden, ohne zu beachten, dass dieser Auswahlprozess einen historischen Kontext besitzt; also höchstens eine Momentaufnahme der Ideenkollektion zu einem konkreten Zeitpunkt liefert.

Der Autor sieht in der fundamentalen Idee „Teamarbeit“, die der strukturierten Zerlegung und dort dem Zweig der Modularisierung zugeordnet wurde, ein Problem und eine Chance dieses theoretischen Ansatzes zugleich. Die Chance liegt darin, dass die fundamentalen Ideen der nicht formalisierbaren Bereiche der Informatik in das Bildungskonzept einbezogen werden können. Das Problem besteht darin, dass die partizipative Softwareentwicklung mit starker Betonung der Kommunikationsprozesse nicht das letzte Blatt eines Baums der fundamentalen Ideen sein kann, denn Softwareentwicklung ist ein kooperativer Prozess zwischen Menschen, der sich nicht allein mit Algorithmisierung und strukturierter Zerlegung erklären läßt.

Dieser – aus Sicht des Autors – besonders bedeutsame Beitrag zur Fundierung der Didaktik der Informatik führte jedoch zu einer Polarisierung in der Lehrerschaft, da theoretische Informatik in den fundamentalen Ideen überproportional häufig vertreten ist – Elemente der technischen Informatik hingegen (fast) völlig fehlen. Schwerwiegender erscheint dem Autor der fehlende Zugang zu soziotechnischen Systemen und den damit verbundenen Anwendungsbereichen.

Die Stärke der Konzepts fundamentaler Ideen für das Schulfach Informatik besteht in der fachlichen Absicherung von Inhalten. Dies setzt voraus, dass vorher entschieden wird, welche Inhalte curricular umgesetzt werden sollen. Informatik in der Schule kann nicht allein über die Fachsystematik begründet werden, sondern benötigt einen allgemeinpädagogischen Kontext. Das heißt, Inhalte sollten sich aus der Lebenswelt der Schüler ergeben und müssen als epochale Schlüsselfragen [Klafki 1985a] im Informatikzusammenhang thematisiert werden.<sup>183</sup>

Rüdeger BAUMANN legt ein Konzept zur Ausgestaltung des Unterrichts für die Sekundarstufe II vor, in dem er die „zeitgemäße didaktische Weiterentwicklung des Informatikunterrichts“ [Baumann 1993, S. 12] in Form eines systemorientierten Ansatzes fordert: „Damit ist informatisches Problemlösen als Systementwicklung (Entwicklung eines Informatiksystems) charakterisiert, wobei [...] die [...] Informatikmethoden ihre zentrale Stellung behalten. Ferner wird der Computer nicht als isoliertes Einzelgerät, sondern als Teil umfassender Systeme gesehen.“ Der Ansatz wird von BAUMANN „als Synthese der bisherigen Ansätze (einschließlich der Vorgeschichte)“ charakterisiert [Baumann 1993, Zitatenskollage, S. 12].

Ausgehend von drei Leitfragen, entwickelt er jeweils ein fachliches Richtziel:

- A) Wie werden Informatiksysteme entworfen, programmiert und damit zum Lösen lebensweltlicher Probleme befähigt?

Problemlösen als methodischer Entwurf von Informatiksystemen.

<sup>183</sup> Eine weit über die hier formulierte Kritik hinausreichende Auseinandersetzung mit den „Fundamentalen Ideen der Informatik“ wird 1998 von Rüdeger BAUMANN formuliert (vgl. [Baumann 1998]). Andererseits geht Baumann hin und benennt die fundamentalen Ideen: *Formalisierung*, *Algorithmisierung* und *Vernetzung*. Er weist darauf hin, dass der Informatikunterricht sich bisher auf die ersten beiden stützt(e) und die Zeit für die dritte Idee reif sei (vgl. [Baumann 1998, S. 104]).

- B)** Wie sind Informatiksysteme aufgebaut, wie wirken ihre Komponenten zusammen und wie ordnen sie sich in umfassendere soziotechnische Systemzusammenhänge ein?

Struktur und Funktion von Informatiksystemen im soziotechnischen Kontext.

- C)** Wo liegen die prinzipiellen Grenzen technischer Informationsverarbeitung, und was ist unter Information und Kommunikation überhaupt zu verstehen?

Prinzipielle Grenzen technischer Informationsverarbeitung sowie grundlegende Konzepte von Information und Kommunikation.

[Baumann 1993, S. 12f]

BAUMANN liefert damit auf dem Hintergrund des Bildungsbegriffs von KLAFKI die Legitimation für Informatik als eigenständiges Schulfach des Sekundarbereichs II. In der Ausformung orientiert er sich sehr klar an dem Nutzen theoretischer Begriffe als konstitutives Element des Bildungsgehalts des Faches (vgl. [Baumann 1993, z. B. S. 13]).

Die GI legt „Empfehlungen zur Gestaltung des Informatikunterrichts für die Sekundarstufe II“ vor. Damit wird eine Tradition fortgesetzt, die 1976 mit der Veröffentlichung der „Zielsetzungen und Lerninhalte des Informatikunterrichts“ [Brauer u. a. 1976] einen Meilenstein für die Entwicklung der Informatik in der Schule setzte. Es wird herausgestellt, dass „Sichtweisen benötigt [werden], die die Entwicklung der Informatik aufnehmen und den [...] Anforderungen der Allgemeinbildung genügen“ [Schulz-Zander u. a. 1993, S. 206]. In den Ausführungen werden Ziele und Inhalte des Informatikunterrichts angegeben. Die vorgeschlagene Struktur umfasst drei Sichtweisen. Im Folgenden werden die für den thematischen Zusammenhang zentralen Elemente der Gliederung aufgeführt.

- A)** Mensch-Computer  
Exemplarisch: Methoden und Verfahren der Modellierung, Möglichkeiten der Gestaltung von Benutzungsschnittstellen und Software-Entwicklungsumgebungen, Möglichkeiten von Teamorganisation und Selbstorganisation
- B)** Formalisierung und Automatisierung geistiger Arbeit  
theoretische Fundierung der Informatik: einfache Automatenmodelle und zugehörige Grammatiken, Wirkprinzipien und die Architektur von Informatiksystemen
- C)** Informatiksysteme, Gesellschaft und Umwelt  
fundierte Einsichten in Chancen, Risiken und Grenzen eines verantwortbaren Einsatzes von Informatiksystemen, das Grundrecht auf informationelle Selbstbestimmung, Beurteilungsfähigkeit von Auswirkungen der Anwendungen der Informatiksysteme auf die Lebens- und Arbeitswelt, auf Politik und Umwelt sowie die kulturelle, historische und anthropologische Dimension des Einsatzes von Informatiksystemen  
(vgl. [Schulz-Zander u. a. 1993, S. 208ff]).

Den beiden eher theoretisch orientierten Veröffentlichungen von SCHWILL und BAUMANN stehen die notwendigerweise pragmatisch orientierten Empfehlungen der GI von 1993 gegenüber. Die Stärken dieser Empfehlungen bestehen darin, dass gegenüber vorherigen Empfehlungen deutlich gemacht wird, dass „eine exemplarische Behandlung von Themen im Unterricht [vorzusehen ist], die alle Sichtweisen abdeckt. [...] Bei der Gestaltung des Unterrichts ist anzustreben, die verschiedenen Sichtweisen zu integrieren“ [Schulz-Zander u. a. 1993, S. 208].

Damit wird der Zergliederung der Inhalte und der Reduzierung auf Einzelaspekte eine Absage erteilt. Überdauernde Grundkonzepte (= Bewährtes) werden benannt, z. B. „Entwicklung von Algorithmen und Datenstrukturen, [...] einfache Automatenmodelle und zugehörige Grammatiken“, es wird deutlich auf verschiedene Paradigmen hingewiesen, und gefordert, dass sie ihren Niederschlag im Unterricht finden sollen.

Die vorgelegten Empfehlungen machen deutlich, dass die GI die kontinuierliche Weiterentwicklung der Informatik in der Schule befürwortet.

1995

Steffen FRIEDRICH hält die Diskussion über den allgemeinbildenden Charakter der Informatik für abgeschlossen und fokussiert auf folgende Grundprinzipien:

- Grundprinzipien des Aufbaus, des Funktionierens und der Wirkung von Maschinen zur Verarbeitung von Informationen;
- Strategien der Problemlösungen mit Werkzeugen der Informatik, Betrachtungen zu deren Möglichkeiten und Grenzen;
- Gesichtspunkte und Konsequenzen der Abstraktion und Modellbildung mit Mitteln der Informatik.

[Friedrich 1995, S. 38]

Er fordert die Fachdidaktik auf, die Ausgestaltung des Schulfachs Informatik im Ensemble der anderen Fächer in der Schule – sowohl bezogen auf die Inhalte (hier vor allem Begriffsbildung), wie auch auf die spezifischen informatischen Arbeitsmethoden – zu konturieren.<sup>184</sup>

Bei den von FRIEDRICH formulierten Anforderungen wird z. B. nicht spezifiziert, welche Modellbildung der Informatik für die Schule gewählt werden soll.

Dieter ENGBRING assoziiert mit dem Begriff „Technik“ die Gemeinsamkeit aller vorliegenden Inhalts- und Begründungsfragmente zum Schulfach Informatik.

Informatik kann als Teil technischer Bildung einen Beitrag zur Erklärung technischer Phänomene sowie zur Einschätzung, Beurteilung und Bewertung von Technik leisten, so wie die Naturwissenschaften in den Schulen einen Beitrag zur Erklärung von Phänomenen der Natur leisten. Dieser letztgenannte Aspekt technischer Bildung ist in der gegenwärtigen Situation für das gesellschaftliche Zusammenleben und die Entmystifizierung von Technik, speziell von Computern, genauso wichtig oder gar wichtiger, als es heute die Naturwissenschaften für die Entmystifizierung von Naturphänomenen noch sind.

[Engbring 1995, S. 76]

Diese Position ist umstritten, die Chance besteht darin, dass mit dem Schulfach Informatik in der Schule der Themenbereich „Technik“ und „Gestaltung der Technik“ einen den gesellschaftlichen Anforderungen angemessenen Raum erhält. Andererseits kann in einem neu aufzurichtenden allgemein bildenden Fach Informatik nicht bereits zu Beginn erwartet werden, dass ein allgemeiner Konsens über die Basis, die das Schulfach konstituiert, erreicht wurde.

SCHUBERT und SCHWILL dokumentieren Ergebnisse einer fachdidaktischen Arbeitsgruppe zur „Strukturierung der Schulfachs Informatik aus der Sicht der Fachwissenschaft“. Die Arbeitsgruppe stellt fest, dass das Konzept der fundamentalen Ideen allein nicht ausreicht, um zu einer Systematik zu gelangen, weil wissenschaftstheoretische Sichtweisen der Informatik (Woher kommt, was ist, wohin geht die Informatik?) und Fragen nach dem Beitrag des Schulfachs Informatik zur Allgemeinbildung nur unzureichend berücksichtigt werden (vgl. [Schubert und Schwill 1996]).

1997

Sybille KRÄMER stellt mit ihren Thesen über den Umgang mit dem technischen Artefakt Computer die Zeitgebundenheit der Inhalte und den damit verbundenen Perspektivenwechsel heraus: „Damit wird, was als eine Denkleistung zählt, abhängig von den sich historisch wandelnden kulturellen Praktiken unseres Zeichengebrauches. [...] Was der Computer leistet, hat [...] mit der Automatisierung von Symboloperationen, die wir immer schon extern vollzogen haben [zu tun]. Der Computer zeigt sich [...] als der apparative Vollzug von Kulturtechniken, die auf dem Einsatz von »symbolischen Maschinen« beruhen“ [Krämer

<sup>184</sup> Der Wunsch, die „endlose Debatte“ (vgl. den Titel [Friedrich 1998]) um die Schul informatik zu beenden, mag subjektiv verständlich sein, ist aber auf dem Hintergrund der Weiterentwicklung der Bezugswissenschaft (vgl. Kapitel 2) als verfehlt einzustufen. Die Feststellung, dass Informatik allgemein bildend ist, wird damit aktuell keineswegs in Frage gestellt. Dennoch muss der immer wieder konkret diskutiert werden, um Fehlentwicklungen vorzubeugen.

1997, S. 8]. Sie fügt „[der kanonischen] Unterscheidung von instrumentellem und kommunikativem Handeln“<sup>185</sup> eine Überlegung für die Möglichkeit einer dritte Modalität unseres Handelns hinzu: die „spielerische Interaktion“ (vgl. [Krämer 1997, S. 12f]). Daraus ergibt für sich abzeichnende Veränderungen die Fragestellung: „[Kann die instrumentelle Perspektive] erschließen, worin dessen [<sup>186</sup>] zukunftssträchtiges Potential liegen kann?“ [Krämer 1997, S. 7]

Diese Frage kann nach Meinung des Autors am Ende des Artikels allgemeiner formuliert werden, indem sie um die beiden von KRÄMER zusätzlich herangezogenen Perspektiven Denkzeug, Spielzeug erweitert wird.

Für den hier betrachteten Zusammenhang ist es wichtig, festzuhalten, dass die verschiedenen Perspektiven zwar historisch in Folge herausgearbeitet wurden, aber keine als ausschließlich gültige Perspektive angesehen werden kann. Jede der Perspektiven hat ihren Gültigkeitsbereich und läßt unterschiedliche Entwicklungsmöglichkeiten erkennen. Der Nutzung dieser Erkenntnisse zur Gestaltung und Umsetzung curricularer Elemente Informatischer Bildung wurde bisher nicht Rechnung getragen. Sie verweisen auf die Möglichkeit, über die bisherigen Zieldimensionen hinausreichende Ansätze erfolgreich zu begründen.

Peter HUBWIESER und Manfred BROY rücken in [Hubwieser und Broy 1997] als Ziel des Informatikunterrichts „Fähigkeiten zum Umgang mit Information“ in den Mittelpunkt ihrer Betrachtungen. Dies soll umgesetzt werden, indem Schülerinnen Vorstellungen angeboten werden „[...] um] die charakteristischen Eigenschaften von Informations- und Kommunikationssystemen [...] zu begreifen und [...] zu beherrschen“ [Hubwieser und Broy 1997, S. 41]. Die Modellierung wird als Unterrichtsprinzip benannt und in das Zentrum der konkreten Arbeit der Informatik als Schulfach gerückt. Im Zusammenhang mit methodischen Vorschlägen wird herausgearbeitet, dass als Ausgangspunkt eines so orientierten Unterrichts Problemstellungen aus der Praxis zur Problemgewinnung gewählt werden sollten, die – in projektorientierte Unterrichtssequenzen heruntergebrochen – die Phasierung des Modellbildungs- und Simulationsprozesses im Unterrichtsprozess widerspiegeln (vgl. [Hubwieser und Broy 1997, S. 42f]).

HUBWIESER beschreibt zusammenfassend die Phasierung einer so verstandenen Unterrichtsgestaltung:

- Problemgewinnung,
- informelle Problembeschreibung,
- Formale Modellierung,
- Realisierung von Lösungsansätzen/Simulation und
- Bewertung.

(vgl. [Hubwieser 1997, S. 216])

Mit diesem Beitrag wird eine theoriegeleitete Klammer zum Schulfach Informatik vorgelegt. Der Ansatz wird auch als informationszentrierter Ansatz bezeichnet (vgl. [Hubwieser und Broy 1996]). Durch diesen Vorschlag wird die Modellierung im Informatikunterricht in das Zentrum der didaktischen Überlegungen gerückt. Allerdings wird die Realisierung nicht „unter den Tisch gekehrt“, sondern ist, wie in der Phasierung bei HUBWIESER deutlich wird, Bestandteil der Unterrichtsgestaltung.

Helmut WITTEN und Johann PENON dokumentieren und extrapolieren Erfahrungen, die sie im Zusammenhang mit der Nutzung elaborierter Informatiksysteme im Informatikunterricht und als Grundlage zur schulweiten Präsentation von Ergebnissen gemacht haben. Dabei stellen sie ihre Unterrichtserfahrungen mit vernetzten Systemen als Unterrichtsgegenstand vor. In der Reflexion eines durchgeführten Kurses „Telekommunikation“<sup>187</sup> weisen sie auf Perspektiven für den Informatikunterricht hin: „Ein nächster Schritt wäre es, Teile des Informatik-Unterrichts, z. B. Softwareprojekte, mit Hilfe von Workgroup-Software zu

<sup>185</sup> „[als] Differenz zwischen einem Tun, das erfolgsorientiert ist und auf die technikgestützte Leistungssteigerung im Umgang mit Sachen abzielt und einem Tun, das verständigungsorientiert ist und die sprachvermittelte Anerkennung anderer Personen impliziert.“ [Krämer 1997, S. 12]

<sup>186</sup> gemeint ist die Nutzung des Computers

<sup>187</sup> Vertiefungsgebiet im 4. Kurshalbjahr – lt. Berliner Rahmenplan können hier auf Antrag neue Themen erprobt werden [Witten und Penon 1997, S. 172]

realisieren. Hier bietet sich das für Schulen frei erhältliche Programm BSCW der GMD <sup>[188]</sup> an. Dieses Programm ermöglicht für einen geschlossenen Benutzerkreis den gezielten Austausch von Dokumenten und deren Verwaltung, eine fortlaufende ergebnisorientierte Problemdiskussion und vieles mehr“ [Witten und Penon 1997, S. 173]. Außerdem betonen sie, dass Fragen der Codierung, der Computersicherheit und der Kryptologie im Zusammenhang mit dieser Unterrichtsgestaltung eine neue Bedeutung erlangen, die auch geschichtlich interessant sei (vgl. [Witten und Penon 1997, S. 173]).

Der Beitrag macht deutlich, dass die Orientierung des Informatikunterrichts an aktuellen Problemstellungen dazu führen kann, dass bereits „überholt“ geglaubte Fragestellungen der Informatik in der Schule unter einer neuen didaktischen Perspektive Bedeutung erlangen. Auch Bereiche, die bereits seit langem für wichtig erachtet werden, aber der unterrichtlichen Umsetzung nur schwer zugänglich sind, sollten (zumindest versuchsweise) umgesetzt werden, damit Erfahrungen mit der didaktischen Gestaltung informatischer Inhalte gesammelt werden können, denen ggf. zukünftig eine wichtige Rolle zukommt.

Peter BERGER dokumentiert die Untersuchung des Computer-Weltbildes von Informatiklehrkräften an Schulen<sup>189</sup>. Die in der Studie angewandte Forschungsmethodik basiert auf Verfahren der qualitativen Sozialforschung. Die dokumentierten Ergebnisse der Untersuchungen werden im Folgenden zusammengefasst. „Insbesondere bei den akademisch ausgebildeten Informatiklehrern ist [...] eine Hinwendung zu dem als fundamental eingeschätzten Konzept der Algorithmik [festzustellen]. [...] Sie neigen [...] zu einer Betonung des Grundlegenden und Bleibenden [...]. Das Lernen lokaler Taktiken [...] tritt zurück zugunsten des Erwerbs globaler Strategien (Denken in Prozessen und Systemen)“ [Berger 1997, S. 30]. Zum Themenfeld 'Informatik als Wissenschaft' führt BERGER aus: „Standardisiert man die individuellen Beschreibungen, so läßt sich die überwiegende Mehrheit [...] den vier Hauptkategorien Informationswissenschaft, Strukturwissenschaft [...], Computerwissenschaft und Algorithmentheorie zuordnen“ [Berger 1997, S. 29f].

Ergebnisse zu dem ebenfalls untersuchten Themenbereich „Bild vom Lehren und Lernen von Informatik“ [Berger 1997, S. 33ff] stellt BERGER folgendermaßen dar: „Das traditionelle Paradigma *Schule* – charakterisiert durch Schlüsselbegriffe wie Unterricht, Hausaufgabe, Klassenarbeit, lehren, erziehen, prüfen, benoten etc. – wird wenn auch nicht geradezu verdrängt, so doch zunehmend ergänzt und überlagert von einem neuen Paradigma *Berufswelt* mit den Leitkonzepten Projekt, Produkt, Team, Diskussion, beraten, delegieren, mitbestimmen und kooperieren [...] Allerdings ist dieser Wechsel zumeist nicht als Folge bewußter didaktischer Innovation und zielgerichteter Entwicklung eines neuen Unterrichtsstils zu erklären. [...] Prononciert könnte man formulieren: Im innovativen Schulfach Informatik findet Innovation zur Zeit weniger von *innen* statt, durch den innovativen Lehrer, der ein neues Paradigma des Lehrens und Lernens findet – als vielmehr von *außen*, durch ein neues Paradigma, das 'seinen Lehrer findet' und ihn, auch den durchaus traditionell eingestellten, zunehmend zu innovativen Mustern greifen läßt“ [Berger 1997, S. 38].

Anhand dieser Ergebnisse wird deutlich, dass die Veränderung der Gewichtung der Kernbereiche und der zentralen Konzepte des Informatikunterrichts über die Zeit eine zentrale Größe in der Selbsteinschätzung der Informatiklehrerinnen darstellt. Nicht unterschätzt werden sollten die Aussagen bzgl. der Unterrichtskultur und damit der didaktisch-methodischen Orientierung des Informatikunterrichts. Für die Didaktik der Informatik bedeutsam erscheint die Feststellung eines Paradigmenwechsels hin zu modernen Leitkonzepten, die mit innovativen Mustern unterrichtlich umgesetzt werden. Damit kann das Schulfach Informatik zum Leitfach für notwendige Umgestaltungen der schulischen Praxis werden.

Dabei sollten nicht die Fehler wiederholt werden, die im Zusammenhang mit dem Versuch der Einführung der informationstechnischen Grundbildung (ITG) in bundesdeutschen Schulen<sup>190</sup> gemacht wurde. Eine Ursache für das Scheitern der ITG liegt in der mangelnden Qualifikation der Lehrerinnen für die informatischen Grundlagen – eine zweite darin, dass die ITG so gestaltet wurde, dass der notwendigen Erarbeitung der Grundlagen in der Umsetzung kein Platz eingeräumt wurde. Darüber hinaus ist festzustellen, dass mit der Einführung der ITG die Hoffnung auf eine erhebliche Ausweitung des projektorientierten Unterrichts verbunden wurde, da nur so die gesteckten Ziele eingelöst werden könnten. Diese Erwartung konnte u. a. aus den genannten Gründen nicht erfüllt werden (vgl. [Altermann-Köster u. a. 1990]).

<sup>188</sup> GMD – Gesellschaft für Mathematik und Datenverarbeitung mbH

<sup>189</sup> im Bundesland Nordrhein-Westfalen

<sup>190</sup> Die Einführung der ITG in der Schule wird inzwischen als gescheitert eingeschätzt (exemplarisch [Wilkens 2000]).

Johannes MAGENHEIM, Carsten SCHULTE und Thorsten HAMPEL dokumentieren einen fachdidaktischen Ansatz zur objektorientierten Modellierung unter Bezugnahme auf die Konzepte Dekonstruktion (als Möglichkeit zur Durchdringung) und Konstruktion (als Erweiterung) existierender Informatiksysteme. Damit soll „Schülerinnen und Schülern die Einsicht in Verfahren der objektorientierten Modellierung sowie in relevante Methoden der Gestaltung und Bewertung von Informatiksystemen gegeben werden“ [Magenheim u. a. 1999, S. 149]. Als Bezug wird die systemorientierte Didaktik der Informatik ausgewiesen. Projektorientierter Unterricht wird für notwendig erachtet, und in der Konkretion und Ausgestaltung die objektorientierte Modellierung favorisiert.

1999

Das Konzept lässt zu diesem Zeitpunkt allerdings Fragen nach der Konkretisierung der Umsetzung unbeantwortet. Es ist deutlich, dass der Arbeitsgruppe MAGENHEIM, SCHULTE u. a. eine nicht zu unterschätzende Arbeit bevorsteht. Ein zentrales Problem besteht darin, eine notwendige Komplexität der zu dekonstruierenden Systeme mit einer Erweiterungsmöglichkeit so zu verbinden, dass sie handhabbar werden und offen bleiben. Bezogen auf die beschriebene, angestrebte unterrichtliche Umsetzung stellt die Orientierung des Ansatzes auf die Softwaretechnik ein weiteres Problem dar. So werden die Elemente der technischen Informatik und damit ein wesentlicher Teil der Wirkprinzipien von Informatiksystemen bei dem Konzept noch nicht in den Blick genommen.

Ludger HUMBERT unternimmt in seinem Beitrag den Versuch, eine Bestandsaufnahme des Faches Informatik im allgemeinen und in der Schule, speziell der gymnasialen Oberstufe, zu leisten. Sie soll dem langfristigen Ziel dienen, einen Beitrag zur Entwicklung eines verpflichtenden und aufeinander abgestimmten Curriculums für den Informatikunterricht in den Sekundarstufen I und II zu leisten. Es wird ein Überblick über didaktische Zugänge zur Erarbeitung informatischer Inhalte im Unterricht gegeben und eine Beziehung zu fachwissenschaftlichen Sichtweisen der jeweiligen Thematik hergestellt. Als mögliche zentrale Gegenstände einer Theorie der Informatik werden Wirkprinzipien und die Verantwortung der Informatiker, der technische Umgang mit Wissen bzw. Information, Sprachen zur Codierung von Entwürfen, der Computer als symbolverarbeitende Maschine sowie Sinn und Zweck der Informatik vorgestellt. HUMBERT umreißt mit dieser komprimierten Beschreibung eines komplexen Sachverhalts den Kontext, innerhalb dessen sich eine curriculare Fundierung der Informatik in der Schule vollziehen sollte. Er klassifiziert die vorgestellten Beispiele nach den Kategorien Zugang, Modellierungsmethode und Sprache. Hinsichtlich der Kategorie Zugang werden die vorgestellten Beispiele den didaktischen Konzepten eines anwendungs- bzw. gesellschaftsorientierten Ansatzes zugeordnet. In den beschriebenen Beispielen werden imperative, objektorientierte und prädikative Modellierungsmethoden verwendet. Es werden Vorüberlegungen zu einem Spiralcurriculum des Informatikunterrichts präsentiert, der bereits in der Sekundarstufe I zu einer fundierten informatischen Grundausbildung führen soll. Diese Grundausbildung soll die persönliche Erstbegegnung der Schüler mit dem Informatikunterricht positiv besetzen. Als Leitlinien eines solchen Spiralcurriculums werden u. a. benannt: Informatische Grundlagen der Nutzung informationstechnisch unterstützter Gruppenarbeit, Ablösung der strukturierten Programmierung durch die objektorientierte Modellierung, Einsatz von Problemlösungsmethoden der Informatik zur Förderung von Denkstrukturen, die nicht nur zur Bearbeitung von informatikspezifischen Problemstellungen genutzt werden (vgl. [Humbert 1999]). Die Skizze eines 'Curriculumvorschlags' für die Jahrgangsstufen 5 – 10 und ergänzende Varianten sind in ihrer Darstellung jedoch so knapp, dass damit mehr Fragen erzeugt als Antworten gegeben werden.

Die Vorlage des Gesamtkonzepts zur informatischen Bildung [Gesellschaft für Informatik 2000] durch die GI spiegelt das inzwischen entwickelte Selbstbewusstsein der Didaktik der Informatik wider und verdeutlicht darüber hinaus Handlungsnotwendigkeiten im Bereich der allgemeinen Bildung. Die Orientierung an Leitlinien<sup>191</sup> zeigt eine Umorientierung der Zielvorstellungen von der Fachstruktur zu einer didaktisch reflektierten Gestaltungsgrundlage. Es werden keine Hinweise für die konkrete Unterrichtspraxis gegeben, so dass zur Zeit die Anforderungen des Gesamtkonzepts hinsichtlich beispielhafter Umsetzungen nicht geprüft werden können.

2000

<sup>191</sup> Interaktion mit Informatiksystemen; Wirkprinzipien von Informatiksystemen; Informatische Modellierung; Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft

2001

Eine fachwissenschaftlich begründete Strukturierung eines Zugangs zur objektorientierten Modellierung wird von Torsten BRINDA vorgestellt. Mit informatischen Methoden wird der Gegenstandsbereich untersucht und es werden erste Ergebnisse dieser Untersuchungen dargestellt. Hervorzuheben ist die Aussage, dass „die sachlogische Struktur der Fachwissenschaft allein ungeeignet zur Strukturierung des Unterrichts [ist]“ [Brinda 2001, S. 79].

Die Analyse von Aufgaben aus Lehrbüchern (zum Informatikstudium insbesondere zur Objektorientierung) wird in dem Beitrag zu Aufgabenklassen verdichtet, die in einer hierarchischen Anordnung der Fachkonzepte in einer Baumstruktur ihren Ausdruck findet [Brinda 2001, S. 81].

Der scheinbare Widerspruch zu der oben dokumentierten Aussage wird von BRINDA durch den ausdrücklichen Hinweis auf das Ziel der Strukturierung aufgelöst: „Ziel [...] ist es nicht, kreative Prozesse der Unterrichtsgestaltung durch schematische Darstellungen einzuengen. Vielmehr soll dazu beigetragen werden, geeignete Fachkonzepte fachdidaktisch leichter zugänglich zu machen [...]“ [Brinda 2001, S. 83]. Die dynamischen Basiskonzepte werden nicht expliziert. Dies ist m. E. ein notwendiger weiterer Baustein, zu dem aber in dem vorgelegten Beitrag noch keine Ergebnisse vorgestellt werden.

Aus fachdidaktischer Sicht ist es beispielsweise möglich, die objektorientierte Modellierung informal mit Sequenzdiagrammen zu beginnen, ohne den Begriff Objekt vorauszusetzen, ohne den Begriff Klasse überhaupt zu thematisieren. Diese unterrichtsmethodische Variante korreliert mit dem Ansatz der Anwendungsfälle, wie er in der Softwaretechnik eingesetzt wird, ohne diese fachwissenschaftliche Methode einzusetzen oder gar vorauszusetzen.

Marco THOMAS beschreibt den Suchraum zur Identifikation der Vielfalt der Modelle in der Fachwissenschaft. In [Thomas 2001] zeigt er, dass unter dem Gesichtspunkt der allgemeinen Modelltheorie (nach Herbert STACHOWIAK [Stachowiak 1973]) bezogen auf die Informatik festzustellen ist, dass die Anforderung, Modellbildung als Leitlinie informatikdidaktisch zu reflektieren, einen notwendigen Anfang darstellt, der in Richtung auf Modellierung von Modellen weiterentwickelt werden kann. Diese Linie wird in seiner inzwischen vorliegenden Dissertationsschrift [Thomas 2002] weiter vertieft (siehe 2002, S. 71).

Mit Überlegungen zur Nachhaltigkeit [Humbert 2001a] versucht Ludger HUMBERT den Fokus auf überdauernde Elemente einer informatischen Bildung zu richten. Dies kann nur durch die Verankerung in der allgemeinen Bildung gelingen. Dabei wird deutlich, dass der Informatikdidaktik eine große Bedeutung zukommt, die auch darin besteht, eine Bestandsaufnahme bezüglich der informatischen Bildung vorzulegen. Für die Umsetzung der informatischen Bildung schlägt er ein Modulkonzept als strukturelle Klammer für die notwendigen informatischen Inhalte auf den verschiedenen Ebenen vor. Dabei weist er dem Modul „Informatiksysteme verantwortlich nutzen und verstehen“ eine Schlüsselrolle zu und fordert, dass dieser auf jedem Niveau zu Beginn unterrichtlich einbezogen werden muß.

Eine integrative Medientheorie wird von Bardo HERZIG zur Diskussion gestellt. Damit sollen Überlegungen zur Medienbildung und informatische Elemente aufeinander bezogen werden. „[Es] wird deutlich, dass informatische Prozesse bereits in frühen Entwicklungsstadien angelegt sind [...]“ [Herzig 2001, S. 107]. Zur Frage nach dem allgemein bildenden Schulfach Informatik verweist HERZIG auf eine zunächst fachunabhängig zu erstellende Gesamtkonzeption (vgl. [Herzig 2001, S. 118]). Der Hinweis auf die Empfehlungen der GI zur Informatischen Bildung und Medienerziehung [Hauf-Tulodziecki u. a. 1999] zeigt, dass der Schulinformatik im Kontext dieser Überlegungen eine wichtige, gestaltende Rolle zufällt, die bisher allerdings erst theoretisch zu einer anschlussfähigen Sicht geführt hat, die es weiter zu präzisieren und konkretisieren gilt.

Carsten SCHULTE wirft die Frage nach der Gestaltung von Informatiksystemen [Schulte 2001] auf, um einen über die konkrete Modellierung hinausweisenden Ansatz zu diskutieren. In einer informatikdidaktischen Skizze dokumentiert er die Geschichte der Diskussion der grundlegenden Bausteine der Informatik in der Schule. Er kommt zu dem Ergebnis, dass die Frage nach den „Auswirkungen der Modellierung“ nicht zufriedenstellend einbezogen wird, wenn der Unterrichtsprozess auf das Wasserfallmodell der Softwareentwicklung und auf Algorithmisierung bezogen ist. Allerdings bietet – nach seiner Einschätzung – der Wechsel zur Objektorientierung die Chance, zu inkrementellen, unscharfen Zieldimensionen gerecht werdenden Prozessen zu kommen, mit denen das gesamte unterrichtliche Feld erweitert werden kann. „Es gibt [...] Hinweise, dass durch eine Hinwendung zu offenen Fragestellungen (im Gegensatz zu algorithmisch lösbaren Problemen) sowohl die kognitiven Lernziele (Denkfähigkeiten), die informatischen

Lernziele (Fragestellungen und Werkzeuge, Methoden der Softwaretechnik) und der Aspekt der Wechselwirkungen zwischen Informatik und Gesellschaft profitieren können. Objektorientierung kann helfen, diese Öffnung zu ermöglichen“ [Schulte 2001, S. 15].

Dieser Beitrag macht am Beispiel des Softwareentwicklungsprozesses deutlich, dass fachliche Entwicklungen und ihre Diskussion im fachdidaktischen Kontext eine notwendige Voraussetzung für die Weiterentwicklung der Didaktik der Informatik darstellen. Problematisch erscheint der nicht einlösbare Verweis auf Transfermöglichkeiten, die zwar von jedem Schulfach in der Argumentation für die Notwendigkeit der jeweiligen Inhalte und Methoden herangezogen werden, aber bisher empirisch nicht nachweisbar sind.

Als ein Ergebnis seiner Dissertation kommt Marco THOMAS zu dem Schluß: „Informatische Modelle stellen ein Bildungsgut zur Enkulturation des Modellierens von Modellen dar und lassen sich an interessanten und anspruchsvollen Themen konstruktiv und explorierend erschließen. Die Merkmale der Allgemeinbildungsbegriffe [...] können auf informatische Modelle und auf das informatische Modellieren von Modellen mit Erfolg angewandt werden, so dass eine Leitlinie »Informatische Modellbildung« als allgemeinbildend gelten darf“ [Thomas 2002, S. 76].

2002

Belege für den Stellenwert der Modellierung findet THOMAS durch eine (quantitative) Analyse der vorliegenden universitären Skripte zu Veranstaltung aus dem Bereich der Kerninformatik. Unbeantwortet bleiben die Fragen nach einer Definition von „Informatik“ und damit der Beschränkung des Suchraums. Dennoch macht die vorgelegte Untersuchung deutlich, dass Fragen der Modellierung zum Selbstverständnis der Informatik einen wesentlichen Beitrag leisten. Das oben dargestellte Ergebnis weist in signifikanter Weise über die in den GI-Empfehlungen zum Gesamtkonzept ausgewiesene Leitlinie „Informatisches Modellieren“ (vgl. Fußnote 191) hinaus. Die mit den beschriebenen Ergebnissen verbundenen curricularen und unterrichtlichen Konsequenzen müssen allerdings noch erarbeitet und evaluiert werden, wie THOMAS an einigen Stellen unmissverständlich fordert.

Jahr	Autorin	Quelle	Charakterisierung	Bemerkungen
1989	Petra Knöß	[Knöß 1989]	Fundamentale Ideen der Informatik	In dieser Diss. wird Mathematikunterricht auf fundamentale Ideen der Informatik hin untersucht
	Rudolf Peschke	[Peschke 1989]	Krise des Informatikunterrichts	Schulinformatik steht auf dem Stand von 1976 – Fachwissenschaft hat sich erheblich weiter entwickelt
1991	Sigrid Schubert	[Schubert 1991]	Theoretische Elemente der Informatik	Schulinformatik muss grundlegende Elemente des Bezugsfaches umfassen
1993	Andreas Schwill	[Schwill 1993]	Fundamentale Ideen der Informatik	Kerninformatik ↔ Algorithmisierung, strukturierte Zerlegung und Sprache
	Rüdeger Baumann	[Baumann 1993]	Systemorientierter Ansatz	Problemlösen, Aufbau Informatiksystem, Grenzen der Informatik
	GI	[Schulz-Zander u. a. 1993]	Empfehlungen Sek II	Mensch-Computer; Formalisierung/Automatisierung geistiger Arbeit; Informatiksysteme, Gesellschaft/Umwelt
1995	Steffen Friedrich	[Friedrich 1995]	Allgemeinbildung klar, was nun?	Grundprinzipien der Informatik, Strategien der Problemlösungen und Grenzen, Gesichtspunkt und Konsequenzen der Modellbildung
	Dieter Engbring	[Engbring 1995]	Gestaltung der Technik	Konstruktiver Aspekt der Schulinformatik sollte betont werden
	Sigrid Schubert, Andreas Schwill	[Schubert und Schwill 1996]	Struktur des Schulfaches	Anmerkungen zu fundamentalen Ideen – Woher kommen die Inhalte?
1997	Sybille Krämer	[Krämer 1997]	Symbolische Maschinen	„Dritte Modalität“: neben instrumentellem und kommunikativem Handeln – spielerische Interaktion
	Peter Hubwieser, Manfred Broy	[Hubwieser und Broy 1997]	Informationszentrierter Ansatz	Modellierung (inkl. Implementierung) bildet das Zentrum der Schulinformatik
	Helmut Witten, Johann Penon	[Witten und Penon 1997]	Vernetzte Informatiksysteme	Unterrichtserfahrung – Kryptologie – Extrapolation
	Peter Berger	[Berger 1997]	Empirische Studie	Computer-Weltbild von Lehrerinnen – das neue Paradigma findet „seinen Lehrer“
1999	Johannes Magenheimer, Carsten Schulte	[Magenheimer u. a. 1999]	Dekonstruktion	Objektorientierte Modellierung – Projektorientierung, Systemtheoretische Didaktik
	Ludger Humbert	[Humbert 1999]	Bestandsaufnahme	Curriculare Vorüberlegungen
2000	GI	[GI 2000]	Perspektive für die Informatische Bildung	Orientierung an Leitlinien
2001	Torsten Brinda	[Brinda 2001]	Fachwissenschaft → Gestaltungshinweise zur Didaktik	Objektorientierte Modellierung
	Marco Thomas	[Thomas 2001]	Modelle in der Informatik	Informatik: alle Arten von Modellen – wird nicht thematisiert
	Ludger Humbert	[Humbert 2001a]	Nachhaltigkeit Informatischer Bildung	Nachhaltigkeit Informatischer Bildung
	Bardo Herzig	[Herzig 2001]	Medienbildung und Informatik	Integrative Medienbildungstheorie braucht Informatik als Bezug
	Carsten Schulte	[Schulte 2001]	Gestaltung als Ziel des Informatikunterrichts	Objektorientierung als Impuls für einen neuen Informatikunterricht?
2002	Marco Thomas	[Thomas 2002]	Modelle in der Informatik	Modellieren von Modellen – Bildungsgut zur Enkulturation

Tabelle 4.4: Ausgewählte deutsche fachdidaktische Veröffentlichungen – Quellen und Einordnung

### 4.2.2 Internationale Diskussion

Im Folgenden werden Beiträge zu einer Didaktik der Informatik dargestellt, die insoweit bedeutsam erscheinen, als sie ermöglichen, den engen Bereich bundesrepublikanischer Diskussionen zu überschreiten. Die Auswahl wurde nach folgenden Kriterien vorgenommen:

- lange Tradition (ACM),
- Verbreitung (IFIP/UNESCO) und
- Nähe zur bundesrepublikanischen Diskussion (EBERLE).

Im internationalen Feld der Didaktik für die Schulinformatik wurden bisher keine umfangreichere Vergleichsstudien durchgeführt.<sup>192</sup> Eine gründliche Untersuchung der Aktivitäten in verschiedenen Ländern kann aus Gründen mangelnder Ressourcen an dieser Stelle nicht stattfinden. Es steht zu hoffen, dass für eine breite internationale Vergleichsuntersuchung zukünftig Mittel bereitgestellt werden, um international bedeutsame Entwicklungen für die informatische Fachdidaktik zugänglich zu machen.

#### Model High School Computer Science Curriculum<sup>193</sup> – ACM (Association for Computing Machinery) 1993

Von der »Task Force on the Core of Computer Science« (ACM) wurde die folgende inhaltliche Bestimmung von *computer science as a discipline* erarbeitet: „Computer science and engineering is the systematic study of algorithmic processes – their theory, analysis, design, efficiency, implementation, and application – that describe and transform information. The fundamental question underlying all of computing is, What can be (efficiently) automated“ [Denning u. a. 1989, S. 16].<sup>194</sup> Die damit verbundene Sicht auf die Informatik stellt den Hintergrund für curriculare Überlegungen dar.

In der aus sieben Bereichen<sup>195</sup> bestehenden Themenübersicht des Curriculums für die allgemein bildende Sekundarstufe II fällt der Bereich Social, Ethical and Professional Context ins Auge. Die quantitative Ausgestaltung dieses mit den Inhalten: ethische, soziale, juristische und kulturelle Aspekte, sowie Verständnis der historischen Entwicklung der Informatik auszugestaltenden Bereichs ist recht gering, stehen doch nur 11 – von insgesamt 303 – Stunden – dafür zur Verfügung. Da dieser Bereich für die Diskussion bedeutsam ist, werden die von der ACM ausgewiesenen Elemente in Tabelle 4.5 dokumentiert.

In der Einführung wird deutlich gemacht, dass die Beschäftigung mit technischen Details vermieden werden muss: „Because the details of the technology change from day to day, keeping up with those details is difficult and often unproductive. Therefore the study of the subject must concentrate on the fundamental scientific principles and concepts of the field. [...] The focus of the course is on fundamental concepts of computer science. Several model course curricula show how different settings can be used to illustrate these concepts. As much as possible, students will conduct experiments and write programs that demonstrate the abstract concepts, confirm the theory and demonstrate the power of computers“ [ACM 1997, Introduction, Motivation]. Verschiedene Methoden zur Erarbeitung werden – auch in Kombination – für die Bereiche für möglich erachtet: die Erarbeitung mit Hilfe der Programmierung von Beispielen, durch Übungen, in Projekten und mit Hilfe von Berichten für die zentralen Bereiche – aber auch die Arbeit mit Anwendungspaketen. In der Zusammenfassung wird deutlich formuliert, dass es sich um ein verpflichtendes Curriculum für alle Schüler handelt, dessen Fundamente zu vermitteln sind. Diese liegen – nach dem o. g. Rahmen – in den Bereichen [Rechner-]Architektur, Betriebssysteme, Algorithmen und Datenstrukturen, Programmiersprachen und Softwaretechnik (in Summe über 85%, wenn die Einführung in eine konkrete Programmiersprache hinzugenommen wird). „Computer science is essential for the education of every citizen. The ACM model high school curriculum identifies the essential concepts in computer science which every high school student must understand“ [ACM 1997, Summary].

<sup>192</sup> vgl. Kapitel 1, S. 6

<sup>193</sup> [ACM 1993] und [ACM 1997], die 97er Fassung wurde gegenüber 1993 nur unwesentlich verändert.

<sup>194</sup> Diese Definition wurde von Wolfgang COY ins Deutsche übertragen [Coy 1992, S. 2f]: „Die Disziplin der Informatik ist das systematische Studium algorithmischer Prozesse, die Information beschreiben und transformieren; Theorie, Analyse, Entwurf, Effizienz, Implementierung und Anwendung dieser Prozesse. Die grundlegende Fragestellung der Informatik ist 'Was kann effizient automatisiert werden?'“

<sup>195</sup> Algorithms; Programming Languages; Operating Systems and User Support; Computer Architectures; Social, Ethical and Professional Context; Computer Applications; Additional Topics

Core topics	Recommended topics	Optional topics
Impact of technology on today's society	Future of computer technology	
Ethics in an electronic community	Risks and liability in computing, viruses	
	Computer support of the disabled	
	Software, public domain and private	Legal issues
	Privacy, reliability and system security	
Team solution of problems	Uses, misuses, and limits of computer technology	
	Electronic crime: stealing and spying, Intellectual property, infringements and penalties	

vgl. [ACM 1997, Course Topics and Models]

Tabelle 4.5: Social, Ethical, and Professional Context

Die curriculare Arbeit der ACM ist von großer Kontinuität geprägt. Damit verläuft allerdings die Weiterentwicklung in kleinen Schritten. Die Aufnahme des neuen Bereichs „Social, Ethical and Professional Context“ ist quantitativ sehr gering. Insgesamt ist festzustellen, dass die curriculare Orientierung an gewachsenen Teilgebieten der Informatik erfolgt. Eine konzeptionelle Orientierung als Klammer des Curriculums wird nicht expliziert.

1994

**Informatics for secondary education: a curriculum for schools** – IFIP/UNESCO (International Federation for Information Processing/United Nations Educational, Scientific and Cultural Organization)

Durchgängig wird der Begriff Informatics benutzt: „The following definition[s] have been used by the working party: Informatics: the science dealing with the design, realisation, evaluation, use and maintenance of information processing systems; including hardware, software organisational and human aspects, and the industrial, commercial, governmental and political implications (UNESCO/IBI)“ [van Weert u. a. 1994, S. 8].

Unter Berücksichtigung der weltweit normierenden Absicht, die mit der Veröffentlichung der Dokumente der UNESCO verbunden sind<sup>196</sup>, muss eine kritische Prüfung der Voraussetzungen und der Ziele vorgenommen werden. Als Hauptzielbereiche des UNESCO-Curriculums werden die folgenden Elemente ausgewiesen:

1. Computer Literacy [...],
  2. Application of IT Tools in other Subject Areas [...],
  3. Application of Informatics in other Subject Areas – Students should be able to use methods and techniques from informatics in combination with information technology tools to solve problems in other subject areas. This main objective should preferably be addressed in General Education at the Advanced Level.,
  4. Application of Informatics in Professional Areas – Students should be able to use methods and techniques from informatics in combination with information technology tools to solve professional problems from business and industry. This main objective should preferably be addressed in Vocational Education at the Advanced Level [...]
- General Education at the Advanced Level – Having fully met Objective 1 (Computer Literacy), the focus is on Main Objectives 2 and 3 [...] Objective 3 involves the following sequence of problem solving skills using the techniques and tools of informatics, namely the

<sup>196</sup> Konkret nachzuweisen ist dies im deutschen Sprachraum z. B. an dem schweizerischen Lehrwerk „Informatik“, das in der 5. Auflage vorliegt und im Vorwort deutlich macht: „[...] folgt den Richtlinien der Curricula der EDK (Eidgenössische Erziehungsdirektorenkonferenz) und der Unesco“ [Anderes u. a. 1999, S. III].

methodical modelling of the problem, design of an algorithmic solution, programming the solution either in a general or computer specific way, and an evaluation of the proposed solution. This implies that students have developed a functional model of a computer system and its programming environment.

[van Weert u. a. 1994, S. 9f]

Trotz der durch den Titel ausgewiesenen Orientierung auf Informatik wird in den Zielbereichen 1 und 2 eine an Anwendungen orientierte Ausrichtung expliziert. Informatik ist diesem curricularem Entwurf zufolge nachgelagert. Dies soll auf der anderen Seite durch eine grundlegend am problemlösenden Denken orientierten didaktischen Grundhaltung umgesetzt werden. Dabei wird – nach meiner Einschätzung – aus den vorgeschlagenen Bereichen kein durchgängiges Informatikcurriculum, sondern ein Konzept, das die Informatik isoliert und künstlich von Informatics Technology (IT) zu trennen versucht.

Die Vorschläge ACM und IFIP/UNESCO lassen unterschiedliche Ansätze erkennen, die dem Verständnis von Informatik zuzuschreiben ist, die jeweils zur Anwendung kommt. Die Ergebnisse auf der curricularen Ebenen weichen hinsichtlich des fachlichen und des pädagogischen Hintergrunds voneinander ab: IFIP/UNESCO fordert Problemlösekompetenz, verwendet den Begriff Modellierung und erklärt die Informatik als Hilfsmittel zur Problemlösung. ACM hingegen legt ein „konservatives“ Informatikverständnis zugrunde und fordert, dass die Inhalte auf zentrale wissenschaftliche Prinzipien und Konzepte orientiert zu erarbeiten sind.

Gerade aus dieser Diskrepanz heraus ist eine verstärkte internationale Auseinandersetzung um Informatik als Inhalt allgemeiner Bildung notwendig. Es wäre zu begrüßen, wenn die beiden curricularen Vorschläge so miteinander verzahnt würden, dass die Fachkonzepte der Informatik auf dem Hintergrund moderner pädagogischer Konzepte erfolgreich vermittelt werden.

### Überlegungen zu einer Didaktik der Informatik – Franz EBERLE

1996

In einer „Abschliessenden Übersicht“ fasst EBERLE die im Zusammenhang mit seiner Habilitationsschrift [Eberle 1996] entwickelten Elemente einer „Didaktik der Informatik ...“ zusammen. Dabei macht er u. a. deutlich:

Das Schulfach Informatik [...] hat sich seit seinen Anfängen aus einer Rechnerkunde zu einem Bildungsgefäß<sup>197</sup> entwickelt, über dessen Inhalte stark auseinanderdriftende Ansichten bestehen. Je nach Phase, Ansatz oder pragmatischer Festschreibung wurden und werden Themen der Wissenschaftsdisziplin Informatik, anderer Disziplinen der Informationswissenschaften, der technischen Anwendungen, der wirtschaftlichen und gesellschaftlichen Auswirkungen u. a. vorgeschlagen und behandelt. Darin spiegeln sich vor allem die Neuheit des Fachs, die rasante Entwicklung der Technologien, die unterschiedlichen Einschätzungen von deren Bedeutung, der z. T. fehlende oder zu wenig konsistente bildungstheoretische Begründungsrahmen, aber auch ideologische Differenzen wider. Dies führte zu einer Palette möglicher Bildungsmassnahmen, von der sich angesichts beschränkter Stundenzahlen alle Bildungsinteressierten und -verantwortlichen jenes Stück nehmen konnten und können, das ihrem Verständnis und ihren Wünschen entspricht [...].

Zum methodischen Regelwerk, das auf der Grundlage fachimmanenter Eigenheiten und allgemeindidaktischer Erkenntnisse erstellt und in schulpraktischen Empfehlungen konkretisiert worden ist, kann festgehalten werden, dass die unterrichtliche Verwirklichung der informationstechnologischen Bildung starke Eigenheiten aufweist, die sie – entgegen anderen Aussagen – zu einem keinesfalls zu unterschätzenden sondern zu einem anspruchsvollen Unterfangen machen, wofür allgemeindidaktische Überlegungen und Erfahrungen aus anderen Fächern allein nicht mehr genügen. [...].

[Eberle 1996, S. 429f]

<sup>197</sup> Rechtschreibung lt. Original

**Forschungsfragen – EBERLE**

Nach der Darstellung ausgewählter Elemente zur Umsetzung informatischer Inhalte stellt EBERLE „Eine Auswahl offener Forschungsfragen“ vor. „Viele Aussagen in dieser Arbeit mit deskriptivem Charakter sind empirisch gar nicht oder mangelhaft nachgewiesen, beruhen nur auf Alltagsbeobachtungen oder basieren auf nichtrepräsentativer qualitativer Forschungsmethodik. Daraus ergibt sich eine breite Palette von Forschungsfragen, die empirisch geklärt werden sollten“ [Eberle 1996, S. 427].

EBERLE hat in seiner Habilitationsschrift eine große Menge an Quellen zusammengetragen, die in seiner Auswertung zu den oben angegebenen offenen Forschungsfragen führen. Der Bearbeitung dieser offenen Fragen wird die Didaktik der Informatik einen Teil ihrer Forschungen zu widmen haben. Durch meine Studien möchte ich einen kleinen Beitrag dazu leisten. Auf dem Hintergrund der von EBERLE zu Recht eingeforderten empirischen Forschungsarbeiten ist es an der Zeit, neben qualitative Studien auch quantitative zu stellen, die die Chance eröffnen, aus vorhandenen Erfahrungen zu validen Aussagen in diesem Feld zu gelangen.

2000

**Information and Communication Technology in Secondary Education – A Curriculum for Schools – IFIP/UNESCO**

In der aktualisierten Auflage der Überlegungen von 1994<sup>198</sup> präsentieren IFIP/UNESCO [van Weert u. a. 2000] eine Überarbeitung zur Einordnung zentraler Begriffe.<sup>199</sup>

**„Informatics Technology** The technological applications (artefacts) of informatics in society.

**Information and Communication Technology (ICT)** The combination of informatics technology with other, related technologies, specifically communication technology.

In this document these definitions have been collapsed into one, all encompassing, definition of Information and Communication Technology (ICT). This implies that ICT will be used, applied and integrated in activities of working and learning on the basis of conceptual understanding and methods of informatics.

[van Weert u. a. 2000, S. 9]

Bezüglich der Inhalte kann aus Sicht der Informatik keine maßgebliche Veränderung gegenüber der Vorgängerausfassung festgestellt werden. In dem Curriculum findet sich ein deutlicher Hinweis auf die Zertifizierung curriculärer Elemente durch die im Europäischen Computer-Führerschein (European Computer Driving Licence, ECDL) angebotenen Fertigkeiten (skills) [van Weert u. a. 2000, S. 42]. Diese Entwicklung ist mit der Konsequenz verbunden, dass m. E. durch die offensichtlichen Möglichkeiten zur Kommerzialisierung ein zunehmender Teil der informatischen Bildung durch außerschulische Träger abgedeckt wird.<sup>200</sup>

**The Informatics Curriculum Framework 2000 [ICF-2000] – IFIP/UNESCO**

Mit [Mulder und van Weert 2000] wird von IFIP/UNESCO ein curriculärer Rahmen für Informatik in der Sekundarstufe II (Higher Education) vorgelegt.

Nach Auffassung des IFIP/UNESCO-Autorenteams kann die Bezeichnung Informatik nicht mehr kommentarlos vorausgesetzt werden. Auf der inhaltlichen Ebene wird deutlich, dass Modellierung Eingang in die Themenliste<sup>201</sup> des Kerncurriculums<sup>202</sup> gefunden hat. In diesem curricularen Vorschlag werden Elemente des ECDL zur Umsetzung konkreter Elemente vorgeschlagen. Die Module des ECDL werden im Anhang G des curricularen Rahmenvorschlags ausführlich vorgestellt [Mulder und van Weert 2000, S. 131-136].

Es wird herausgestellt, dass für die Kommission eine terminologische Schwierigkeit bestand:

<sup>198</sup> [van Weert u. a. 1994] – bemerkenswerterweise erscheint Informatik nicht mehr in dem 2000er Titel

<sup>199</sup> Die Arbeitsdefinition für „Informatics“ wird aus dem 1994er Vorschlag unverändert übernommen und hier nicht wiederholt.

<sup>200</sup> Dies wird z. B. in Österreich praktiziert (vgl. [Micheuz 2002, Folie 7]).

<sup>201</sup> „Representation of information, Formalism in information processing, Information modelling, Algorithmics, System design, Software development, Potentials and limitations of computing and related technologies, Computer systems and architectures, Computer-based communication, Social and ethical implications, Personal and interpersonal skills, Broader perspectives and context (includes links with other disciplines)“ vgl. [Mulder und van Weert 2000, S. 31ff]

<sup>202</sup> „core curriculums“

There is a continuing discussion about the best term to be used for the broad area as addressed in the IFIP/UNESCO project. Staying away from this discussion, we adopt the term 'informatics' (or its abbreviation by the letter 'I'), just for convenience. This term has its significance mainly in the European tradition. However, within the IFIP/UNESCO project it is intended to be nothing more than an 'umbrella' label. Hence, 'informatics' or 'I' refers to a diverse, yet related family of domains, including 'computing', 'computer science', 'computer engineering', 'information systems', 'management information systems', 'computer information systems', 'software engineering', 'artificial intelligence' or 'AI', 'information technology' or 'IT', 'information and communication technology' or 'ICT', and so on.

[Mulder und van Weert 2000, S. iv]

Die bereits oben angeführte Kommentierung der Vorschläge von ACM versus IFIP/UNESCO ist weiterhin gültig, wenn sich auch eine gewisse Annäherung der Positionen andeutet, wie durch das Zitat aus dem aktuellen Vorschlag der IFIP/UNESCO deutlich wird. Gemessen an dem Bildungsanspruch, der in der deutschen Diskussion eine wesentliche Rolle spielt, muß festgestellt werden, dass die Zugänge in den Vorschlägen von ACM und IFIP/UNESCO sehr pragmatisch-inhaltlich und wenig konzeptionell ausgearbeitet sind. Darüber hinaus wird deutlich, dass die IFIP/UNESCO-Vorschläge einer Entwicklung Vorschub leisten, den Erwerb von Fertigkeiten von der Reflexion über die Möglichkeiten zu trennen. Diese Entwicklung ist abzulehnen, führt sie doch zu einer benutzungsorientierten Sicht auf die Informatik, die den pädagogischen Bestrebungen einer bereits wesentlich über den bloßen Erwerb von Fertigkeiten zur Handhabung und Benutzung von Informatiksystemen hinausreichenden Informatikdidaktik entgegengesetzt ist. Fertigkeiten sind im Kontext auszubilden und zu beleuchten. Sie sollten aus didaktischen Gründen nicht künstlich vom Kontext getrennt werden.

Um die aktuelle Situation in zwei Nachbarländern zu charakterisieren, werden Aussagen zur aktuellen Situation in der Schweiz und in Frankreich dokumentiert.

#### – Schweiz

Im Jahr 1995 wurde Informatik als eigenständiges Fach in der allgemein bildenden Sekundarstufe II abgeschafft, in der vermeintlichen Hoffnung, die Informatik könne durch Integration in die anderen Fächer "nebenbei" überall unterrichtlich umgesetzt werden. Dies hat sich offenbar inzwischen als nicht praktikabel herausgestellt, so dass zunehmend Stimmen laut werden, Informatik wieder als Schulfach einzuführen.

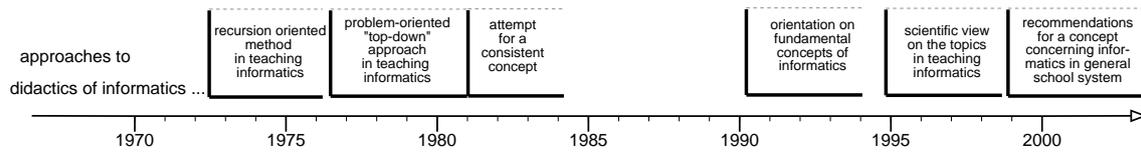
[Quelle: persönliche Mitteilung, Dr. Werner HARTMANN, ETH Zürich 2002]

#### – Frankreich

Ergebnisse einer Recherche im Sommer 2002 zum Stand der Schulinformatik in Frankreich:

Ein Fach Informatik, wie es bei uns in der gymnasialen Oberstufe unterrichtet wird, gibt es im französischen Schulsystem nicht. Auf dem «collège» wird im letzten Jahr, der sogenannten «troisième» ggf. auch noch nach dem Übergang zum «lycée» (vergleichbar mit dem deutschen Gymnasium) im ersten Jahr, der sogenannten «seconde» ein praktisch ausgerichteter Kurs mit zwei Kompetenzstufen angeboten, dem «brevet informatique niveau 1» und «... niveau 2», kurz B2i genannt. Die jeweiligen Kompetenzen sind unter <http://www.educnet.education.fr/plan/b2i.htm> detailliert aufgelistet. Darüber hinaus können die Schülerinnen im ersten Jahr des Gymnasiums, der «seconde» im Rahmen ihrer Wahlmöglichkeiten 2 Stunden pro Woche einen Kurs in Form von «gestion et informatique» oder «informatique et électronique» belegen. In den letzten beiden Klassen des «lycée» wird bei allen Differenzierungsmöglichkeiten (L=langues, ES = économique et social, S= scientifique) das Fach Informatik nicht mehr angeboten. Mit dem Schuljahr 2002/2003 befindet sich in Erprobung das B2i niveau 3, das vergleichbar ist mit dem «certificat informatique et internet» C2i; dieser Kurs wird an den Hochschulen/Universitäten angeboten.

[Quelle: persönliche Mitteilung, StD Rainer Haberkern, Fachleiter Französisch am Studien-seminar für das Lehramt für die Sekundarstufe II, Hamm 2002]



[Humbert und Schubert 2002, S. 22]

Abbildung 4.2: Zeitleiste zu didaktischen Orientierungen

## Prüfung/Schärfen/Verdichten der Arbeitshypothese zur Lerntheorie

Die Ergebnisse der Beiträge zur Didaktik der Informatik, die einer kritischen Würdigung unterzogen wurden, machen eine Erweiterung der als Ergebnis des 3. Kapitels formulierten Arbeitshypothese (vgl. S. 54) erforderlich, die hier aus Gründen der Lesbarkeit wiederholt wird:

- Die Bedeutung des Subjekts für erfolgreiche Lernprozesse im Sinne institutionell-politischer Vorgaben führt in der Konsequenz zu Schülerorientierung und projektorientierten Unterrichtsformen – die durch informatische Methoden bereits vorausgesetzt werden.<sup>203</sup>

Die von KRÄMER als „spielerische Interaktion“ bezeichnete dritte Modalität des Handelns (siehe Abschnitt 4.2.1, S. 66) stellt für die Auseinandersetzung mit Informatiksystemen im Bildungsprozess eine bisher nicht aufgenommene Herausforderung dar. Für den Informatikunterricht sind damit die Perspektiven Denkzeug, Spielzeug verfügbar zu machen. Die Bedeutung der Überlegungen von THOMAS zur „Modellierung von Modellen“ ist zu konkretisieren und auf ihre Eignung für den konkreten Informatikunterricht zu untersuchen. Dies führt zu einer Erweiterung der Arbeitshypothese:

- Die Bedeutung des Subjekts für erfolgreiche Lernprozesse im Sinne institutionell-politischer Vorgaben führt in der Konsequenz zu Schülerorientierung und projektorientierten Unterrichtsformen – die durch informatische Methoden bereits vorausgesetzt werden.<sup>204</sup> Darüber hinaus gilt es, metatheoretische Überlegungen bei der Gestaltung von Lernprozessen zu berücksichtigen.

Aus den bisher vorgestellten Überlegungen werden Fragestellungen abgeleitet, die in den folgenden Ausführungen als Richtschnur dienen. Dabei wird mit den ersten beiden Fragestellungen<sup>205</sup> ein konstruktiver Beitrag entwickelt. Ein Aspekt der Umsetzung soll unter der besonderen Blickrichtung des „Bildes der Informatik“ bei Schülerinnen durch die dritte Fragestellung beleuchtet werden.

### Fragestellungen

#### 1. Zugänge zu Problemklassen

Welche Zugänge erweisen sich als erfolgreich und richtungweisend, so dass die Chance besteht, dass Schülerinnen im Rahmen des Informatikunterrichts eine von aktuellen Moden unabhängige Informatische Bildung erfahren?

#### 2. Strukturierung der Fachinhalte unter didaktischen Gesichtspunkten

Auf welche Weise können die Inhalte so strukturiert werden, dass nachhaltig Informatische Bildung vermittelt wird?

#### 3. Einfluss des Informatikunterrichts auf das Bild der Informatik bei Schülerinnen

Wie verändert ein bewusst auf die Zielsetzung „Generierung eines validen Bildes der Informatik als Wissenschaft“ orientierter Informatikunterricht die Sichtweise der Schülerinnen auf die Informatik?

<sup>203</sup> Problemlösen, Projektarbeit, Spiral- und Stufenmodelle (hierarchische [De-]Konstruktion) sind wichtige Elemente der Didaktik der Informatik.

<sup>204</sup> siehe Fußnote 203

<sup>205</sup> unter Maßgabe der entwickelten Arbeitshypothesen

## 4.3 Problemlösen – die zentrale Kategorie des Informatikunterrichts

### Zum Begriff, seine Einordnung in die Informatik und die Pädagogik

Problemlösen orientiert sich in der Fachwissenschaft Informatik häufig an ingenieurmäßigen Arbeitsweisen. Für Informatikerinnen bezeichnet Problemlösen einen Prozess, der dazu führt, dass eine vormals von Menschen/Organisationen durchgeführte Tätigkeit auf einen [halb-]automatischen Ablauf umgestellt wird. Eine informatische Problemlösung stellt typischerweise ein Informatiksystem oder Teile davon zur Verfügung.

Problemlösen im Zusammenhang mit schulischer Bildung verweist auf fächerübergreifende Kompetenzen. Probleme lösen zu können besteht – so verstanden – darin „lebensraumübergreifende“ Kompetenzen auszubilden. Diese werden auch als „Cross-Curricular Competencies (CCC)“ bezeichnet. Vorüberlegungen im Kontext internationaler Vergleichsuntersuchungen zeigen exemplarisch die Bedeutung dieser Kompetenzen (vgl. Abschnitt 3.3, S. 52). Zudem ist Problemlösen im pädagogischen Kontext ein Unterrichtskonzept und eine unterrichtliche Methode (vgl. Abschnitt 3.2, S. 42). „Problemlösen ist zielorientiertes Denken und Handeln in Situationen, für deren Bewältigung keine Routinen verfügbar sind. Der Problemlöser hat ein mehr oder weniger gut definiertes Ziel, weiß aber nicht unmittelbar, wie es zu erreichen ist. Die Inkongruenz von Zielen und verfügbaren Mitteln ist konstitutiv für ein Problem. Das Verstehen der Problemsituation und deren schrittweise Veränderung, gestützt auf planendes und schlussfolgerndes Denken, sind konstitutiv für den Prozeß des Problemlösens“ [OECD 2001, S. 25].<sup>206</sup> Darüber hinaus stellt der soziale Kontext eine wichtige Rolle für das Problemlösen dar. Dies gilt vor allem dann, wenn z. B. Problemstellungen geklärt oder von den Schülerinnen selbst entwickelt und anschließend kooperativ bearbeitet werden. Die Qualität des Problemlösens wird bestimmt durch das Verständnis der Problemsituation, die Denkprozesse bei der Problembearbeitung und die Angemessenheit der erreichten Lösung (nach [OECD 2001, S. 25f]). Im Vordergrund stehen „authentische Aufgaben, die von Situationen ausgehen, die zwar gelegentlich fiktiv sein mögen, aber doch die Art von Problemen repräsentieren, mit denen Schüler im wirklichen Leben konfrontiert werden“ [OECD 2001, S. 26].

Problemlösen wird in Anlehnung an George PÓLYA (vgl. Abschnitt 3.5, S. 43) durch die folgenden Phasen beschrieben: Problem aufwerfen, Problem verstehen, Aufstellen eines Plans, Ausführen des Plans, Reflexion – Evaluation. Dieses „[...] Phasenmodell hat heuristischen Nutzen, wenn Problemlöseprozesse vorstrukturiert [und exemplifiziert] werden [. Damit ist die Anwendung als... ] Technik des Problemlösens, als didaktische Strukturierung oder als Raster zur Diagnose von Problemlöseleistungen [möglich...]. Das Modell [...] dient nicht zur... ] Beschreibung tatsächlicher kognitiver Prozesse. Diese sind [...] vielfältiger und stärker verschachtelt, als ein einfaches Phasenmodell es [...] beschreiben könnte]. Wie systematisch jemand vorgeht – ob er beispielsweise einzelne Komponenten der Problemsituation in kontrollierter Weise untersucht, ob er den Lösungsprozess global oder lokal plant, ob er alternative Lösungsschritte gezielt ausprobiert, ob er Feedback sucht und nutzt – [...] ist eine wichtige qualitativ unterscheidbare Ausprägung der [...] Problemlösekompetenz“ [Baumert u. a. 1999, S. 4].

### Projektunterricht – oft gefordert, aber selten dokumentiert

Neben dem Begriff Problemlösen hat auch der Begriff Projektunterricht Eingang in die Didaktik der Informatik gefunden (vgl. Kapitel 3, S. 47). In der folgenden Darstellung wird die Spannweite des verwendeten Projektbegriffs im Zusammenhang mit dem Informatikunterricht verdeutlicht. Die mit der jeweiligen Schwerpunktsetzung verbundenen Konsequenzen werden dargestellt und bewertet.

### Gescheiterter Projektorientierter Unterricht

Die fehlende begriffliche Schärfe führte beispielsweise im Kontext der ITG zu der Forderung nach projektorientierten Unterrichtskonzepten. Dabei sollte die Umsetzung mit detailliert vorbereiteten Materialien

<sup>206</sup> vgl. auch [Baumert u. a. 1999, S. 3]

(bis hin zu vorgefertigten Arbeitsblättern) zu neuen Inhalten erfolgen. Durch diese Verbindung des für die Lehrerinnen neuen Inhaltes mit einem neuen methodisch-didaktischen Konzept wurde der Sache des Informatikunterrichts nicht gedient. Deutlich wird dies bei der externen Evaluation<sup>207</sup> am Beispiel der geschlechtsspezifischen Rollenzuweisung: „Bei verschiedenen Unterrichtsbeobachtungen konnten wir beobachten, daß aufgrund des geringen Kenntnisvorsprungs der Lehrenden gegenüber den Schüler/innen häufig Informatiklehrer um Hilfe gebeten werden mußten. Häufig ist die hilfeschuchende Person eine Frau und der Hilfegebende ein Mann. Dies kann leicht Vorurteile bei Schüler/innen bestärken, daß Frauen "keine Ahnung" von Naturwissenschaften und Technik, insbesondere neuen Technologien haben“ [Altermann-Köster u. a. 1990, S. 159]. Daraus ist m. E. eine Schlussfolgerung zu ziehen: Alle Lehrerinnen müssen über fundierte informatische Kenntnisse verfügen, damit die beschriebene Situation nicht eintritt. Dieser naheliegende Schluss wird allerdings nicht gezogen. Es wird deutlich, dass fachliche Sicherheit der Lehrerinnen eine unabdingbare Voraussetzung für die Nutzung flexibler Unterrichtsformen ist.

### Verständnis von Projektunterricht

Im Informatikunterricht wird (seit der Vorlage der GI-Empfehlungen [Brauer u. a. 1976]) Projekten eine besondere Bedeutung zuerkannt – häufig verstanden als Softwareprojekt und damit der Bearbeitung einer komplexen informatikbezogenen Problemstellung – i. d. R. interpretiert als komplexe softwaretechnische Aufgabe. Zur Bearbeitung solcher Aufgaben werden Methoden des Projektmanagements vorgeschlagen, wie sie im professionellen Umfeld eingesetzt werden (vgl. [Koerber und Reker 1982, S. 81ff]). [Koerber und Reker 1982, S. 85]: „Dieser Teil [...] soll sich mit den Voraussetzungen zur Durchführung von Projekten mit Schülern im Informatikunterricht beschäftigen. Übungsaufgabe [<sup>208</sup> ...]: Zählen Sie in Stichworten auf, welche Kriterien ein 'gutes' Programm erfüllen muß!“ Eine Zuordnung von Projektphasen aus der Informatik zu Unterrichtsphasen wird tabellarisch dargestellt [Koerber und Reker 1982, S. 113]. Damit wird deutlich, wie die Phasen eines Softwareprojektes – nach Auffassung von Bernhard KOERBER und Jörg REKER – auf Unterrichtsphasen abgebildet werden können. Dabei wird das informatische Projektverständnis als Vorlage für projektorientierten Informatikunterricht herangezogen. Die pädagogischen Dimension der in Abschnitt 3.2 (S. 43ff) dargestellten Projektmethode findet keine explizite Berücksichtigung.

Eberhard LEHMANN stellt in [Lehmann 1985] eine Reihe durchgeführter [Software-]Projekte an Hand der Ergebnisse vor. Die Auseinandersetzung mit dem Projektbegriff der Pädagogik findet in einem Abschnitt gegen Ende des Buches statt (vgl. [Lehmann 1985, S. 204-207]) und erschöpft sich in einigen allgemeinen und im Wesentlichen pragmatisch orientierten Ausführungen. Es entsteht der Eindruck, dass „Projektarbeit“ im Informatikunterricht einzig dem Ziel verpflichtet ist, ein funktionierendes Softwaresystem zu erstellen. Diese einseitige Sichtweise

- auf ausgewählte Fachinhalte und
- auf eine ausgewählte (fachdidaktische) Vorgehensweise

reicht m. E. so nicht aus, dem allgemein bildenden Zielen im Schulfach Informatik gerecht zu werden. Darüber hinaus geht der Autor davon aus, dass durch das Studium seines Buchs eine Grundlage für Projektarbeit im Informatikunterricht geschaffen werden kann:

In Ihrem bisherigen Informatikunterricht oder durch Selbststudium haben Sie grundlegende Kenntnisse in der Datenverarbeitung erworben. Sie beherrschen die wichtigsten Bestandteile einer Programmiersprache und können dabei insbesondere mit Prozeduren, Records und Dateien umgehen. Sie haben Ihre Erfahrungen durch die Bearbeitung vieler kleiner Probleme gewonnen und möchten sich nun an umfangreicheren Aufgabenstellungen versuchen. [...] An dieser Stelle setzt im fortgeschrittenen Informatikunterricht oder in Ihrer häuslichen Arbeit mit dem Rechner die Bearbeitung umfangreicher Softwareprodukte ein, die auch in der DVA-Praxis [<sup>209</sup>] wichtig sind. Man spricht in diesem Zusammenhang von Projekten. [...]

<sup>207</sup> dokumentiert in [Altermann-Köster u. a. 1990] Grundlage des Unterrichts [Der Kultusminister des Landes Nordrhein-Westfalen 1985]

<sup>208</sup> Für Lehrerinnen, die das Material durcharbeiten.

<sup>209</sup> DVA – Datenverarbeitungsanlage

Zur Durchführung derartiger Projekte benötigen Sie vielleicht weitere Programmierkenntnisse, aber insbesondere fehlt es Ihnen an Methoden, wie man an die sehr komplexen Problemstellungen herangeht. Zur Erarbeitung derartiger Methoden und zur Erweiterung Ihrer DVA-Kenntnisse wird Ihnen [...] ein fertiges Softwareprodukt vorgestellt, das Sie auch gut in Ihrem Privatbereich einsetzen können.

[Lehmann 1985, S. 7]

Die vorgestellten Befunde machen deutlich, dass eine deutlichere begriffliche Abgrenzung des Projektbegriffs im Kontext des Informatikunterrichts sowohl von dem Projektbegriff der Informatik aber auch von dem Projektbegriff der Pädagogik notwendig erscheint. Mit anderen Worten: es gilt auszuweisen, was als Projekt im Informatikunterricht zu bezeichnen ist und damit vor allem die mit dem Projektunterricht im Kontext der Schulinformatik verfolgten Ziele deutlich zu machen. Auf einer fachlich fundierten Grundlage muss der Projektbegriff der Fachwissenschaft einer kritischen Betrachtung unterzogen werden, bevor er für didaktisch gestaltete unterrichtliche Zwecke genutzt wird. Dies setzt voraus, dass die Lehrerin vor der Durchführung von Informatikprojekten im Unterricht eigene Erfahrungen in Informatikprojekten gesammelt hat – diese Erfahrung kann weder durch ein Seminar „über Projekte“ noch durch eine Vorlesung „über Projektunterricht“ erworben werden – die einzige Möglichkeit besteht darin, in ein „reales Projekt“ aktiv eingebunden zu sein. Damit soll eine realistische Sichtweise auf Informatikprojekte ermöglicht werden. Bei der Durchführung von Projekten im Informatikunterricht ist (vor allem bei Lehrerinnen – nicht aber bei Schülerinnen) eine Sichtweise verbreitet, die dazu führt, dass ein Unterrichtsprojekt [nur] dann als gelungen betrachtet wird, wenn das gewünschte Produkt als Ergebnis des Projekts „geliefert“ wird und den Ansprüchen der Auftraggeber mindestens genügt. Projekte, die „scheitern“, können durchaus lernförderlich sein – nicht nur „gelungene Projekte“. Bei kommunizierten Projektergebnissen werden typischerweise die Ergebnisse<sup>210</sup> mitgeteilt, selten der Weg, auf dem das Ergebnis erreicht wurde, die Hindernisse, die nicht beseitigt werden konnten. Damit verpflichten sich die Lehrerinnen einer „Kultur des Gelingens“, die für jede Innovation den Erfolg vorsieht und m. E. nicht realistisch mit den Ressourcen und den allgemein bildenden Zielen des Informatikunterrichts umgeht.

## 4.4 Besondere Zieldimensionen des Informatikunterrichts

Im Schulfach Informatik besteht die Notwendigkeit der Berücksichtigung besonderer Zieldimensionen, die im Sinne eines „heimlichen Lehrplans“<sup>211</sup> Eingang in den Unterricht gefunden haben (vgl. Zitat von Peter BERGER, Abschnitt 4.2.1 – 1997, S. 68). Veränderte Anforderungen an die Wissenschaft Informatik in der modernen Gesellschaft und die Schulinformatik führen zur notwendigen regelmäßigen Vergewisserung der Lehrerin, provoziert eine Innovationskraft und -bereitschaft bezogen auf die fachliche und methodische Orientierung der Informatik in der Schule. Diesen Anforderungen kann nur durch eine qualifizierte fachdidaktische Vorbereitung auf einer soliden fachlichen Basis Rechnung getragen werden, die durch regelmäßige Fortbildungen Reflexion und Aktualisierung erfahren.<sup>212</sup> Andererseits gilt es, die konkreten [und aktuellen] Interessen der Schülerinnen zu berücksichtigen.

Diese Zielorientierungen sind mit einigen Randbedingungen verbunden:

- Erheblich höherer Zeitaufwand der Lehrerinnen für die Unterrichtsvorbereitung, als für lehrgangsartig strukturierten Unterricht.
- Die Schülerinnen müssen in ihrer Interessenlage ernst genommen werden und die Notwendigkeit der Erarbeitung theoretischer Elemente erkennen können.

<sup>210</sup> Dies ist im Schulfach Informatik fast ausschliesslich „funktionierende Software“ die entwickelt wurde.

<sup>211</sup> hier allerdings nicht negativ konnotiert

<sup>212</sup> Diesen Anforderungen ist durch die „dritte Phase“ der Lehrerbildung Rechnung zu tragen, die verpflichtende Qualifikationsbausteine umfasst, die auf einer ausgewiesenen fachlichen und fachdidaktischen Basis durchgeführt werden.

- Für die praktische Umsetzung und die Erprobung theoretisch erarbeiteter und durchdrungener Konzepte müssen den Schülerinnen Zeit und Ressourcen (vernetzte Computer) in ausreichender Zahl auch außerhalb der Informatikräume<sup>213</sup> zur Verfügung gestellt werden.
- Um inhaltliche und methodische „Spiel“räume füllen zu können, sind die – ohne Frage notwendigen – Reglementierungen zu minimieren.
- Für eine Bewertung der Umsetzung der Leitkonzepte Projekt, Produkt, Team, Diskussion, beraten, delegieren, mitbestimmen und kooperieren<sup>214</sup> gibt es nur in beschränktem Maße Muster, auf diesem Feld gilt es Erfahrungen zu sammeln: besondere Lernleistung; Facharbeiten; Vorträge; Projektarbeiten, in denen die Leistungen der einzelnen Schülerin nicht ohne weiteres erkennbar sind.

---

<sup>213</sup> z. B. in der Schulbibliothek, in Arbeits- und Aufenthaltsräumen

<sup>214</sup> vgl. Abschnitt 4.2.1– 1997, S. 68

Die einzige Methode, die diesem erbarmungswürdigen Sohn unserer Epoche übriggelassen ist, um einmal selbst ein Ding zu machen, besteht eben darin – anderen Rohstoff bietet ihm seine Fertigwarenwelt nicht mehr – ein fertiges Produkt auseinanderzunehmen; und nachdem er so, zu Demolierung verurteilt, „Rohstoff“ aus dem fertigen Dinge hergestellt, dieses Ding in zweiter Schöpfung noch einmal herzustellen; wodurch er sich die kleine Freude verschafft, es selber noch einmal oder mindestens beinahe selber gemacht zu haben.

[Anders 1983, S. 201]

## Kapitel 5

# Lehrexpertise zum Informatikunterricht

In den ersten Kapiteln der vorliegenden Arbeit wurden theoriegeleitete Überlegungen zu Informatik und Wissenschaftstheorie sowie zu Lerntheorien dargestellt, die durch den Bezug auf Ergebnisse aus dem Kontext der Schulinformatik schrittweise zu Arbeitshypothesen verdichtet wurden (vgl. Abschnitt 4.1, S. 60 sowie Abschnitt 4.2, S. 78).

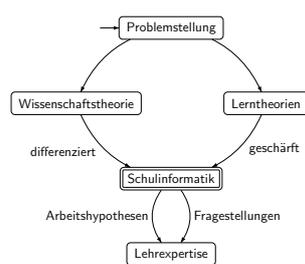


Abbildung 5.1: Struktur Kap. 1–5

Darüber hinaus wurde „Problemlösen“ als zentrale methodische Kategorie des Informatikunterrichts dargestellt (vgl. Abschnitt 4.3) und die besondere Zieldimensionen des Informatikunterrichts und ihre Konsequenzen beschrieben (vgl. Abschnitt 4.4).

Der Prozess der Entwicklung der Arbeitshypothesen und Fragestellungen in den Kapiteln 1 bis 4 ist in der nebenstehenden Strukturübersicht (Abbildung 5.1) skizziert. Im vorliegenden Kapitel werden die Arbeitshypothesen einer ersten Prüfung unterzogen. Diese Untersuchung soll Expertinnenwissen auf die ausgewiesenen Hypothesen beziehen, um

so zu einer qualifizierten Differenzierung der Arbeitshypothesen beizutragen. Damit sollen Positionen ausgewiesener Expertinnen<sup>215</sup> hinsichtlich des Informatikunterrichts bei den weiteren Untersuchungen berücksichtigt werden. Um Ziele, Orientierungen, Vorlieben, Kenntnisse, aber auch bisher nicht beschriebene und veröffentlichte Erfahrungen von Informatiklehrerinnen in einem aktuellen Forschungskontext zu berücksichtigen, bedarf es qualitativer Studien (vgl. [Bortz und Döring 1995, S. 271], [Bortz und Döring 2002, Kapitel 5]). Zentrale Elemente der Untersuchungs-gestaltung werden dargestellt, damit Annahmen und Entscheidungen nachvollziehbar werden. Die Darstellung der Studien und ihrer Ergebnisse umfassen einen deskriptiven und einen analytischen Teil.

## 5.1 Untersuchungsgestaltung

### 5.1.1 Ziele der Expertise

Die Studien werden durchgeführt, um die Bandbreite der Anforderungen, Positionen und Einstellungen von Expertinnen im Forschungsgefüge zu berücksichtigen. Es wird nicht angestrebt, eine repräsentative Studie durchzuführen, d. h. durch eine möglichst große Anzahl von Interviews die Gesamtpopulation der Informatiklehrerinnen zu beschreiben. Ziel ist vielmehr, Expertinnenwissen zur Prüfung der bisher gewonnenen Arbeitshypothesen heranzuziehen, bisher nicht berücksichtigte Anforderungen zu identifizieren, die

<sup>215</sup> Details zu den Kriterien des Auswahlverfahrens werden in Abschnitt 5.1.3 dokumentiert.

Spannbreite und die verschiedenen Ausprägungen – aber auch die Gemeinsamkeiten – in der Positionierung gegenüber grundlegenden Konzepten der Schulinformatik zu erkunden. Vor Beginn der Untersuchung wurden Arbeitshypothesen für die Expertinnenbefragung formuliert (vgl. Tabelle 5.1).

Kategorie	Arbeitshypothese
Daten und Fragen zur Person	<ul style="list-style-type: none"> <li>• Informatiklehrerinnen sind primär für die Schulfächer Mathematik und Physik qualifiziert.</li> </ul>
Konzepte des Informatikunterrichts	<ul style="list-style-type: none"> <li>• Informatische Modellierung ist als konstitutiver Bestandteil des Informatikunterrichts allgemein akzeptiert.</li> <li>• Unter den Unterrichtenden gibt es große Übereinstimmung, dass die objektorientierte Modellierung gegenüber der funktionalen und der wissensbasierten Modellierung aus verschiedenen Gründen „zu bevorzugen ist“.</li> <li>• Bestimmte Fachkonzepte (wie z. B. wissensbasierte/prädikative Modellierung, funktionale Modellierung) werden für wichtig erachtet, aber unterrichtlich eher selten umgesetzt.</li> </ul>
Methodisch-didaktische Aspekte	<ul style="list-style-type: none"> <li>• Schemata und Checklisten werden kaum verwendet, Programmablaufpläne (PAP) spielen in der informatischen Bildung keine große Rolle mehr.</li> <li>• Unified Modelling Language (UML) und ihre didaktisch aufbereiteten Teilmengen werden zunehmend eingesetzt.</li> <li>• Es gibt eine Vielzahl von „undokumentierten“ Hilfsmitteln zur Visualisierung abstrakter und komplexer Strukturen.</li> </ul>
[Software-]Technische Unterstützung des Informatikunterrichts	<ul style="list-style-type: none"> <li>• Es gibt keine durchgängig benutzte softwaretechnisch nach didaktischen Gesichtspunkten gestaltete Umgebung, die breiten Einsatz findet.</li> <li>• Der Rückgriff auf Werkzeuge, die für andere Zielgruppen entwickelt werden, führt häufig zu Problemen im Unterricht.</li> <li>• Informatiklehrerinnen möchten unterrichtlich eingesetzte Informatiksysteme an ihre konkreten Lerngruppen anpassen [können].</li> </ul>

Tabelle 5.1: Arbeitshypothesen für die Expertinnenbefragung

### 5.1.2 Zur Auswahl der Forschungsmethode

Mit den begrenzten Ressourcen, die in diesem Forschungsvorhaben zur Verfügung stehen, kann keine breit angelegte quantitative empirische Untersuchung durchgeführt werden. Um die Arbeitshypothesen (vgl. Tabelle 5.1) im aktuellen Forschungskontext prüfen zu können, wird eine qualitative Untersuchung geplant und durchgeführt. Darüber hinaus ermöglicht eine qualitative Studie die Einbeziehung offener Fragestellungen. Die individuelle Sicht der Befragten auf die Forschungsgegenstände findet so Eingang in die Untersuchung. Die besondere Zielsetzung der Interviews besteht darin, an ausgewählten Punkten das Wissen von Expertinnen konstruktiv in die Forschungsarbeit einfließen zu lassen.

Explorative Studien in Form von Interviews stellen ein methodisch ausgereiftes Werkzeug zur Verfügung, das diesen Anforderungen gerecht werden kann. Mit der Möglichkeit, vorgegebene Strukturen zu füllen, aber auch unerwarteten Reaktionen Raum zu geben und diese ggf. zu vertiefen, kann die für den offenen Charakter notwendige Flexibilität erreicht werden. Da es nicht möglich ist, für den explorativen Anteil der Studie geschlossene Fragen zu formulieren, wurde der Gestaltung strukturierender Leitfragen besondere Aufmerksamkeit gewidmet. Impulse als Erzählanstöße sollen den Expertinnen einen direkten Einfluss auf die Gestaltung und die Vertiefung individuell bedeutsamer Bereiche eröffnen.

### 5.1.3 Auswahl der Expertinnen

Als potenzielle Kandidatinnen kommen Personen in Betracht, die

1. als Informatiklehrerinnen tätig sind und
2. Beiträge zur [Weiter-]Entwicklung der Fachdidaktik leisten.<sup>216</sup>

Die aktive Teilnahme an fachdidaktisch ausgerichteten Veranstaltungen<sup>217</sup> stellt ein Auswahlkriterium dar. Darüber hinaus wird die Einhaltung der Kriterien im Zusammenhang mit der Kontaktaufnahme und während der Durchführung der Interviews verifiziert.<sup>218</sup>

Diese Anforderungen schränken den Kreis der in Frage kommenden Lehrerinnen stark ein. Die ausgewählten Informatiklehrerinnen wurden angeschrieben und um einen Terminvorschlag für die Durchführung eines Interviews gebeten (Einladungsschreiben siehe Anhang A.1, S. 161). Als Ergebnis dieser Anfrage wurden 19 Lehrerinnen für die Durchführung eines Interviews gewonnen. Über einen Zeitraum von ungefähr einem Jahr wurden vom Autor 16 Interviews mit Informatiklehrerinnen<sup>219</sup> durchgeführt.

Auch bei dieser kleiner Zahl exponierter Personen sind quantitative Aussagen möglich. Diese Ergebnisse können aber nicht verallgemeinert werden. Studien hierzu stehen noch aus (vergleiche z. B. Abschnitt 4.2.2 – 1996, S. 76).

### 5.1.4 Phasierung der Interviews

<ul style="list-style-type: none"> <li>• Daten und Fragen zur Person biographische Details, mit offenen Fragestellungen</li> </ul>	Abschnitt I
<ul style="list-style-type: none"> <li>• Interviewthesen zum Teil provozierende Aussagen</li> </ul>	Abschnitt II
<ul style="list-style-type: none"> <li>• Grundlegende Konzepte des Informatikunterrichts mit offenen Fragestellungen</li> </ul>	Abschnitt III
<ul style="list-style-type: none"> <li>• Methodisch-didaktische Aspekte Darstellung und Kommunikation von Strukturen mit Hilfe von Schemata</li> </ul>	Abschnitt IV
<ul style="list-style-type: none"> <li>• [Software-]Technische Unterstützung Möglichkeit, Details darzustellen</li> </ul>	Abschnitt V

Tabelle 5.2: Phasierung der Expertinneninterviews

<sup>216</sup> das umfasst: Veröffentlichung von i. d. R. begutachteten Beiträgen anlässlich der INFOS-Tagungen, GI-Jahrestagungen mit Workshops zur Didaktik der Informatik, Teilnahme an den Fachdidaktischen Gesprächen in Königstein (sächsische Schweiz).

<sup>217</sup> sowie Aktivitäten in der Fachgruppe für Informatiklehrerinnen und Informatiklehrer in der GI.

<sup>218</sup> Alle Interviewten stimmten dieser Offenlegung der Auswahlkriterien zu.

<sup>219</sup> aus sieben Bundesländern

Die Grundstruktur der Phasen soll garantieren, dass aussagekräftige Erkenntnisse auch unter engen zeitlichen Rahmenbedingungen gewonnen werden können.

### 5.1.5 Ausgestaltung – Konkretisierung der Fragen

Aus den Arbeitshypothesen (vgl. Tabelle 5.1) werden Fragestellungen abgeleitet, die dazu dienen, die bis auf Abschnitt II offen gehaltenen leitfadengestützten, narrativen Interviews vor zu strukturieren. Um eine gemeinsame Struktur für die Interviews zu finden, wird ein Leitfaden mit offenen Fragen entwickelt, der im Anhang A.2 (S. 162ff) dokumentiert ist. In der Tabelle 5.3 sind die Elemente des Leitfadens dargestellt, die für die qualitative Dimension der narrativen Interviews<sup>220</sup> bedeutsam sind. Den offenen Fragen und ihrer Platzierung innerhalb der Interviews kommt in diesem Zusammenhang eine Schlüsselfunktion zu, da sie eine Struktur ermöglichen, in der neben geschlossenen Fragen immer wieder Elemente treten, die offene Gesprächssituationen ermöglichen. Die offenen Fragen wurden so formuliert, dass ihnen ein Impulscharakter zugeschrieben werden kann.

Kennzeichen	Beschreibung	Verweis (vgl. Anhang A.2, S. 162)
<b>Daten und Fragen zur Person</b>		
	Warum gerade Informatik?	Abschnitt I – 7
	Warum wurde gerade dieser Schultyp gewählt?	Abschnitt I – 10
<b>Interviewthesen</b>		
	keine offenen Fragen	
<b>Gestaltung des Unterrichts – Unterrichtskonzepte für die Sekundarstufe II</b>		
	Welches Konzept vertritt die Interviewte für die Schule (Sekundarstufe II, allgemeinbildend)?	Abschnitt III – 6ff
<b>methodische Hilfsmittel zur Umsetzung des Konzepts</b>		
	Wie setzt die Interviewte ihr Konzept in der Schule um?	Abschnitt IV – 5ff
<b>didaktisch gestaltete [Software-]Technische Unterstützung für den Informatikunterricht</b>		
	Impulse, um die Zielrichtung der Fragestellung zu verdeutlichen. Es bleibt den Expertinnen überlassen, hier geeignete eigene Anforderungen zu formulieren, ihre Erfahrung mit Hilfsmitteln zur Durchführung des Informatikunterrichts darzustellen.	Abschnitt V

Tabelle 5.3: „offene“ Fragen – Zuordnung zu Interviewphasen

Eines der Ziele der Interviewserie besteht in der Erkundung der Gestaltungsprinzipien des Informatikunterrichts in der allgemein bildenden Sekundarstufe II. Hierbei geht es im Wesentlichen um die Konzepte, die der Unterrichtsgestaltung zu Grunde liegen.

#### Leitlinien für die Interviewführung (nach [Bortz und Döring 1995, S. 292])

1. Der Interviewer sollte die Reaktionen der Befragten nicht beeinflussen – die Gesprächsführung sollte nicht direktiv [im Sinne von klaren Verhaltensregeln] sein und es den Befragten ermöglichen, ihre persönliche Interpretation der Situation zu geben.
2. Das Gespräch sollte situationsspezifisch geführt werden, um herauszufinden, welche Bedeutung die befragte Person einzelnen Teilen oder Elementen der untersuchten Situation beimisst.

<sup>220</sup> im Unterschied zu standardisierten Fragebögen

3. Die Gesprächsführung sollte für unerwartete Reaktionen Raum lassen und diese aufgreifen. Die vom Interviewleitfaden abweichenden Gesprächsteile sind besonders geeignet, neue Hypothesen über die Wirkungsweise bzw. die Art, wie die Situation verarbeitet wird, aufzustellen.
4. Das Gespräch sollte „tiefgründig“ geführt werden. Der Interviewer sollte sich bemühen, die Interviewten zu „selbstenthüllenden“ Kommentaren zu ermuntern.

## 5.2 Ergebnisse

Bei den im Folgenden ausgewiesenen prozentualen Ergebnissen ist zu beachten: die Prozentangaben wurden gerundet und sollen eine schnelle quantitative Einordnung ermöglichen. Es handelt sich nicht um eine repräsentative Studie – prozentuale Angaben sollen Tendenzen unter den ausgewählten Informatiklehrerinnen verdeutlichen.

Da zu erwarten steht, dass die Interviews – bezogen auf die jeweiligen Fragestellungen – spezifische Details pointiert beleuchten, werden neben summarischen Ergebnissen ausgewählte Elemente aus den Transkripten übernommen.

### 5.2.1 Daten und Fragen zur Person

Die erhobenen personenbezogenen Daten<sup>221</sup> werden verdichtet. Die Arbeitshypothese, dass ein großer Teil der Expertinnen im Wesentlichen für das Lehramt Mathematik und/oder Physik qualifiziert sind, kann bestätigt werden. Fünf Expertinnen (d. h. fast  $\frac{1}{3}$  der Interviewten) sind primär für das Lehramt Informatik qualifiziert. Überraschenderweise findet sich neben der Qualifikation für das Lehramt bei neun ( $\sim 56\%$ ) Expertinnen eine weitere berufliche Qualifikation. Es konnte keine Hochschule als dominierende Quelle für die Ausbildung von Informatiklehrerinnen identifiziert werden. Das Ergebnis zeigt, dass die Auswahl der Lehrerinnen keine Präferenzen – bezogen auf den Ort der universitären Qualifikation – nach sich zog. Alle interviewten Lehrkräfte lesen die Zeitschrift LOG IN.<sup>222</sup>

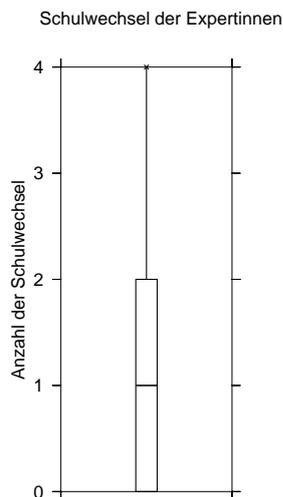


Abbildung 5.2: Anzahl der Schulwechsel

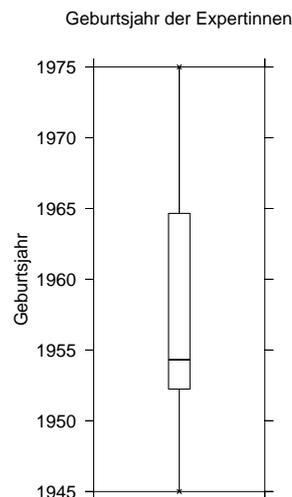


Abbildung 5.3: Expertinnen – Geburtsjahrgänge

<sup>221</sup> Eine tabellarische kumulierte Auswertung findet sich im Anhang A.3.1 (S. 164).

<sup>222</sup> – 38% IBU (Erscheinen inzwischen eingestellt)

– 38% Computer & Unterricht

– 44% c't

Die Anzahl der Schulwechsel nach vollständigem Abschluss der Ausbildung der Informatiklehrerinnen, die interviewt wurden, ist sehr gering ( $\bar{x} = 1$  vgl. Abbildung 5.2<sup>223</sup>). Das mittlere Geburtsjahr der Expertinnen ist 1958 – in der Abbildung 5.3 sind die Geburtsjahrgänge der Expertinnen dargestellt. Damit liegt das durchschnittliche Alter (im Jahr 2002) mit 44 Jahren nur drei Jahre unter dem Altersdurchschnitt der bundesdeutschen Lehrerinnen [Statistisches Bundesamt 2002].

Die Altersstruktur der ausgewählten Expertinnen weicht nicht signifikant von der Altersstruktur der Gesamtpopulation der Lehrerinnen in der Bundesrepublik ab. Obwohl dies nicht explizit bei den Auswahlkriterien Berücksichtigung fand, ist es offenbar gelungen, Expertinnen aus dem gesamten Altersspektrum bei den Interviews zu erreichen.

Die Hälfte der Expertinnen leiten die Fachkonferenz Informatik an ihrer Schule – dieser relativ geringe Anteil hängt damit zusammen, dass die Interviewten häufig außerschulisch dienstliche Aktivitäten<sup>224</sup> entfalten und damit ihrer Schule nicht mit voller Stundenzahl zur Verfügung stehen. Zu den Aktivitäten werden in Tabelle 5.4 quantitative Aussagen ausgewählt, die diese Aussage differenziert stützen.<sup>225</sup>

Aktivitäten der Expertinnen in informatikdidaktischen Zusammenhängen (geordnet nach der Anzahl der Nennungen – die fett gesetzten Aktivitäten werden teilweise durch Abminderungsstunden abgegolten)	
Lokale Initiative(n)	81%
Königstein – fachdidaktische Gespräche	75%
Betreuung eines Webangebots zur Schulinformatik	75%
<b>Mittelbehörde - Lehrerfortbildung IF</b>	<b>63%</b>
<b>KM – aktuelle Lehrplanentwicklung</b>	<b>50%</b>
<b>Bund – Schulen ans Netz</b>	<b>25%</b>

Tabelle 5.4: Informatikdidaktische Aktivitäten der Expertinnen

Deutlich ist die Zurückhaltung bzgl. der Projekte zu *Schulen ans Netz*. Die Ursachen sind vielfältig, z. B. werden in einigen Bundesländern Informatiklehrerinnen ganz bewusst nicht [mehr] in die Umsetzung eingebunden, da sie auf Qualifikationen im Bereich der informatischen Bildung als notwendige Grundlage für die erfolgreiche Arbeit mit Internetdiensten verweisen. Dass 88% der interviewten Informatiklehrerinnen Mitglied in einer der Vereinigungen für Informatiklehrerinnen sind, ist auf Grund der Auswahlkriterien nicht überraschend.

### Wie kamen die Expertinnen zur Informatik?

Im Folgenden werden einige Antworten der Expertinnen auf den Impuls (Abschnitt I, Frage 7) „Warum Informatik?“ wiedergegeben.<sup>226</sup>

<sup>223</sup> Bei den Abbildungen handelt es sich um sogenannte Box-Plots [Tukey 1977]. Dabei repräsentiert der mittlere Balken den Median. Untere und obere Begrenzungen der Kästchen geben das Percentil 25 und 75 wieder. In beiden Fällen sind keine „Ausreißer“ vorhanden, d. h. alle Werte befinden sich innerhalb des Bereichs  $1,5 \times (0,75 - 0,25) = 0,75$  um die Kästchen.

<sup>224</sup> z. B. in der Lehrerfortbildung – siehe die folgenden detaillierten Ausführungen zu diesem Punkt

<sup>225</sup> Die vollständige tabellarische Übersicht findet sich im Anhang A.3.1 (S. 164).

<sup>226</sup>

#### Bemerkungen zu den wiedergegebenen Transkriptausschnitten

Aus Gründen des Persönlichkeitsschutzes werden aus den im Folgenden zitierten Teilen der Transkripte Elemente entfernt, die Rückschlüsse auf direkt oder indirekt Beteiligte zulassen. Darüber hinaus werden – aus Gründen der Lesbarkeit – Wiederholungen (die in gesprochener Sprache nicht weiter stören) und Füllsilben wie „hmm“, „eh“ nicht aus den Transkripten übernommen (vgl. [Bortz und Döring 1995, S. 287]). Des Weiteren werden folgende Regeln bei der Transkription befolgt: es wird keine Großschreibung benutzt, um keine Hervorhebungen zu provozieren; in den Transkripten sollten sich keine Satzzeichen befinden. Sekundengenaue Pausen sind in den Originaltranskripten explizit ausgewiesen, werden aber in den folgenden Ausschnitten nicht angegeben. Die Zitation aus den Transkripten erfolgt nach dem Muster:

[Transkript a, Abschnitt b, c' d', S. e, Z. f–g ]

a  $\cong$  Nummer des Interviews; b  $\in$  {I, ..., V}; c  $\cong$  Minute, d  $\cong$  Sekunde ;

e  $\cong$  Seitennummer im Transkriptband ; f, g  $\cong$  Zeile des jeweiligen Gesamttranskripts

[Transkript 1, Abschnitt I, 2'00", S. 2, Z. 18–24]:

Expertin: also als die beschäftigung die ich als jugendlicher gemacht hab mehr im außerschulischen bereich habe ich mich vor allem mit elektronischen fragestellungen beschäftigt und hatte auch theoretisches interesse und von da aus gesehen war das für mich eigentlich mehr eine konsequente weiterentwicklung so dass und kann ich mich nachhinein feststellen schon bei der berufswahlgeschichte im abitur stand softwareingenieur obwohl ich nicht gewusst hätte dass es den begriff damals schon gegeben hätte also war damals sehr nahe liegend also sehr früh angelegt dass ich irgendwo in diese ecke informatik elektronik e-technik gehen würde

[Transkript 3, Abschnitt I, 3'36", S. 20, Z. 48–59]:

Expertin: ... es war eine situation wir hatten eine wohngemeinschaft und hatten in dieser gemeinschaft einen der hat informatik studiert und hat sehr früh einen pet 2001 ... und dies gerät stand in der wohngemeinschaft zur verfügung und man konnte darauf viele experimente ... veranstalten ... so dass das interesse dann eben auch mit der möglichkeit die dinge auszuprobieren günstig verknüpft war und ich sehr viele experimente gemacht habe so viel dinge probiert habe viel gelernt habe über den computer das sich wesentlich entwickelt hat im schulischen bereich war aber dann die beschäftigung mit der informatik anfangs relativ gering ich habe anfang 82 dann schon erste ags dazu gemacht aber intensiv hat es eingesetzt nachdem ich oder als ich begonnen habe mich beruflich zu qualifizieren und eine lehrerfortbildungsmassnahme absolviert habe

[Transkript 4, Abschnitt I, 4'45", S. 31, Z. 61–76]:

Expertin: macht spass ganz interessant ich kann auch natürlich viele andere themen begründen ... ich denke also sage ich mal zunächst ursprünglich auch schulzeit in informatik gehabt und gesehen dass ist schon spannend einem computer beizubringen etwas zu machen was man möchte also das sind die ursprüngliche motivation ehrlich gesagt ... das hängt sicherlich damit zusammen dass mir logisches denken einigermaßen leicht fällt im vergleich zu anderen also für nichtinformatiker dann und später dann als motivation sicherlich dass ich sehe ... die beeinflussung der gesellschaft durch informatik und so heisst informatiksystem was auch immer sein soll und das ist im effekt zwischen halt schlüsselqualifikation ist und richtung kulturtechnik geht oder sogar ... diese informatiksysteme ganz einfach zu beherrschen

[Transkript 5, Abschnitt I, 3'18", S. 40, Z. 39–45]

Expertin: damals war es das reizte neu einfach probleme zu lösen mit hilfe einer maschine jetzt ist eigentlich mehr in die richtung dass der bedarf da ist vor allem bei den schülern das grundwissen zu vermitteln und damit in der schnellen zeit systematik zu erzeugen die dann hilft später sich selbst wieder zurecht zu finden ... auch wenn nach zwei jahren ... anders aussieht und damals war es eigentlich der reiz des computers da beschäftigung so in die richtung wo ich ausprobiere was ist möglich

[Transkript 6, Abschnitt I, 3'38", S. 48, Z. 53–60]

Expertin: also meine staatsexamensarbeit ging um numerische mathematik ... so eigenwertprobleme von matrizen ... und also meine erste programmiererfahrung habe ich mit einer ... erworben und da habe ich eben die sachen die ich als teil des staatsexamens geschrieben habe entsprechend auch die algorithmen da programmiert und sofern war alles eigentlich klar ich bin über die mathematik reingekommen ... also die praktische mathematik

[Transkript 7, Abschnitt I, 6'54", S. 58, Z. 107–120]:

Expertin: weil ich das für eins der ... typischen schlüsselprobleme halte das ist die eine seite ... daher komme ich eigentlich von der berufsschule weil ich es für die berufsausbildung für unerlässlich halte dass es in der berufsschule unterrichtet wird und dadurch bin ich in die allgemeinbildung reingerutscht weil in der berufsbildung hatten die schwierigkeiten mit einer solchen informatik das ist ihnen ja nicht praktisch genug und wie man das immer nennen soll keine handlungsanweisung und da hatte ich schwierigkeiten und dann bin ich zur allgemeinbildung gewechselt ja und dann persönliches interesse schon seit der studienzeit da habe ich

– habe jetzt gerade festgestellt – zu meiner grossen überraschung mit einer funktionalen programmiersprache angefangen habe weil ich apl gehabt habe dass war damals das einzige was die wirtschaftswissenschaftler als programmiersprache angeboten haben und ich habe schon in meiner diplomarbeit da mit einem grossen kasten mit lochkarten drin und befragte die studenten und ich habe gesagt können die einlesen und dann habe ich die mit apl auseinandersortiert mit der grossen matrix ... und dann untersucht ob das schulpraktikum eine auswirkung auf die motivation der studenten hat einfach die studenten eingeteilt vor dem praktikum nach dem praktikum ...

---

[Transkript 8, Abschnitt I, 1'31", S. 67, Z. 22–28]:

Expertin: das hat sich ergeben ... ich habe das war untypisch glaube ich ich habe mathematische trickfilme gemacht als visualisierung von mathematik bei der gelegenheit sind wir dann übergegangen von der alten tricktechnik die wir vorher geändert haben für komplizierte szenen ... computergraphik einzusetzen und zwar damals noch auf papier geplottet ausgeschnitten und auf folien geklebt und verfilmt

---

[Transkript 9, Abschnitt I, 1'41", S. 78, Z. 21–26]:

Expertin: nach meinem ersten examen waren die einstellungschancen sehr schlecht und ich dachte ich muss einen zweiten beruf erlernen und bin ich auf die informatik gekommen habe mich dann in der fernuni in hagen eingeschrieben für informatik diplomstudiengang und hab dann informatik studiert

Interviewer: an der fernuni neben deiner anderen tätigkeit

Expertin: genau

---

[Transkript 10, Abschnitt I, 2'20", S. 98, Z. 34–38]:

Expertin: weil ich freude an problemlösungen habe – weil es reizvoll ist – einfach ein problem zu knacken und dann es mit der maschine zu bearbeiten das heißt also man kriegt erfolgsmeldung dadurch dass einem das gelungen ist und man kann es lösen durch logisches nachdenken das heißt die freude ist etwas vermindert wenn ich etwa erst mal problem zu tun habe das ich im logik nicht im griff kommen lassen zum beispiel probleme mit bestimmten betriebssystemen

---

[Transkript 11, Abschnitt I, 2'00", S. 106, Z. 25–34]:

Expertin: es hat mich einfach interessiert ... ich komme ja aus der mathematik und habe kein naturwissenschaftliches fach wie physik weil ich von einer Mädchenschule kam – physik hat mich sehr interessiert während der schule aber die anderen mädchen halt nicht – also habe ich die letzten zwei jahren nicht gemacht – aber ich habe mich für politische themen interessiert – trotzdem hatte ich so ... von der mathematik endlich mal was festes dann hab ich viel gerechnet am anfang – ich habe fortran gemacht und da hast du vor allen gerechnet erst mal

Interviewer: du hast dir die arbeit damit erleichtert aus dem bereich der mathematik

---

[Transkript 12, Abschnitt I, 2'10", S. 117, Z. 36–37]:

Expertin: weil mich die möglichkeiten der maschinen oder der systeme fasziniert haben mit allen konsequenzen auch für die schüler

---

[Transkript 13, Abschnitt I, 3'31", S. 126, Z. 54–55]:

Expertin: <lacht> nee damals interesse gründe also von anfang an zu studieren war gerade rechner kam überhaupt erst in den einzugsbereich einer studentin und mich reizt diese technik damals.

---

[Transkript 14, Abschnitt I, 2'01", S. 135f, Z. 29–38]:

Expertin: hmm ich wollte gern mathematik studieren und habe ich informatik erst mal gewählt weil ich zweites fach brauchte und weil ich damit in der schule klar mitgekommen bin ... und weil es halt in der nähe zur mathematik hatte und ich mir ganz etwas einfacher machen wollte

anstatt geschichte zu wählen was ich mir auch überlegt hatte ... ich habe aber das nicht bereut ... ich habe eigentlich während des studiums informatik lieben gelernt ... und der prozess geht eigentlich immer noch weiter dass ich eh immer interessanter finde

---

[Transkript 15, Abschnitt I, 3'01", S. 146, Z. 45–50]:

Expertin: ja ich habe mich damals für informatik entschieden weil ich lehrerin werden wollte ich brauchte ein zweites fach hmm und da kam für mich eventuell physik in frage und fand ich die informatik nicht interessanter aber irgendwie hat mich die mehr gereizt ... das war keine so fest überlegung dass ich sage dies möchte ich unbedingt machen sondern das war für mich so eine überlegung was willst du unterrichten

---

[Transkript 16, Abschnitt I, 4'01", S. 160, Z. 40–48]:

Expertin: das kann ich relativ klar sagen em es war bei mir so dass ich in der oberstufe schon festgestellt habe dass ich eine gewisse vorliebe für naturwissenschaftliche und mathematisch orientierte fragestellungen hatte em eigentlich hatte ich grosse vorlieben für mathematik und physik ich hatte mich allerdings dann später immer wieder selbstkritisch gefragt was man damit tun kann und so bin auch dann eigentlich zunehmend in die richtung gerutscht dass ich also da schon festgestellt habe dass ich da schon trotzdem eher in diese informatikorientierte richtung dann dort hineingelangen wird und ich hatte informatik auch als grundkurs belegt und habe mich dann gefragt warum eigentlich nicht direkt das original studieren wenn man ansonsten ne kopie studiert und sich dann später dann doch mit informatischen fragestellungen auseinander setzt

Die individuellen Zugänge zur Informatik sind äußerst vielfältig. Der Beginn der individuellen Auseinandersetzung mit informatischen Fragestellungen erfolgt zum Teil über konkrete Problemstellungen, die mit Hilfe von Informatiksystemen bearbeitet wurden, zu einem anderen Teil über experimentelle Zugänge zu Informatiksystemen aber auch durch die unterrichtliche Umsetzung erster hardwareorientierter Ansätze.

Es wird deutlich, dass der gesellschaftliche Stellenwert der Informatik (Stichwort: Schlüsselqualifikation) die primäre Entscheidung für das Lehramt für das Schulfach Informatik offensichtlich kaum beeinflusst hat.

### **Warum wurde gerade dieser Schultyp gewählt?**

Die offene Frage (Abschnitt I, Frage 10) erwies sich nicht als geeignet, mit den Expertinnen in eine offene Gesprächssituation einzutreten. Für den Großteil der Expertinnen ist der Lehrerberuf durch die eigene Schulausbildung am Gymnasium geprägt, so dass sie der Frage nach einer anderen Schulformen eher ablehnend gegenüberstanden. Da ihre eigene Ausbildung typischerweise durch das Gymnasium geprägt wurde, kommt [bei den interviewten Expertinnen] bei der Entscheidung für den Lehrerberuf primär das gymnasiale Lehramt in Betracht.

---

[Transkript 3, Abschnitt I, 6'08", S. 21, Z. 82–86]:

Interviewer: warum ist das gerade gymnasium geworden wie ist das entstanden

Expertin: ja ich war auf dem gymnasium und mich hat immer das fach oder die fächer sehr interessiert auch die mathematik die hat mich sehr fasziniert eigentlich und von daher war es eigentlich nie ein thema weil ich immer der auffassung bin dass natürlich das fachliche im gymnasium einen viel höheren stellenwert hat als in anderen schulformen

---

Die Nützlichkeit des Zeitschriftenstudiums wird von einer Expertin sehr kritisch gesehen (Item 16); dies wird exemplarisch durch eine Antwort zu den Fragen nach den gelesenen Zeitschriften illustriert.

---

[Transkript 4, Abschnitt I, 9'00", S. 32, Z. 117–121]:

Expertin: log in habe ich gelesen ... weil ich aus der log in erstens nicht viel rausziehen kann zweitens auch nicht die zeit finde ... ich muss zuerst rausziehen was für mich drinsteckt dann soll ich das runterbrechen auf meinen unterricht dann mache ich lieber den kurzen weg ... dann mache ich gleich meinen richtigen unterricht

---

Diese Kritik kann als Indiz verstanden werden, dass es der Zeitschrift LOG IN offenbar bisher nicht immer gelungen zu sein scheint, eine der informatisch hochqualifizierten Lehrerinnen mit Material so zu versorgen, dass die fachliche Ebene mit der fachdidaktischen Gestaltung übertragbar verschränkt wird.

## 5.2.2 Konzepte des Informatikunterrichts

### Arbeitshypothesen

- Informatische Modellierung ist als konstitutiver Bestandteil des Informatikunterrichts allgemein akzeptiert.
- Unter den Unterrichtenden gibt es große Übereinstimmung, dass die objektorientierte Modellierung gegenüber der funktionalen und der wissensbasierten Modellierung aus verschiedenen Gründen „zu bevorzugen ist“.
- Bestimmte Fachkonzepte (wie z. B. prädikative Modellierung, funktionale Modellierung) werden für wichtig erachtet, aber unterrichtlich eher selten umgesetzt.

Die Position der Expertinnen zu dem Bereich der Schwerpunktsetzung in der informatischen Modellierung fand als Element im Gesprächsleitfaden (siehe Anhang A.2, S. 162ff) Berücksichtigung. Die Ergebnisse bezüglich der Konzepte des Informatikunterrichts werden im Folgenden dokumentiert, um die formulierten Arbeitshypothesen zu prüfen.

### Informatische Modellierung als zentrales Konzept

#### Ausprägung: objektorientiert, funktional, wissensbasiert

Informatische Modellierung wird als zentrales Konzept implizit von allen interviewten Expertinnen bestätigt. Drei der befragten Expertinnen fordern explizit die Aufnahme der Modellierung als zu thematisierendes übergreifendes Konzept, dem sich Ausprägungen unterordnen sollen.

---

[Transkript 7, Abschnitt III, 21'10", S. 65, Z. 332–342]:

Expertin: also ich denke diese ganze ebene der modellierung – ja du hast die drei genannt ... die würde ich sagen das ist sehr wichtig ... extra ausweisen ... also modellierung ... würde ich auch als einen der wichtigen punkte sehen

---

[Transkript 14, Abschnitt III, 18'22", S. 141, Z. 263–266]:

Expertin: ja ich weiß jetzt nicht ob das dazugehört aber halt solche sachen wie modellierung und projektorientiertes arbeiten einfach diese arbeitsweisen die meiner meinung nach ... ja das hängt ja mit der informatik zusammen

---

[Transkript 16, Abschnitt III, 17', S. 164, Z. 233]:

Expertin: modellierungstechniken wär vielleicht noch ein interessantes schlagwort

Es ist darüber hinaus festzustellen, dass der funktionalen Modellierung aus Sicht des Großteils der Expertinnen eine nachgelagerte Rolle zukommt, während der objektorientierten eine Schlüsselrolle zugesprochen wird. Diejenigen, die sich für eine stärkere Berücksichtigung der funktionalen Modellierung im unterrichtlichen Kontext einsetzen, sprechen sich regelrecht begeistert für diese Modellierung aus.<sup>227</sup> Andere weisen darauf hin, dass die Mathematiklastigkeit der ihnen bekannten Beispiele zur funktionalen Modellierung diese Modellierung als ungeeignet für eine stärkere Berücksichtigung als konzeptionelle Grundlage des Informatikunterrichts erscheinen lassen. Es muss darüber hinaus festgestellt werden, dass ein überwiegender Teil der Expertinnen bisher keine unterrichtliche Umsetzung der funktionalen Modellierung erprobt haben.

---

[Transkript 3, Abschnitt III, 21'21", S. 23, Z. 173]:

Expertin: ich sag jetzt eher nein keine erfahrung keine basis nie was gemacht mit

---

[Transkript 6, Abschnitt III, 18'10", S. 52, Z. 279–282]:

Expertin: hach na ja also eher nein also nein würde ich nicht sagen aber weil ... ist eher ein randthema denk ich mal ja

---

[Transkript 8, Abschnitt IV, 14'55", S. 72, Z. 225–232]:

Expertin: also ... aus der mathematikerfahrung heraus für solche kalkülhaften vorgehen ... also ich möchte nicht dass sozusagen ... die informatik kann an bestimmten stellen mathematik ersetzen ... aber das muss besser sein und das heisst gegen diesen kalkülhaften ansatz einfach term also endlose term umformung zu machen so zu sagen ab klasse sieben mathematik unendliche termumformung ...

---

[Transkript 9, Abschnitt III, 18'02", S. 83, Z. 226–228]:

Expertin: damit habe ich nie was gemacht

Interviewer: würdest also sagen für dich nein

Expertin: ja

### **Informatische Modellierung im Kontext**

Einige Expertinnen machen in ihren Ausführungen deutlich, dass die Verkürzung auf eine Art der informatischen Modellierung im Informatikunterricht vermieden werden soll. Vielmehr ist es notwendig, von einem Anwendungskontext auszugehen.

---

[Transkript 4, Abschnitt III, 17'50", S. 35f, Z. 243–249]:

Expertin: ich weiß nicht, ob das in die anderen rein passt ich nenns mal anwendungsbereiche ich kann zum beispiel mal kryptologie machen beim nächsten mal mache ich dann chipkarten public key systeme oder sonst wie, wo so etwas mit reinkommt ... anwendungskontext den ich in irgendeiner form also exemplarisch dann mal behandle ... wo ich sage ... nehmen wir auch mal in einer abgewandelten form die dann zu einer struktur verwendet werden

---

<sup>227</sup> Eine Expertin möchte der funktionalen Modellierung eine gewichtigere Rolle als der objektorientierten zukommen lassen:

[Transkript 8, Abschnitt IV, 10'05", S. 71, Z. 163–166]:

Interviewer: funktionale modellierung

Expertin: ja sehr ja also müsstest du bei den anderen also wenn dann müsstest du bei den anderen immer

Interviewer: eher mehr

Expertin: ja machen und hier ja also das ist ja auch ganz wenig worüber berichtet wird

In der Umsetzung komme es dabei weniger auf Vollständigkeit (vor allem bezogen auf die Implementierung) als vielmehr auf das Wissen um alternative Modellierungsmöglichkeiten an, so die Expertinnen. Folgerichtig werden allgemeine Prinzipien der informatischen Modellierung von einigen Lehrerinnen explizit als zentrales – übergreifendes – Konzept eines modernen Informatikunterrichts benannt. In diesem Kontext sind deutliche Hinweise auf Algorithmen und Datenstrukturen zu sehen, die konzeptionell eine zentrale Bedeutung im unterrichtlichen Zusammenhang haben. Damit wird ein zentrales Problem des aktuellen Informatikunterrichts deutlich: im Kontext der informatischen Modellierung kommt der Erarbeitung ausgewählter Elemente aus Algorithmen und Datenstrukturen<sup>228</sup> eine wichtige Funktion zu. Bei dem beschränkten Zeitbudget und mangelnder Vorkenntnisse der Schülerinnen stehen alle Informatiklehrerinnen vor dem Problem, diesen Anspruch – ohne fachliche Verkürzungen – exemplarisch einzulösen.

Zu den im Interview (vgl. Interviewleitfaden Anhang A.2, S. 162) vorgegebenen Konzepten liegen von allen Expertinnen Positionen vor, die in der Tabelle 5.5 dargestellt sind. Aus den angegebenen Zahlen können folgende Schlüsse gezogen werden: Objektorientierung sowie Algorithmen und Datenstrukturen sind nach Auffassung der Expertinnen Basisbereiche der Schulinformatik, denen ungeteilte Zustimmung durch die Expertinnen zukommt. Der wissensbasierten und der funktionalen Modellierung hingegen schreiben einige der Expertinnen eine eher nachgeordnete Wichtigkeit zu – wobei die funktionale Modellierung (unter anderem mit den bereits oben dokumentierten Kommentaren) erheblich deutlicher abgelehnt wird. Dass auch theoretisch orientierter Informatikunterricht als konzeptionelle Basis für die allgemein bildende Schule umzusetzen sei, wird nur von einem Teil der Expertinnen befürwortet.<sup>229</sup>

Die Arbeitshypothese „Informatische Modellierung als konstitutiver Bestandteil des Informatikunterrichts“ wird nicht abgelehnt, da alle Expertinnen zumindest einer der drei explizierten Modellierungen zugestimmt haben. Die Akzeptanz der Objektorientierung ist ungeteilt; somit fallen sowohl die wissensbasierte aber erst recht die funktionale Modellierung deutlich zurück.

Basis	nein	eher weniger	eher mehr	ja
Objektorientierung	-	-	5	11
Algorithmen und Datenstrukturen	-	-	9	7
funktionale Modellierung	3	9	2	2
wissensbasierte Modellierung	-	7	7	2
theoretisch orientierte Ausrichtung	2	5	5	4

Tabelle 5.5: Auswertung zu vorgegebenen Orientierungen

## Weitere konzeptionelle Orientierungen

Eine Zielstellung der Interviews besteht darin, den Erfahrungsberichten der Expertinnen zu entnehmen, welche über die vorstrukturierten und benannten Elemente hinausreichenden konzeptionellen Grundlagen für den erfolgreichen Informatikunterricht dargestellt wurden. Alle Expertinnen weisen darauf hin, dass über die im Leitfaden ausgewiesenen Elemente hinaus weitere konzeptionelle Elemente im Informatikunterricht Berücksichtigung finden müssen.

Dazu finden sich auf Grund der offenen Fragen des Leitfadens viele Hinweise in den Interviews, die an dieser Stelle zu qualitativen Aussagen verdichtet werden. Es werden im Folgenden zentrale Elemente ausgewiesen, die im Laufe der Interviewserie von den Expertinnen genannt werden. Die Verschränkung der inhaltlichen Ebene über die Thematisierung eines größeren Zusammenhangs ist nach Auffassung der Expertinnen eine originäre Aufgabe des Informatikunterrichts. Hiermit wird das Berufsverständnis der Expertinnen deutlich, nämlich die Einbeziehung der fachlichen Konzepte in übergreifende Zusammenhänge. Im unterrichtlichen Kontext sollen sowohl die gesellschaftlichen Implikationen der Informatik, wie auch die

<sup>228</sup> nach wie vor

<sup>229</sup> Die große Varianz ist m. E. auf die Interpretationsmöglichkeiten des Anspruchs, der mit der theoretischen Informatik verbunden ist, zurückzuführen.

Bereich	nein	eher weniger	eher mehr	ja
Informatik und Gesellschaft				5
Anwendungsbereiche				4
informatisches Modellieren				3
Rechnernetze und verteilte Systeme (RvS)			2	2
Projektarbeit				2
Datenschutz			1	1
parallele Prozesse			2	
Computersicherheit				1
Geschichte der Informatik				1
Berufsbilder				1
Protokolle und Interaktivität			1	
Berechenbarkeit und Komplexität			1	
Grenzen der Informatik			1	
Teamarbeit			1	
technische Informatik / Wirkprinzipien	3	2	3	2
programmiersprachenorientierter Unterricht	1			
mathematisch orientierter Informatikunterricht	1			

Tabelle 5.6: Auswertung zur fachlichen Basis

geschichtliche Dimension konkreter Fachkonzepte in den Blick genommen werden. Bereiche, die die besonderen Arbeitsweisen innerhalb der Fachwissenschaft Informatik betreffen, werden von den Expertinnen ebenfalls als zentrale konzeptionelle Grundlage des Informatikunterrichts in der Sekundarstufe bezeichnet – dies betrifft sowohl die Projektorientierung als informatiktypische Arbeitsweise, wie auch die für die Teamarbeit notwendige Gestaltung des Unterrichtsprozesses. Es wird deutlich, dass die mit der Fachsystematik einhergehenden Arbeitsweisen, Sichten und Auswirkungen der Informatik für die Lehrerinnen zu den konzeptionell grundlegenden Elementen gerechnet werden. So wird beispielsweise eine problemorientierte oder auch anwendungsorientierte<sup>230</sup> Zugangsweise höherwertiger als bestimmte Fachkonzepte (und damit auch die Modellierung) angesehen. Damit stellt sich die Frage nach dem Primat der Fachdidaktik.

Beispiel für eine wiederholt genannte inhaltliche Erweiterung der als grundlegend erkannten konzeptionellen Grundlagen: Vernetzte Strukturen machen die Notwendigkeit erweiterter Anforderungen an Informatiksysteme deutlich. Einige der Expertinnen stimmen darin überein, dass den besonderen Fachkonzepten aus *Rechnernetze und verteilte Systeme (RvS)* eine wichtige Rolle im Kontext der informatischen Modellierung zukommt, die für die unterrichtliche Gestaltung Konsequenzen hat.

Bei der Einschätzung zu *Technische Informatik, Wirkprinzipien* werden Unterschiede in der Einschätzung der Expertinnen deutlich. Eine hardwarenahe Orientierung wird i. d. R. zurückgewiesen, informatische Wirkprinzipien hingegen durchaus als generischer Bestandteil eines Informatikunterrichts angegeben. Programmiersprachenorientierter Unterricht und mathematisch orientierter Informatikunterricht wurden von jeweils einem Experten expliziert und als „abzulehnen“ gekennzeichnet. Quantifizierende Detailangaben sind in gewichteter Reihenfolge in der Tabelle 5.6 aufgelistet.<sup>231</sup>

Um der Vielzahl der in den Interviews verdeutlichten Hinweise gerecht zu werden, ist eine Anschlussuntersuchung angezeigt. Alle erwähnten Elemente sollten dabei als Items einer Bewertung zugänglich gemacht werden.

Zusammenfassend ist festzustellen, dass die Arbeitshypothesen sich als tragfähige Annahmen zur Beschreibung der Positionen von ausgewiesenen Expertinnen des Informatikunterrichts erwiesen. Darüber hinaus

<sup>230</sup> um kein Missverständnis aufkommen zu lassen: Hiermit wird die Variante bezeichnet, von konkreten Anwendungen ausgehend informatisch zu modellieren.

<sup>231</sup> Im Anhang A.3.3 (S. 165) sind die Ergebnisse der Exploration kumuliert dargestellt.

weisen die vielen grundlegenden und detailreichen Anmerkungen der Interviewten auf die Notwendigkeit einer Reformulierung der Aufgaben und Schwerpunkte einer Didaktik der Informatik, die als Bindeglied zwischen der Fachwissenschaft und den Anforderungen an einen modernen Informatikunterricht notwendiger ist denn je.

Über die Hypothesen hinausgehend muss festgestellt werden, dass die Vielgestaltigkeit der als notwendig erkannten konzeptionellen Grundlagen an eine Grenze stößt, die i. W. durch das schmale Stundenkontingent und die mangelhaften Vorkenntnisse der Schülerinnen determiniert ist.

### 5.2.3 Methodisch-didaktische Aspekte

#### Arbeitshypothesen

- Schemata und Checklisten werden kaum verwendet, Programmablaufpläne (PAP) spielen in der informatischen Bildung keine große Rolle mehr.
- Unified Modelling Language (UML) und ihre didaktisch aufbereiteten Teilmengen werden zunehmend eingesetzt.
- Es gibt eine Vielzahl von undokumentierten Hilfsmitteln zur Visualisierung abstrakter und komplexer Strukturen.

Damit eine zielgerichtete Untersuchung in diesem Feld möglich ist, wurde in den Interviews ein Ausschnitt gewählt, der insbesondere auf die Visualisierung konzeptioneller Elemente abhebt. Diese Einschränkung verweist auf die Notwendigkeit, abstrakte Konzepte zu kommunizieren. Im Laufe der Entwicklung der Wissenschaft wurden immer wieder Standards bezüglich der Darstellung dynamischer und statischer Strukturen festgeschrieben, die auch Eingang in die Schulinformatik gefunden haben. Dabei kommt den Expertinnen eine nicht zu unterschätzende Aufgabe zu. Die Standards werden für professionell arbeitende Informatiker geschaffen und sind typischerweise nicht dafür vorgesehen, Novizen im Lernprozess zu unterstützen. Andererseits besteht der (berechtigte) Wunsch, die von der Fachwissenschaft als sinnvoll und nützlich ausgewiesene standardisierte Notationen im Unterricht zu verwenden. Dies ermöglicht die Kommunikation über den individuellen Unterrichtsprozess hinaus und gestattet es auf diese Weise, übergreifende Ansätze fachlich korrekt zu diskutieren.

---

[Transkript 3, Abschnitt IV, 29'02", S. 25, Z. 283–285]:

Expertin: also prolog bei der karte die haben vierports

Interviewer: ach diese

Expertin: exit ... und damit man

---

[Transkript 4, Abschnitt III, 18'05", S. 36, Z. 262–265]:

Expertin: das sehe ich auch also ideen sind da so geschichten wie diese crc-karten mit diesen magneten an die tafel und dann pfeile und linien zu ziehen und wieder wegwischen zu können das hat sich diesmal als für die schüler als sehr gut erwiesen also das man gemeinsam um die tafel sitzt und dann die modellierung gestaltet

---

[Transkript 10, Abschnitt III, 18'35", S. 103, Z. 244–253]:

Expertin: etwa bei der funktionalen modellierung da setze ich solche diagramme ein ... wo man sieht in welcher ebene welche funktion aufgerufen wird mit welchem parameter dass man die rekursionstiefe sehen kann und da wieder hoch wie man rauskommt ... helfen um rekursion zu visualisieren ... auch in dem zusammenhang arbeite ich mit fotokopien ... wo es darum geht die rekursion zu verstehen dann fertige ich ganze viele schnitzel von der funktion und lass die schüler sie in ein diagramm kleben und jeweils die aktuellen werte daran schreiben

---

[Transkript 14, Abschnitt III, 24'15", S. 143, Z. 355–368]:

Expertin: meinst du so sachen die man auch so anfassen kann ... ja was was nehmen wir an also bei den ereignissen habe ich zum beispiel den roboter einfach (in der elf) noch mal benutzt die roboter um ereignisse einzuführen wobei ich das anschaulich fand und damit gute erfahrung in der zehn gemacht hatte ... ich nehme auch immer meine diskette mit ... die auseinander nimmt und so eine scheibe von der festplatte eh halt ja den computer selber reingeguckt ... so parameterübergabe habe ich im meinen kasten gebaut wo man die aktuellen parameter dann reinsteckt ... der kasten der prozedur oder die methode hmm darstellt

Völlig überraschend ist<sup>232</sup> das Ergebnis, dass der Einsatz von Programmablaufplänen immerhin von 44% der befragten Expertinnen nach wie vor befürwortet wird. Argumente, die für PAPs ins Feld geführt werden: für einfache Algorithmen schnell zu überblicken, leicht zu ändern, auch für Schülerinnen in der Sekundarstufe I leicht zugänglich. Der Einsatz von Struktogrammen findet mit 63% eine erheblich höhere Zustimmung, wird aber immerhin von 31% abgelehnt. Die Ablehnung betrifft i. W. Probleme, die mit der Unflexibilität bzgl. nachträglicher Änderungen begründet werden. Zur objektorientierten Modellierung findet UML (inkl. didaktisch gestalteter Teilmengen) bei 69% der Expertinnen Zustimmung für den unterrichtlichen Einsatz.

Damit kann – für diese Population – die Hypothese über die Programmablaufpläne (PAP) nicht aufrecht erhalten werden. Schemata und Checklisten werden nach Analyse der Transkripte tatsächlich kaum eingesetzt, da nach Meinung einiger der Expertinnen die Gefahr besteht, ein schematisches Umgehen (Abhaken) der Schülerinnen mit Problemstellungen auf diese Art zu befördern und so problemlösendes Herangehen zu konterkarieren. Der Einsatz von UML (und Derivaten) wird zu großen Teilen als lernförderlich charakterisiert. Die Eigenentwicklung von Hilfsmitteln zur Visualisierung komplexer Strukturen ist nicht verbreitet, wird aber von einigen Expertinnen produktiv vorangetrieben.

## 5.2.4 [Software-]Technische Unterstützung des Informatikunterrichts

### Arbeitshypothesen

- Es gibt keine durchgängig benutzte softwaretechnisch nach didaktischen Gesichtspunkten gestaltete Umgebung, die breiten Einsatz findet.
- Der Rückgriff auf Werkzeuge, die für andere Zielgruppen entwickelt werden, führt häufig zu Problemen im Unterricht.
- Informatiklehrerinnen möchten unterrichtlich eingesetzte Informatiksysteme an ihre konkreten Lerngruppen anpassen [können].

Die Nutzung professioneller Werkzeuge zu Unterrichtszwecken in anderen Fachkontexten ist unüblich. Im Schulfach Informatik hingegen wird häufig auf Werkzeuge zurückgegriffen, die nicht speziell für den Unterricht angepasst wurden.

Lernprozesse stellen besondere Anforderungen an die Softwareentwicklung. In diesem Kontext soll die Frage diskutiert werden, ob es didaktisch umsetzbare Anforderungen an Werkzeuge gibt, die zu Gestaltungsvorschlägen verdichtet, Eingang in einen didaktisch reflektierten Gebrauch von Werkzeugen finden (können oder sollen).

Die folgenden Transkriptausschnitte werfen einen differenzierten Blick auf die Gesamtsituation. Die Expertinnen haben sehr unterschiedliche Strategien entwickelt, um mit der Situation im Unterrichtszusammenhang produktiv umgehen zu können.

<sup>232</sup> für den Autor

---

[Transkript 2, Abschnitt V, 36'00", S. 16f, Z. 387–400 ]

Expertin: also wenn die den entwurf also dann haben erstmal dann schon visualisiert haben auch teilweise algorithmus wie das geht dann ist für sie fehlerfrei einzutippen das klappt nicht mehr – sie tippen was ein sie machen irgendwo nachher ... schluss wenn alles zusammengesetzt ist ... finde ich keinen studierenden der das fertig macht das ist mein job ... ich kann ihnen hinterher erzählen da und da waren die fehler ich kann auch vorher sagen aber die zeit die das benötigen würde um das fehlerfrei zu machen die ist auch nicht mehr da also die auch nicht mehr bereit im kurs zu investieren ... eigentlich von den studierenden auch nicht gewünscht die wollen eine idee davon haben wie das geht aber das muss nachher nicht lauffähig sein aber sie möchten schon wissen wie es aussieht wenn es läuft

---

[Transkript 3, Abschnitt IV, 29'12", S. 26, Z. 286–313]:

Interviewer: sonst noch was was vielleicht wirklich außerhalb der kodifizierten form liegt irgendwelche unterstützung mittel die im bereich der visualisierung liegen

Expertin: ... in bezug auf wissensbasierte modellierung da nutze ich noch ein allgemein terminvisualisierer mit dem man prologterm darstellen kann ... hier geht's um datenstrukturen dargestellt für die für die arbeitsweise haben wir diese werkzeuge ... aber für die datenstrukturen ist halt ein werkzeug das habe ich entwickelt mit dem man die datenstruktur in inhaltlicher weise darstellen kann

Interviewer: das heißt das hast du entwickelt und das steht dann kolleginnen und kollegen möglicherweise zur verfügung die das bearbeiten sollen

Expertin: ja das steht auf dem bildungsserver zur verfügung

Interviewer: das ist gut dass der hinweis kommt weil damit hat man eine quelle des materials für kolleginnen und kollegen ich weiß dass du auch einen editor in zusammenhang mit objektorientierung entwickelt hast

Expertin: ja ja aber das ist nur ein entwicklungssystem halt

Interviewer: da geht's um entwicklung also einfacher editor das interpretieren ermöglicht oder wie sieht das aus

Expertin: das kommt ja wenn man ein entwicklungssystem eben hat ziemlich nackt her auf die konsole kann man damit schon arbeiten das ist ... und deswegen die entwicklung dieses javaeditors der im unterschied zu entwicklungsumgebungen wie sie von firmen hergestellt werden wie mit power... oder auch forte ... von sun einfach zu überladen sind halt

---

[Transkript 4, Abschnitt IV, 26'01", S. 38, Z. 346–368]:

Expertin: also mit entwicklungsumgebungen sage ich mal das ist mir hundertprozentig klar deswegen erzähle ich ein bisschen ... ich musste ja in der einföhrung mit meinem jetzigen zwölfte letzten jahr mit delphi anfangen dass ist auch entwicklungsumgebung weiß eben nicht ob es jetzt für dich drunter fällt oder nicht aber zumindestens ist quasi professionell ... und da muss ich sagen habe ich meine schwierigkeit damit oder habe ich bisschen negative erfahrung gemacht ein kollege hat auch damit nach einem jahr damit aufgehört wieder macht jetzt zu diesem diesjährigen elfern komplett was anderes weil er sagt das ist zu komplex für die schüler und deswegen würde ich das schon übertragen allgemein auf professionelle entwicklungsumgebungen natürlich ganz nett insbesondere für die schüler ist es ganz schön ein kleines windowsprogramm zu haben was sie schnell auf einer diskette speichern können und zu hause vorführen können das kann ich mit vielen anderen entwicklungsprogrammiersprachen schlechter machen oder gar nicht ... das negative das eben dass den schüler die möglichkeiten der entwicklungsumgebung zunächst erschlägt vor allem im unterricht und ich dann die vielen elemente gar nicht auffassen kann die ich nicht verstehe und dann wieder der aspekt kaufen muss und hinterher ... dass ich da mehr nachteile als vorteile sehe

Interviewer: würdest du denn da forderung stellen an die didaktik würdest du sagen die didaktik muss zur verfügung stellen ... das wäre eigentlich schon eine frage an die ...

Expertin: wenn ich jetzt die andere geschichte sehe wie diese klassenbibliothek stifte und mäuse dann sage ich das ist genau das richtige weil ich sage ich kann die prinzipien verfolgen aber für die schüler wurde es runtergebrochen auf handhabbares niveau wo ich klassen verwende von denen ich gar weiß ... wie die funktionieren aber wie ich sie verwende ist einfach

---

[Transkript 7, Abschnitt V, 26'27", S. 66, Z. 413–430]:

Expertin: also ich finde man braucht entwicklungs Umgebung weil man also eh mit der ... oberfläche und mit früher cc dann dann die einzelne klasse dann kriegst käme mit dem ... problem klar aber gerade an der schule verhältnismäßig viele durchschnittliche und unterdurchschnittliche schüler und die brauchen ja ich würde fast sagen auch heute so sozialisiert sind die brauchen irgend was wo sie dran festhalten und die brauchen irgend hilfe bei der entwicklungs Umgebung ob die kommerziellen entwicklungs Umgebung das gelbe vom ei sind, wage ich zu bezweifeln also wenn ich gucke ... wir machen mit delphi ich habe aber schon ... benutzt ... das ist alles für profis und da ist verhältnismäßig viel aufwand nötig um die schüler darauf zu programmieren auf der anderen seite ist was du sagst da ... wenn ich es selbst stricke und dann irgendwie zwei jahre später fällt derjenige weg der mir das schon gestrickt hat und dann das läuft nicht mehr auf der maschine x sehe ich auch alt aus ... also das ist verhältnismäßig schwierig finde ich und ... auf der anderen seite wird es überschätzt ... wenn man das modell vernünftig baut ist das eigentlich verhältnismäßig wurscht in welcher sprache ich es umsetze wenn das modell stimmt dann muss das eigentlich in jeder programmiersprache verhältnismäßig sauber umzusetzen sein und das darf eigentlich in keiner dieser gängigen programmiersprachen zu großen sprüngen führen dürfen wenn das passiert dann ist das nach meiner ansicht am modell irgendwas falsch

---

[Transkript 8, Abschnitt V, 21'40", S. 74, Z. 323–341]:

Expertin: also das läuft bei mir eigentlich ganz traditionell ab ich bin kein freund von entwicklungs Umgebungen weil mir zu viel versteckt ist beziehungsweise also ich kenne halt weder kommerzielle case-Umgebungen die ... um benutzen zu können das für die schüler unhandlich das heißt in bei mir sage ich es ist bisschen übertrieben bei den schüler in dem bildschirm befindet sich links oben ein editorfenster wo die programme eintippen und ... rechts unten ist ein fenster wo man das programm aufrufen kann ... das heißt dass man kompilieren muss ... bei java hat man zum beispiel eine Umgebung wo man ein fenster hat wo ein javaknopf ist und man lässt das programm auf den javaknopf fallen und da wird es kompiliert wenn man da eine class-datei darauf lässt dann geht nach option je nachdem was man macht graphisches fenster ... ich möchte eigentlich mehr dazu kommen dass die schüler auch schon größere vorgefertigte programme oder module vielleicht vorfinden wodurch aus dem sourcecode und diese dinge ändern drin also weniger vom anfang an zu schreiben sondern mehr zu ändern und als hilfsmittel dürfen die auch in arbeiten alles benutzen also das heißt ich kläre es mit den schülern ob es bedenken gibt dass da über gegenseitige ... jemand das bevorzugt mitbringt weil er drei bücher oder so was aber die erfahrung zeigt dass es nicht der fall ist

[Transkript 8, Abschnitt V, 26'30", S. 76, Z. 372–379]:

Expertin: also ich habe bis jetzt mit diesen primitiventwicklungs Umgebungen in einführungskursen sozusagen keine über großen probleme gehabt ... das ist immer so dass schüler fehlermeldung nicht lesen sondern ... sie interpretieren das ist ein fehler aber wiederum stochern woran das liegen könnte und dann die fehlermeldungen nicht lesen das ... hat sicherlich damit zu tun dass die fehlermeldungen in englisch kommen und sie können es halt nicht richtig lesen

---

[Transkript 9, Abschnitt V, 35'00", S. 88, Z. 418–431]:

Expertin: ja gut fangen wir an mit dem ersten zustand zur zeit benutze ich nicht das gleiche wir machen java ganz nackt mit texteditor und dem javacompiler aber ich finde das nicht gut eigentlich würde mir wie du geschildert hast eine kleine entwicklungs Umgebung wünschen wir hatten vorher als wir mit pascal programmiert haben hatten wir eben diesen borland-pascal entwicklungs Umgebung so was in der art ist auf jeden fall sinnvoll also integriert eine entwicklungs Umgebung mit der möglichkeit zu debuggen dass man sich die variabelinhalte bei

pascal mal angucken kann zum beispiel das finde ich wünschenswert ich weiss auch dass es alternativen gibt man kann das debuggen und auch das transparent machen und den ablauf eines programms das kann man auch hinkriegen dass man ein programm einbaut das die möglichkeit besteht das ist mir klar aber günstiger fände ich es schon wenn man eine schöne entwicklungsumgebung hätte angemessene komplexität ich kenne mich

Interviewer: es gibt ja solche entwicklungsumgebung die sind aber in der regel nicht für die schule entwickelt worden ist

Expertin: genau ja also was ich von kenne von delphi und entwicklungsumgebung finde ich alle sehr komplex man braucht dafür sehr viel zeit um sich da einzuarbeiten und zeit ist knapp in der schule

In den Beispielen wird die Unzufriedenheit mit professionellen Entwicklungswerkzeugen formuliert. Darüber hinaus werden als Alternative die Entwicklung eigener Werkzeuge und die Anforderung für didaktisch gestaltete Werkzeuge deutlich. Die Didaktik der Informatik befindet sich an dieser Stelle in einer aktuell ungünstigen Situation, da sie es bisher nicht geschafft hat, die Anforderungen zu bündeln und dafür Sorge zu tragen, dass Werkzeuge entwickelt und evaluiert werden, die Eingang in den konkreten Informatikunterricht finden. So sind die Informatiklehrerinnen weiter auf der Suche nach guten Werkzeugen für den Einsatz im unterrichtlichen Zusammenhang.

Die formulierten Arbeitshypothesen werden durch die Beiträge der Expertinnen bestätigt.

### 5.3 Schlussfolgerungen

Offenbar ist es der Fachdidaktik bisher nicht gelungen, die informatische Modellierung als Konzept soweit zu schärfen, dass das Spannungsfeld Fachkonzept versus Unterrichtskonzept produktiv aufgelöst werden kann. Um zu einer konzeptionell elaborierten Variante einer fachdidaktisch geprägten informatischen Modellierung zu gelangen, ist es nötig, die wissenschaftstheoretischen Konzepte der Fachwissenschaft mit lerntheoretischen Konzepten zu verbinden.

- Es fehlt an vergleichenden Darstellungen, die Hilfen für die Entscheidungsfindung für eine konkrete Modellierung liefern. Gefordert sind m. E. Problemstellungen, die mit allen gängigen Modellierungskonzepten analysiert, gestaltet und exemplarisch implementiert werden. Daran kann die Entscheidungsgrundlage für die Konzeptwahl für Schülerinnen transparent gemacht werden können. Ein schülergerechter Zugang kann möglicherweise durch die Wahl von [Implementierungs-]Umgebungen unterstützt werden, die mehrere Arten der informatischen Modellierung unterstützen. Damit können den Schülerinnen im Informatikunterricht Arbeitsmittel für die gängigen Arten informatischer Modellierung zur Verfügung gestellt werden. Der Vorteil eines solchen Zugangs besteht in der Möglichkeit, Teilaufgaben mit angemessenen Fachkonzepten zu modellieren. Der informatischen Modellierung kommt – unabhängig von aktuellen Ausprägungen – auf diesem Hintergrund eine zentrale fachliche Basisfunktion zu.
- Bei der handlungsorientierten Umsetzung von Fachkonzepten in konkrete, ablauffähige Ergebnisse treten bei vielen Schülerinnen Probleme auf. Diese liegen nach Einschätzung der Befragten u. a. darin begründet, dass eine gewisse Routine im Umgang mit konkreten Informatiksystemen eine notwendige Voraussetzung für die erfolgreiche Arbeit im Informatikunterricht darstellen. Das beginnt bei der Bedienung der konkreten in der Schule zur Verfügung stehenden Informatiksysteme, ihrer Benutzungsoberfläche(n) und ihren Nutzungsbedingungen. Es setzt sich über die arbeitsökonomische Bedienung eines Texteditors oder eines Entwicklungssystems bis hin zur Kenntnis von Syntax und Semantik einer bestimmten Programmiersprache fort. Dazu zählen auch Kenntnisse bezogen auf Hilfsmöglichkeiten für die Fehlersuche im eingegebenen Quellcode, seien es nun syntaktische oder durch unangemessene Modellierung verursachte Fehler. Diese zum Teil komplexen Voraussetzungen können nicht sämtlich nebenbei oder integriert erworben werden. Der Weg aus dieser

unbefriedigenden Situation besteht nach Meinung einiger der befragten Expertinnen darin, mit Informatikwerkzeugen zu arbeiten, die die Modellierung unterstützen und Rahmen für die Umsetzung der Modellierung in ablauffähige Programme liefern, die dann gefüllt werden müssen.

- Der fachliche Paradigmenwechsel zur objektorientierten Modellierung befindet sich in der Phase der Konsolidierung und beginnt, seine umfassende Struktur nach und nach zu entfalten. Die befragten Lehrerinnen setzen zunehmend auf „Offene Standards“. Einige Rückmeldungen lassen deutlich werden, dass „moderne Programmiersprachen“ auf ihren didaktischen Wert untersucht werden sollten, bevor sie für den schulischen Einsatz propagiert werden.

Wesentliche Ergebnisse dieser Untersuchung deuten darauf hin, dass eine starke Verschränkung der fachlichen mit fachdidaktisch bedeutsamen Elemente notwendig ist. In den folgenden Ausführungen gilt es, diese empirischen Ergebnisse neben den bereits zuvor theoretisch entwickelten als Hintergrund für konstruktive Ansätze zu berücksichtigen.



## Kapitel 6

# Das Modulkonzept

Dass man in sich sitzt und herausieht, dies Einfache geht zwar früh an. Doch halten es Kinder nicht nur so, auch nachdem sie wissen, dass sie Iche sind. Nicht nur sie selber sehen die Dinge an, diese sehen sie gleichfalls an; das Wasser blinkt her, der Schrank hat ihnen ein Auge, viel Angst kommt von daher. Und wenn wir nach etwas greifen, woran die Oberfläche zu glatt ist, so erscheint diese Glätte als abweisend, während anders Beschaffenes, Griffiges in die greifende Hand einschlägt. Insofern wird dann Inneres nicht nur von außenher benannt und getauft, sondern es leih, was Gefühle angeht, auch dem Draußen Sprache.

[Bloch 1964, S. 38f]

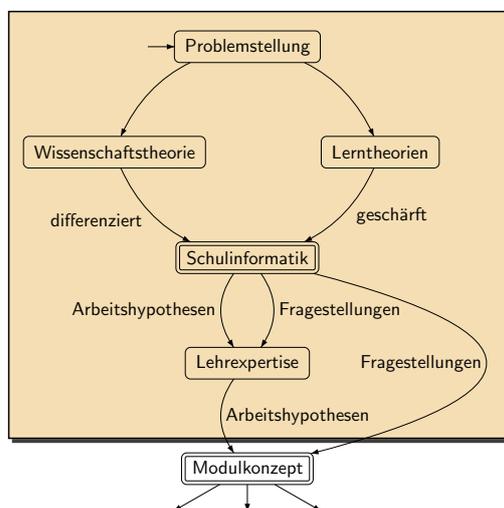


Abbildung 6.1: Struktur Kap. 1–6

Die Ergebnisse der bisher vorgestellten Überlegungen werden im vorliegenden Kapitel konstruktiv zu einem Konzept verdichtet, das sowohl die Erkenntnisse zu den fachlichen als auch zu lerntheoretischen Fragestellungen berücksichtigt.

Die Schulinformatik kann – trotz der in Kapitel 4 ausgewiesenen Probleme – eine erfolgreiche Geschichte vorweisen.

Diese gilt es kritisch und produktiv zu reflektieren und mit zukunftsweisenden Ideen zu verzahnen, so dass den Schülerinnen eine Möglichkeit des Zugangs geboten werden kann, der inhaltlichen, methodischen und aktuellen Erkenntnissen der Lerntheorien gerecht werdende Ansätze konstruktiv zu einem Konzept verdichtet.

## 6.1 Gestaltungsanforderungen

Als gestaltungsleitende Aspekte zur Entwicklung des Curriculums für den Informatikunterricht der Sekundarstufe II dienen die in Kapitel 4 verdichteten Arbeitshypothesen zu Wissenschaftstheorie und Informatik, sowie zur Lerntheorie in Verbindung mit den (ebenfalls dort) dargestellten wissenschaftlichen Fragestellungen. Die Verzahnung zwischen den Fragestellungen und den Arbeitshypothesen werden im Folgenden durch Kreuzreferenzen verdeutlicht.

Darüber hinaus kommt dem grundlegenden Aspekt der Unabhängigkeit der curricularen Elemente über die Zeit eine erhebliche Rolle zu (Zeitinvarianz). Um Entscheidungen für Inhalte zu begründen, bedarf es einer vorgängigen Auswahl von Konzepten für den Informatikunterricht. Dieser Auswahlprozess wurde mit Hilfe der Interviews ausgewählter Expertinnen für den Informatikunterricht (vgl. Kapitel 5) unterstützt. In diesem Auswahlprozess wurden Elemente einer informatischen Bildung, die unterrichtlich zugänglich zu machen sind, benannt und zur Diskussion gestellt. Diese Elemente sind im curricularen Konzept zu berücksichtigen.

Neben fachlich-inhaltlichen Bedingungen sind, wie deutlich herausgearbeitet werden konnte, fachlich-methodische und nicht zuletzt pädagogisch-[fach-]didaktische Bedingungen für einen erfolgreichen Informatikunterricht zu beachten. Die Gestaltung einer technisch-organisatorisch-administrativen Infrastruktur

der Informatiksysteme einer Schule stellt das Handlungsrückgrat für die Umsetzung im Informatikunterricht dar<sup>233</sup>.

Die Zugänge zu informatisch zu bearbeitenden Problemklassen sind primär mit Blick auf die Fachwissenschaft zu strukturieren,<sup>234</sup> wobei der informatischen Modellierung eine Schlüsselfunktion zukommt. Die Frage der lerntheoretischen Orientierung wird unter dieser Fragestellung nicht beleuchtet.<sup>235</sup> Bei der Strukturierung der Inhalte unter didaktischen Gesichtspunkten kann nicht auf eine zu Grunde liegende Ausrichtung an pädagogisch-didaktischen Konzepten, wie sie in der Arbeitshypothese zu lerntheoretischen Fragen formuliert wurde, verzichtet werden. Hier wird die dritte Arbeitshypothese (noch) nicht berücksichtigt, da sie<sup>236</sup> noch zu unscharf ist, um gestaltungsleitende Aspekte in den konstruktiven Prozess einspeisen zu können. Da von den tradierten (Natur- und Ingenieur-) Wissenschaften Informatik<sup>237</sup> als dritte Säule der wissenschaftlichen Arbeitsweise [noch] nicht anerkannt ist, kann diese Arbeitshypothese nicht für Konzepte im Zusammenhang des Informatikunterrichts nutzbar gemacht werden. Solange diese Zielstellung keinen Eingang in den Informatikunterricht findet, ist es in dem hier gestalteten Konzept und einer prototypischen Umsetzung nicht möglich, den gestaltungswirksamen Einfluss zu untersuchen. Die weiteren Kreuzreferenzen können der Tabelle 6.1 entnommen werden.

Fragestellung (Stichwort)	Arbeitshypothese (Stichwort)			
	① Informatik unterscheidet sich von den tradierten Wissenschaften	② informatische Modellierung verändert den „Weltausschnitt“	③ Informatik als 3. wissenschaftliche Arbeitsweise	④ Schülerorientierung, Projektorientierung
Zugänge zu Problemklassen zur Vermittlung nachhaltiger Informatischer Bildung	×	×	(×)	(—)
Strukturierung der Fachinhalte unter didaktischen Gesichtspunkten	×	×	(—)	×
Einfluss des Informatikunterrichts auf das Bild der Informatik bei Schülerinnen	(×)	×	(—)	×

Tabelle 6.1: Arbeitshypothesen und Fragestellungen – Kreuzreferenzen<sup>238</sup>

Damit das Konzept übertragbar und modifizierbar im Sinne einer Fortentwicklung ist, wird ein modularer Aufbau gewählt. Dabei kann – entgegen üblichen informatischen Anforderungen an Module – keine „saubere Schnittstelle“ der Module zueinander expliziert werden. Der Gefahr der Beliebigkeit wird dadurch

<sup>233</sup> Andere Unterrichtsfächer sind ebenfalls zunehmend auf eine solche Infrastruktur angewiesen.

<sup>234</sup> ohne dabei abbilddidaktisch (vgl. Kapitel 1, Fußnote 5, S. 3) zu verfahren – die fachliche Struktur liefert die Basis, die für eine fachlich korrekte Sicht auf Inhalte und Methoden durch die Lehrerinnen eine notwendige Voraussetzung [zur anschließenden Gestaltung im Unterricht] darstellt.

<sup>235</sup> Diese, wie auch andere in der vorliegenden Arbeit nicht weiter verfolgte und nicht oder nur am Rande bearbeitete Fragestellungen stellen weiterhin offene Probleme für die Fachdidaktik dar.

<sup>236</sup> auf Grund der kurzen Geschichte der zu Grunde liegenden Überlegungen

<sup>237</sup> neben theoretischem und experimentellem Vorgehen

<sup>238</sup> Legende:

Zusammenhang und Berücksichtigung Arbeitshypothese und wissenschaftliche Fragestellung

× konstitutiv

(×) möglich, aber nicht sicher, da zur Zeit zu unscharf

(—) Einfluss gering

begegnet, dass verpflichtende – als zeitinvariant erkannte und ausgewiesene – Bestandteile expliziert werden. Wird demnach ein Modul „entfernt“<sup>239</sup>, so sind ggf. Bedingungen verletzt, die durch ein neues Modul ausgeglichen oder durch die stärkere Berücksichtigung in anderen Modulen ausgeglichen werden müssen.

Um die Verbindungen zwischen der Fachwissenschaft, ihrer Entwicklung und pädagogischen Anforderungen zu handlungsleitenden Konzepten zu verdichten, werden Vorschläge unterbreitet, die sowohl dem aktuellen Stand der Bezugswissenschaft entsprechen, aber auch die Einbindung moderner pädagogischer Konzepte ermöglichen.

## 6.2 Entwicklung des Modulkonzepts

Zunächst gilt es, Entscheidungen bezüglich des fachlichen Bereichs zu treffen, der zu Beginn im Anfangsunterricht in der Sekundarstufe II bearbeitet werden soll. Um im Spannungsfeld zwischen Anspruch und Schulwirklichkeit produktive Lösungswege aufzuzeigen, ist es geboten, die eingesetzten Konzepte zu evaluieren. Als grundlegende Elemente eines modernen Informatikunterrichts wurden die Bereiche Rechner-netze und verteilte Systeme sowie informatische Modellierung (inklusive Algorithmen und Datenstrukturen) ausgewiesen.<sup>240</sup> Die Umsetzung der Konzepte im konkreten Informatikunterricht erfolgt problemorientiert.<sup>241</sup>

Die Strukturierung der für Lernprozesse bedeutsamen inhaltlichen Dimensionen soll in enger Verzahnung mit informatischen Fachkonzepten vorgenommen werden. Allerdings darf in schulisch bildenden Prozessen nicht eine fachlich erschöpfende Bearbeitung von Informatikfachthemen erwartet werden. Im Gegenteil ist es anzuraten, die Bearbeitung fachlicher Themen auf verschiedenen Abstraktionsniveaus zu unterschiedlichen Zeitpunkten der schulischen Bildungsprozesse wieder aufzunehmen.<sup>242</sup> Damit sind im Rahmen eines spiralförmigen Curriculums valide, didaktisch gestaltete Informatikinhalte zu thematisieren. Die quantitative Ausprägung der für die Umsetzung des Modulkonzepts verpflichtenden Anteile sollte auf Grund der aktiven und weiterhin sehr dynamischen Weiterentwicklung auf der Anwendungsseite auf einen Teil der Unterrichtszeit im Schulfach Informatik beschränkt bleiben. Das quantitative Fragen durchaus Raum zugestanden wird, zeigen im Zusammenhang mit G8<sup>243</sup> im Saarland dokumentierte Überlegungen: „[...] konzentrieren sich aufs Wesentliche und klopfen nur 60 Prozent der Unterrichtszeit fest. Der Rest ist dem Üben, Vertiefen, Wiederholen und dem Einfallsreichtum der Lehrer überlassen“ [Feuck 2001].

Aktuell bedeutsamen Anwendungsbezügen, aber auch den persönlichen fachlichen Stärken und Interessen der Schülerinnen und der Unterrichtenden, muss im Schulfach Informatik eine Entfaltungsmöglichkeit gegeben werden. Dies gilt vor allem, sobald projektorientiert und fächerübergreifend gearbeitet wird. Ein nicht unerheblicher Teil der Unterrichtszeit ist damit dem Anwendungszusammenhang auf informatischer Basis zu widmen.

Die Arbeit in vernetzten Strukturen setzt soziale Fähigkeiten voraus, deren [Weiter-]Entwicklung in schulischen Aneignungsprozessen explizit Unterrichtszeit zugestanden werden muss.<sup>244</sup> Für diese Aneignungsprozesse kommt dem Schulfach Informatik eine zentrale Funktion zu. Grundlegendes informatisches Verständnis ist dabei eine Voraussetzung zur Unterstützung von Kommunikationsprozessen durch technisch gestaltete Strukturen.

Um die fachlichen Anforderungen zu strukturieren, bietet sich ein modularisiertes Konzept an. Die Arbeit der Schülerinnen mit dem konkreten schulischen Intranet muss auf einer informatischen Fachbasis erfolgen. Daraus folgt, dass dem Modul, das diese Basis zur Verfügung stellt, eine prioritäre Rolle zugestanden werden muss. Vor Beginn einer jeden Arbeit mit den schulischen Informatiksystemen sollten Elemente dieses Moduls [theoretisch durchdrungen und handelnd] erschlossen werden.

<sup>239</sup> z. B. weil es nicht mehr zeitgemäß ist

<sup>240</sup> vgl. hierzu Kapitel 5

<sup>241</sup> vgl. Arbeitshypothese zur Lerntheorie in Abschnitt 4.2.2, S. 78

<sup>242</sup> vgl. Abschnitt 3.3

<sup>243</sup> Kurzform für „Gymnasium in 8 Jahren“

<sup>244</sup> [BMBF 2000, S. 8] führen aus: „[...] wird [von den befragten Unternehmen] die Vermittlung von sozialen und kommunikativen Fähigkeiten wie Team- und Führungsfähigkeiten angemahnt.“

Der Berücksichtigung der historischen Dimension der Module sollte integriert Rechnung getragen werden, d. h. Elemente der Geschichte der Informatik, vergleichende Überlegungen im Zusammenhang mit Ausprägungen für konkrete Implementierungen, programmiersprachliche Konstrukte, sowie Überlegungen zu graphischen Benutzungsoberflächen und zum Recht auf informationelle Selbstbestimmung sollten im Zusammenhang mit konkreten Problemstellungen und Modellierungen thematisiert werden.

Ausgangspunkt des Informatikunterrichts sind Problemstellungen, die formuliert werden und zu denen nach erfolgter Modellierung reflektierend zurückgekehrt wird. Damit stellt die Problemorientierung die methodische Klammer des Informatikunterrichts dar.<sup>245</sup>

## 6.2.1 Vorstellung der Module

### Informatiksysteme verstehen und verantwortlich nutzen

„Eine produktive [...] Nutzung der Informationsressourcen [...] ist nur möglich auf einem hohen Stand informationeller Bildung [...] Informationsgesellschaften verdienen ihren Namen erst, wenn in ihnen die Bürgerinnen und Bürger in die Lage versetzt werden, [...] die Methoden der Informationsverarbeitung zu beherrschen [...] und die [...] erarbeitete Information auch einsetzen zu können. Das Postulat der informationellen Selbstbestimmung, vom Bundesverfassungsgericht zunächst als Aufgabe des Datenschutzes formuliert, sollte als Recht des freien Umgangs mit Information auf kompetenter Grundlage neu formuliert werden“ [Kuhlen 2002, S. 10, 19].

Die zunehmende Nutzung komplexer vernetzter Informatiksysteme im schulischen Zusammenhang sollte dazu führen, dass die Hintergründe für Regelungen in den Blick genommen werden. Die erweiterte Nutzung hat zur Folge, dass unabdingbare Regelungen für Räume, Informatiksysteme und vernetzte Systeme aufgestellt werden (durch die Systembetreuung, die Schulgemeinde, die Schulträger, den Gesetzgeber). Diese Regelungen basieren auf dem Schutz der Persönlichkeit und der Möglichkeit, ein Leben in eigener Verantwortung zu gestalten und der Vermeidung unnötiger Belastung und Schädigung der technisch-administrativen Infrastruktur. Bei allen Nutzerinnen muss soviel Hintergrundwissen vorhanden sein, dass die Umsetzung dieser Anforderungen nicht durch „Gehorchen und Befolgen“, sondern durch Einsicht in die gesellschaftlichen und technischen Hintergründe (informatischer Themenbereich: Rechnernetze und verteilte Systeme) ermöglicht wird. Bei der Arbeit mit elaborierten Lernumgebungen<sup>246</sup>, die zunehmend Bestandteil schulischer Intranetstrukturen sind, müssen Rechte Dritter beachtet werden. Durch Einsicht in schutzwürdige Belange (Netiquette) kann z. B. das Veröffentlichen oder Kopieren von geschützten Materialien wirksamer verhindert werden als ausschließlich durch Verbote.<sup>247</sup> Daraus resultieren verschiedene Regelungen, die in den allgemein bildenden Schulen zu Regelwerken verdichtet, massiv Bildungsprozesse (auch in anderen Fächern) beeinflussen.

Auf die Besonderheit von kooperativen und kollaborativen Unterstützungssystemen Computer Supported Collaborative Work (CSCW) und CSCL soll in diesem Zusammenhang hingewiesen werden. Solche Informatiksysteme können sowohl als Bestandteil dieses Moduls betrachtet werden, sie sind aber auch anderen Dimensionen zuzuordnen. CSCL soll insbesondere Lernprozesse in Gruppen (Teamarbeit, Projektorientierung) technisch unterstützen, darüber hinaus sind CSCW/CSCL-Systeme geeignet, im Modellierungskontext (objektorientierte) Erweiterungen zuzulassen (vgl. [Züllighoven 2001]).

Wenn Betriebssysteme handlungsorientiert erschlossen werden, ist dafür Sorge zu tragen, dass jedes Element auf dem Hintergrund der Verallgemeinerung und Zukunftssicherheit eine unterrichtliche Umsetzung erfährt. Dies kann z. B. dadurch erreicht werden, dass Betriebssystemfunktionen über betriebssystem- und plattformunabhängige [Skript-]Sprachen für explorative Unterrichtseinheiten genutzt werden. Dies eröffnet damit gleichzeitig die Erarbeitung curricularer Elemente, die portabel sind und es den Schülerinnen erlauben, erarbeitete Elemente zur Automatisierung von Betriebssystemfunktionen zu nutzen.

<sup>245</sup> wie in Abschnitt 4.3 ausführlich dargestellt

<sup>246</sup> siehe z. B. Computer Supported Cooperative Learning (CSCL), exemplarisch in [Wessner und Pfister 2001] dargestellt

<sup>247</sup> Interessant ist, dass dieser – für allgemein bildende Schulen wichtige – Punkt bisher nicht differenziert untersucht wurde.

Die Exploration setzt Betriebssysteme voraus, bei denen eine Trennung zwischen administrativen Benutzerinnen und „normalen“ Benutzerinnen betriebssystemimmanent implementiert ist. Ist diese Voraussetzung nicht erfüllt, kann die handelnde Exploration die Systemsicherheit gefährden. Damit kommt der Gestaltung des Handlungsrückgrats eine grosse Bedeutung zu.

Mit der Entwicklung graphischer Benutzungsoberflächen ist ein aktueller Stand zur Benutzung von Informatiksystemen erreicht, der in den nächsten Jahren eine Erweiterung erfahren wird. Soll Informatikunterricht zukunftsweisende Impulse setzen, so ist hier durch die Trennung der verschiedenen Ebenen bei der Modellierung von Informatiksystemen mit Hilfe von Entwurfsmustern, beispielsweise mit dem Konzept Model View Control (MVC) eine Möglichkeit aufzuzeigen, wie zukünftige Entwicklungen berücksichtigt werden können, ohne erarbeitete und von den Schülerinnen verstandene Fachkonzepte obsolet werden zu lassen. Dabei soll nicht unterschlagen werden, dass die Benutzung von Entwurfsmustern im Informatikunterricht nicht ohne Probleme möglich ist. Ein zentrales (bisher nicht zufriedenstellend gelöstes) Problem besteht in der Motivation von Entwurfsmustern auf einer schmalen fachlichen Basis ohne hinreichende Projekterfahrung (im Sinne der Informatik) und der damit verbundenen Notwendigkeit (und Nützlichkeit) dieses Abstraktionsmechanismus.

### **Modellierung – zentrales Feld informatischer Arbeit**

Bei der Modellierung gilt es herauszuarbeiten, wozu dieser Prozess vorgenommen wird. Die Exploration und Deskription des Informatiksystems der Schule am konkreten Schülerarbeitsplatz ist für einen grundlegenden Zugang zur Modellierung geeignet. Damit kann die Verbindung zu dem Modul *Informatiksysteme verantwortlich nutzen und verstehen* hergestellt werden. Mit der Modellierung sollte die Möglichkeit eröffnet werden, über motivierende Probleme hinaus Klassen von Problemen zu bearbeiten. Es ist notwendig, Unterschiede zwischen der Modellierung in der Informatik und der Modellierung in anderen technisch-wissenschaftlichen Gegenstandsbereichen durch Rückwirkung des Modells „auf die Welt“ deutlich werden zu lassen. Als Ergebnis (Produkt) der informatischen Modellierung werden z. B. Arbeitsabläufe nicht nur theoretisch, sondern konkret beeinflusst, verändert oder gar zu großen Teilen automatisiert.

Es gibt nicht nur eine informatische Modellierung, sondern verschiedene, die sich dadurch auszeichnen, dass der Gegenstandsbereich jeweils durch eine besondere „Brille“ betrachtet wird, um besonders effektive Modellbildungsstrategien anwenden zu können (vgl. Abschnitt 2.4, S. 21). Die dadurch vorgenommene Reduktion sollte im konkreten Fall thematisiert werden. Schülerinnen kann die durch die jeweilige Modellierung „abgeschnittene“ oder „ausgeblendete“ Sicht verdeutlicht werden, indem z. B. eine konkrete Problemstellung mit verschiedenen Sichten betrachtet und bearbeitet wird. Methoden und Veranschaulichungen, die auf eine ganz spezielle Modellierung beschränkt sind, haben nur einen begrenzten Zukunftswert. Heute werden umfangreiche Probleme nicht mehr mit Hilfe der Structured Analysis Design Technique (SADT) oder der Jackson-Methode<sup>248</sup>, sondern mit objektorientierten Methoden (z. B. Rational Unified Process (RUP) ) bearbeitet. Andererseits haben Und-Oder-Bäume und das Entity Relationship Model (ERM) weiterhin ihren Platz bei bestimmten Anwendungsfällen. Hier ist von Fall zu Fall zu entscheiden, welche konkrete Darstellung als sinnvoller und notwendiger Beitrag zur Modellierung unterrichtlich verankert werden sollte. Allerdings sollte eine Orientierung an fachlichen Standards vorgenommen werden, wobei eine sinnvolle didaktische Gestaltung angestrebt werden muss.

### **Erkenntnisse der theoretischen Informatik im Anwendungskontext**

Theoretische Überlegungen in der Informatik haben nicht selten unmittelbare Konsequenzen. Sie sind notwendig für das Verständnis grundlegender Wirkprinzipien von Informatiksystemen, lassen sich einsetzen, um Strukturen zu verdeutlichen und zu modellieren.

Die Verbindung mit nicht-naturwissenschaftlichen Fachinhalten kann über das gemeinsame Feld der Kategorisierung von Sprachen erschlossen werden. Fragen, die die Zukunft der Schülerinnen betreffen, berüh-

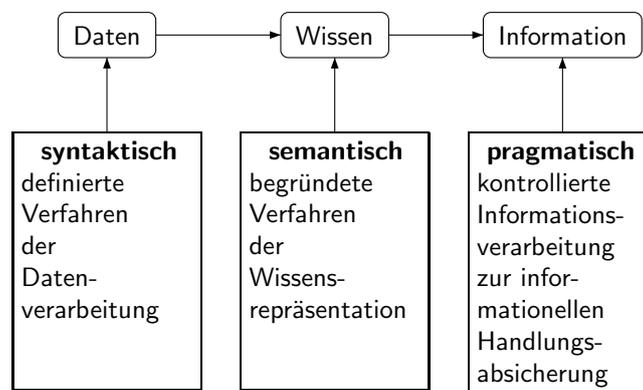
---

<sup>248</sup> Jackson Structured Programming (JSP)

ren die Komplexität möglicher Informatiksysteme, die Chancen der Sprachübersetzung natürlicher Sprachen, die prinzipiellen Grenzen informatischer Modellierung und bieten sich für fächerübergreifende Ansätze aus dem Bereich der Philosophie geradezu an.

### Wissensbasierte Modellierung

Ein zentraler Begriffe der Informatik ist Information. Mit Information können im Zusammenhang von Informatiksystemen nicht nur technische Ziele, sondern auch Absichten (von Menschen) verbunden sein. Genau diese lassen sich kaum angemessen formalisieren.<sup>249</sup> Andererseits ist es notwendig, im Zusammenhang der allgemeinen informatischen Bildung eine »Arbeitsdefinition« zu verwenden. Für Schülerinnen soll deutlich werden, dass es notwendig ist, einen validen Begriffsapparat zu benutzen, der es gestattet, in einer Gemeinschaft einen Inhaltsbereich gestaltend zu erschließen. Die zum Standardrepertoire der Informatik gehörenden Begriffe Syntax, Semantik und Pragmatik stellen einen möglichen informatischen Zugang zur anfänglichen Bestimmung der Begriffe Daten, Wissen und Information (vgl. Abbildung 6.2) bereit.



nach [Fuhr 2000, S. 10]

Abbildung 6.2: Daten – Wissen – Information

Im Zusammenhang mit der zunehmenden Bedeutung großer Datensammlungen (z. B. im Zusammenhang mit dem Internet) ist die Akquisition von Daten leichter denn je. Die Modellierung muss von der Anwendung auf die dahinter liegenden Strukturen und Modellvorstellungen verlagert werden. Damit kann z. B. aus der *Black Box* Suchmaschine eine *White Box* entwickelt werden. Allerdings sind gerade die bei der konkreten Modellierung auftretenden Schwierigkeiten nicht ohne beachtlichen unterrichtlichen Aufwand zu lösen. Der besondere Variablenbegriff der wissensbasierten Modellierung bedarf der ausführlichen Fundierung, sollen die Erkenntnisse nicht an der Oberfläche bleiben.

In diesem Zusammenhang sollten die gesellschaftlichen Rahmenbedingungen für die Arbeit mit großen Datenbeständen beispielhaft illustriert werden: Persönlichkeitsschutz, Datenschutz und Datensicherheit – zur interessengeleiteten Bewertung des Rechts auf informationelle Selbstbestimmung und ihrer rechtlichen Dimension.

Der Gewinnung von Daten und dem Gewinnen von Information sind die beiden Endpunkte der Nutzung von Informatiksystemen zuzuordnen – zum Einen in Richtung auf die Vorbereitung der maschinellen Verarbeitung und zum Anderen in Richtung auf die Benutzung eines Informatiksystems.

<sup>249</sup> detaillierter in Kapitel 2 dargestellt

### Objektorientierte Modellierung (OOM)

Die objektorientierte Beschreibung von konkreten Informatiksystemen liefert ein Modell für die erfolgreiche Implementierung von Funktionalität. Inzwischen existiert ein zunehmender Fundus an Ideen für Problemstellungen, die sich im Schulfach Informatik für die objektorientierte Modellierung anbieten. Im Zusammenhang mit dem Teilgebiet Rechnernetze und verteilte Systeme (RvS), mit ereignisgetriebenen Systemen, mit graphischen Benutzungsoberflächen (GUI), mit Interaktion und Kommunikation und nicht zuletzt mit Simulation kommt der objektorientierten Modellierung eine besondere Bedeutung zu, die unterrichtlich erfolgreich umgesetzt werden kann. Allerdings sind die verpflichtenden Schwerpunkte: einfache Algorithmen und Datenstrukturen und der Variablenbegriff explizit zu thematisieren. Oftmals wird bei der Konstruktion neuer curricularer Elemente zu wenig berücksichtigt, dass notwendige Voraussetzungen aus dem Fundus der bekannten Modellierungen weiterhin berücksichtigt werden müssen und vor allem ihren zeitlichen Tribut im Unterrichtsprozess fordern.

Aus dem Bereich der objektorientierten Modellierung bieten sich an mehreren Stellen Übergangsmöglichkeiten zu anderen Modellierungen an: z. B. kann mit der Anbindung von Datenbankschnittstellen auf umfangreiche Datenbestände zugegriffen werden; außerdem sind Erweiterungen verfügbar, die die Nutzung von Elementen der funktionalen Modellierung erlauben. Als fakultative Schwerpunkte im Zusammenhang mit der objektorientierten Modellierung bieten sich die Bereiche Nebenläufigkeit, Dokumentenbeschreibungssprachen (Äquivalenz von Dokumenten- und Datenstruktur) an.

### Funktionale Modellierung

Funktionen als Argumente von Funktionen sind neben speziellen Möglichkeiten der Anwendung von Operationen auf Listen von Funktionen die zentralen Elemente der funktionalen Modellierung. Anwendungsfälle für den Informatikunterricht kränkten bisher an der Nähe zur Mathematik, die von vielen Lehrerinnen<sup>250</sup> und Schülerinnen nicht positiv annotiert wurden. Es sollte der Versuch unternommen werden, die Integration funktional modellierter Elemente an den Stellen zu unternehmen, die im Modellierungsprozess sinnvoll und angemessen sind. Drei Möglichkeiten werden hier exemplarisch angeben:

1. Durchlauf durch eine Verzeichnisstruktur [Good 2002],
2. Permutation einer (beliebigen) Sequenz [Linkweiler 2002, S. 131],
3. Dateibearbeitung mittels Generator [Cannon 2002].

### Anmerkungen zum Modulkonzept

Das Modulkonzept wurde nicht durch eine Analyse der Curricula der 16 Bundesländer auf Überdeckungen oder Konformität geprüft. Diese Überprüfung ist bei der Konkretion in Stoffverteilungspläne<sup>251</sup> vorzunehmen. Curriculare [Weiter-]Entwicklungen<sup>252</sup> berücksichtigen zunehmend die Leitlinien des von der GI vorgeschlagenen Gesamtkonzepts [Gesellschaft für Informatik 2000]. Bei der Entwicklung des Modulkonzepts fanden in einem frühen Stadium<sup>253</sup> die Leitlinien Eingang.

Die vorgestellten Module stellen einen Rahmen dar, der nicht mit einer Sequenz verwechselt werden darf. Bei der Gestaltung konkreter Kurse ergeben sich Verzahnungsmöglichkeiten zwischen verschiedenen Modulen. Beispiele für modulübergreifende Fragestellungen:

- Mit der prädikativen Modellierung ist es möglich, das Konfigurationsmanagement von Betriebssystemen unterrichtlich zu untersetzen.

<sup>250</sup> vgl. Abschnitt 5.2.2

<sup>251</sup> vgl. exemplarisch Anhang F.1 und F.2

<sup>252</sup> Inzwischen wurden Curricula veröffentlicht, die explizit an den Leitlinien orientiert sind – vgl. [Hessisches Kultusministerium 2002], [Ministerium für Bildung, Wissenschaft und Kultur – Mecklenburg-Vorpommern 2001].

<sup>253</sup> vgl. [Humbert und Schubert 1999] – der Workshop zum Gesamtkonzept während der GI-Jahrestagung 1999 (Paderborn) wurde vom Autor geleitet

- Die Attribute von Dateien können nicht nur über die vom konkreten Betriebssystem zur Verfügung gestellten Anzeigemöglichkeiten, sondern auch über plattformübergreifende und -unabhängige [Skript-]Sprachen herausgefunden werden [Humbert 2000a].
- Mit der Thematisierung von Internet-Protokollen können automatentheoretische Elemente in den Unterrichtszusammenhang eingebunden werden [Deckers 1996].

Die Ausprägung der inhaltlichen Umsetzung variiert mit der konkreten Problemstellung, mit der sich die Schülerinnen modellierend, handelnd, gestaltend und reflektierend auseinandersetzen.

## 6.2.2 Konkretisierung – Beispiele zur Umsetzung

Modul	Kurzbezeichnung/ Charakteristik	Zielorientierung	Methodischer Rahmen	„Werkzeug“
<b>Informatiksysteme verantwortlich nutzen</b>				
	Netiquette	Dienste auf TCP <sup>254</sup> -Basis als informatischer Hintergrund für Regelungen verstehen	Partnerarbeit, Gruppenarbeit	schulisches Intranet – Dienste: Mail, Hypertext
	Betriebssystem	plattformunabhängige Strukturen kennen und explorieren können	explorativ	Skriptsprache
	RvS <sup>255</sup>	Server/Klienten Modell Nebenläufigkeit	arbeitsteilig / Gruppenarbeit	schulisches Intranet und Skriptsprache
	CSCL <sup>256</sup>	Nutzung ausgewählter Elemente der Gruppenarbeit mit Informatiksystemen	Gruppenarbeit	BSCW <sup>257</sup> Infrastruktur
<b>Elemente der theoretischen Informatik</b>				
	Keller	Aufbau und Analyse (Parsen) von Dokumenten	projektorientiert	Skriptsprache
	endliche Automaten	Sprachen und Grammatiken	verschiedene	imperative Sprache
<b>informatische Modellierung</b>				
	OOM <sup>258</sup>	Modellierung des Informatiksystems der Schülerinnen Algorithmen und Datenstrukturen	verschiedene	Skriptsprache
	Prädikative Modellierung	Möglichkeiten und Probleme	Gruppenarbeit	Prolog
	Funktionale Modellierung	fächerkoordinierend: Informatik und Mathematik	verschiedene	Skriptsprache

Tabelle 6.2: Modulkonzept – Beispielübersicht <sup>259</sup>

<sup>254</sup> RvS – Rechnernetze und verteilte Systeme

<sup>255</sup> TCP – transmission control protocol

<sup>256</sup> CSCL – Computer supported cooperative learning

<sup>257</sup> BSCW – Basic System Cooperative Workspace

<sup>258</sup> OOM – objektorientierte Modellierung

<sup>259</sup> Legende:

Modul

im Zusammenhang mit der Auswertung dokumentiert

In der Tabelle 6.2 sind konkrete unterrichtliche Umsetzungen<sup>260</sup> zum Modulkonzept ausgewiesen. Teile des Unterrichts wurden einer „gerichteten“ Auswertung zugänglich gemacht, die in Kapitel 7 dargestellt wird.

Die Notwendigkeit eines konkreten technischen Rahmens („Werkzeug“) führt oftmals zu einer verkürzten Sicht auf den Informatikunterricht. Bei den in der Tabelle angegebenen Beispielen wurde daher fast vollständig auf die Nennung konkreter technischer Ausprägungen verzichtet. Es wird deutlich, dass technisch vernetzte Strukturen, die den Quasistandards des Internet<sup>261</sup> genügen, im schulischen Intranet zur Verfügung stehen sollten. Dieses Handlungsrückgrat (vgl. Abschnitt 6.1, S. 104) stellt die unabdingbare Voraussetzung für den [administrativen] Rahmen dar, in dem Informatikunterricht verantwortlich durchgeführt werden kann. Darüber hinaus ist durch die technisch-organisatorisch-administrative Infrastruktur jeder Schülerin ein individueller (personenbezogener) Account im lokalen Intranet zur Verfügung zu stellen; andere Lösungen widersprechen den Anforderungen, die für die unterrichtliche Umsetzung des Moduls *Informatiksysteme verantwortlich nutzen* vorauszusetzen sind. Ebenso dürfen auf den konkreten Schülerarbeitsplätzen nach Beenden der Arbeit (Ausloggen) keinerlei lokale Daten zugänglich bleiben, damit dem Persönlichkeitsrecht der Schülerinnen Rechnung getragen wird.

### 6.3 Umsetzungsvorschläge

In Abschnitt 6.2.2 wurde auf Unterrichtseinheiten verwiesen, die auf der Basis des Modulkonzepts entwickelt wurden. Um für einen längeren Kursabschnitt Planungsgrundlagen zu entwickeln, müssen die jeweiligen Rahmenbedingungen berücksichtigt werden. Geltende Lehrpläne sind als verbindliche Grundlage für die Gestaltung des Unterrichts heranzuziehen. Für die konkrete Schule sind von der Fachkonferenz Informatik<sup>262</sup> eine Reihe von Entscheidungen zu treffen.<sup>263</sup> Häufig werden die inhaltlichen Entscheidungen der Fachkonferenz in einem Stoffverteilungsplan zusammengefasst. Dieser weist stufenbezogene Schwerpunktsetzungen (auf der Grundlage der geltenden Lehrpläne) für die Umsetzung an der Schule aus. Für den Informatikunterricht in der Sekundarstufe II kommt überdies dem Stellenwert des Schulfachs Informatik für das Abitur der Schülerinnen eine große Bedeutung zu. Von diesen Rahmenbedingungen kann durch die Gestaltung der schulisch für relevant erachteten Elemente innerschulisch der Stellenwert des Faches (wenn auch nur geringfügig) verändert werden. Schulische Profilbildung, wie sie zunehmend (politisch) gefordert wird, kann das Schulfach Informatik in der Einzelschule befördern. Diese Einzelmaßnahmen können aber nicht darüber hinwegtäuschen, dass nur durch politisch-administrative Maßnahmen der KMK auf Bundesebene (vgl. Abschnitt 4.2.1, S. 62) geänderte Rahmenbedingungen für Informatik in der Sekundarstufe II geschaffen werden.

Im Folgenden werden zwei Stoffverteilungspläne vorgestellt, die als Grundlage für den Anfangsunterricht in der gymnasialen Sekundarstufe II im Bundesland Nordrhein-Westfalen vom Autor erprobt wurden. Diese Stoffverteilungspläne und ihre Umsetzung werden im Folgenden an Hand des Modulkonzepts eingeordnet und einer Reflexion unterzogen.

<sup>260</sup> i. d. R. als Unterrichtseinheiten

<sup>261</sup> Request for comment (RFC)

<sup>262</sup> die Gesamtheit aller Lehrerinnen, die Informatik in einer Schule unterrichten

<sup>263</sup> Für das Bundesland Nordrhein-Westfalen werden in [MSWWF 1999, S. 111] die Aufgaben zusammenfassend dargestellt: „[die Fachkonferenz entscheidet über] Grundsätze zur fachmethodischen und fachdidaktischen Arbeit sowie über Grundsätze zur Leistungsbewertung. [...] Die Fachkonferenz berät und entscheidet z. B. in den folgenden Bereichen:

- Präzisierung der fachlichen Obligatorik und Maßnahmen zur Sicherung der Grundlagenkenntnisse
- Absprachen zu den fachspezifischen Grundlagen der Jahrgangsstufe 11
- Absprachen über die konkreten fachspezifischen Methoden und die konkreten Formen selbstständigen Arbeitens
- Absprachen über den Rahmen von Unterrichtssequenzen
- Absprachen über die Formen fachübergreifenden Arbeitens und den Beitrag des Faches zu fächerverbindendem Unterricht
- Koordination des Einsatzes von Facharbeiten
- Absprachen zur besonderen Lernleistung. [...]“

## Endliche Automaten – Grammatiken<sup>264</sup>

### Modulzuordnung für den Stoffverteilungsplan: „Elemente der theoretischen Informatik“

#### *Einführung in ausgewählte Elemente der theoretischen Informatik*

Durch Alltagssituationen, die mit einfachen Mitteln modelliert werden können, findet die Einführung in Modellierungstechniken und in formale Darstellungsformen endlicher Automaten statt. Die modellierten Automaten werden implementiert und die Grenzen der Modellierung aufgezeigt. Um die Grenzen der Leistungsfähigkeit der endlichen Automaten zu überschreiten, werden Kellerautomaten modelliert. Für die Implementierung werden ausschliesslich die unabdingbaren Elemente der jeweils verwendeten Programmiersprache problemnah als Werkzeug eingeführt. Durch einfache Schnittstellen werden die Schülerinnen von dem Ballast der Implementierung von Ein-/Ausgaberroutinen befreit. Der vorliegende Vorschlag wurde mehrfach unterrichtlich erprobt.

#### Charakterisierung des Vorschlags und Reflexion der Durchführungen

- Der Vorschlag ist an dem Basiskonzept Zustandsautomat der Fachwissenschaft<sup>265</sup> ausgerichtet.
- Die unterschiedlichen Darstellungsformen endlicher Automaten unterstützen verschiedene Lerner-typen, da den Schülerinnen aus eigener Handlung<sup>266</sup> („enaktiv“) die Bedienung von Automaten be-kannt ist, die Darstellung als Automatengraph („ikonisch“) in der graphisch gestalteten Abstraktion und darüber hinaus die formalen Darstellungsformen („symbolisch“) als Automatentabelle und in der konkreten Implementierung als aufeinander aufbauend und sich gegenseitig ergänzende Elemen-te angenommen werden.
- Die Rückbindung der durch eine erfolgreiche Implementierung gewonnene Erkenntnis, dass bei der Modellierung Entscheidungen getroffen werden, die über das „Funktionieren“ der Implementierung entscheidet.
- Erweiterungen von vereinfachten (im Sinne der Reduktion der Funktionalität) Automatenmodellen führen schnell zu unübersichtlichen Darstellungen, die nicht mehr „auf den ersten Blick“ durchschaut werden können.

Darüber hinaus konnte in der Reflexion der durchgeführten Reihen die Vermutung bestätigt werden, dass die Schülerinnen keine Vorkenntnisse zur Modellierung von Automaten mitbringen.<sup>267</sup>

Ein Nachteil des Konzepts ist die Beschränkung auf das o. g. Modul und – damit verbunden – eine einsei-tige Sicht auf die Informatik. Versuche, die Unterrichtsreihe zu verkürzen führten dazu, dass wesentliche Teile, die nach der ersten Erweiterung der Modellierung diskutiert werden können, nicht mehr betrachtet werden konnten. Dies gilt vor allem für die über die einfachen Automaten hinausweisenden Betrachtung von Grammatiken ihrer Schichtung und der Diskussion der prinzipiellen Möglichkeiten und Grenzen von Automaten.

<sup>264</sup> Details vgl. Anhang F.1, S. 198

<sup>265</sup> vgl. Tabelle 2.3, S. 28

<sup>266</sup> Die Zuordnungen werden hier nach dem Repräsentationsmodell von BRUNER vorgenommen (vgl. Abschnitt 3.1, S. 38).

<sup>267</sup> Eine Ausnahme stellten einige Schülerinnen dar, die im Wahlpflichtunterricht der Sekundarstufe I (Differenzierungsbereich II, Jahrgangsstufen 9 und 10) eine Ampelanlage modelliert hatten. Diese Schülerinnen kannten dennoch keine formale Beschreibung des zu Grunde liegenden Automaten.

## RvS – Kollaboration – Modellierung<sup>268</sup>

### Modulzuordnung für den Stoffverteilungsplan: „Informatiksysteme verantwortlich nutzen“ und „Informatische Modellierung“

*Erarbeitung der Voraussetzungen für die Arbeit mit den schulischen Informatiksystemen, Themenbereich „Rechnernetze und verteilte Systeme (RvS)“ mit dem Ziel des Erwerbs von Handlungskompetenzen zur effektiven Nutzung der schulischen Intranetinfrastruktur auf einer fachlichen Basis, elektronisches Publizieren, Computer Supported Cooperative Learning, informatisches Modellieren am Beispiel der Objektorientierung.*

Durch einen breiten Zugang, der immer wieder von dem konkreten Informatiksystem ausgehend, Gegenstände und Methoden der Informatik in den Mittelpunkt des Unterrichts rücken, wird die Vielschichtigkeit informatischer Herangehensweisen transparent. Die Schülerinnen erwerben sowohl konkretes Handhabungswissen, sollen aber darüber hinaus dieses Wissen in einen größeren Fachzusammenhang einordnen. Dabei finden Elemente der Geschichte der Informatik<sup>269</sup>, Modellierungskontexte und modulübergreifende Fragestellungen Eingang in die unterrichtliche Umsetzung.

Einigen Schülerinnen sind Elemente (z. B. Arbeit mit E-Mail) bereits aus anderen Kontexten bekannt. Hier werden diese Gegenstände unter informatischen Gesichtspunkten erarbeitet, so dass die Schülerinnen die hinter den Anwendungen liegende Strukturen erarbeiten und zur Konstruktion von Problemlösungen nutzen können.

Probleme des vielgestaltigen Ansatzes: der Zusammenhang hinter den Fragestellungen, die unterrichtlich bearbeitet werden, kann bei einer derart umfangreichen und somit dichten Planung verloren gehen. An einigen Stellen bleibt der Zugang oberflächlich, weitergehende Fragen und die vertiefende Behandlung muss „auf später“ verschoben werden. Da für die Umsetzung der in diesem Stoffverteilungsplan angegebenen Teilbereiche ein ganzes Schuljahr zur Verfügung steht, kann an ausgewählten Stellen eine Vertiefung erreicht werden.

In einem nach diesem Stoffverteilungsplan durchgeführten Informatikkurs konnte eine Untersuchung zum Einfluss des Informatikunterricht auf das Bild der Informatik bei den Schülerinnen durchgeführt werden. Die Planung, Durchführung und Auswertung dieser Untersuchung wird in Kapitel 7 dokumentiert.

## Prädikative Modellierung

### Modulzuordnung für die Einheit: „Informatische Modellierung“

*Möglichkeiten, Probleme und Grenzen der prädikativen Modellierung*

Randbedingungen: Das Konzept wurde in einer besonderen Situation entwickelt: zwei Informatikkurse der gymnasialen Oberstufe (11. und 12. Jahrgang) mussten zusammengefasst werden. Damit konnte der geltende Stoffverteilungsplan (vgl. Abschnitt 6.3, S. 112) nicht umgesetzt werden. In dieser Situation wurde die Entscheidung getroffen, für die Schülerinnen des 12. Jahrgangs eine – verglichen mit den Vorerfahrungen – andere Art der Modellierung einzuführen. Der prädikativen Modellierung ging eine Einführung in RvS voraus, so dass die Schülerinnen mit dem schulischen Intranet auf einer personenbezogenen Basis arbeiten konnten. Darüber hinaus wurden die Schülerinnen mit Hypertextstrukturen (am Beispiel von HTML) vertraut gemacht.<sup>270</sup>

<sup>268</sup> Details vgl. Anhang F.2, S. 210ff

<sup>269</sup> illustriert an konkreten Problemstellungen

<sup>270</sup> In diesem Informatikkurs wurde das universitäre Tagespraktikum für eine Lehramtsstudentin durchgeführt.

Bestandteil des Vorlesungsskriptums zur Didaktik der Informatik I (für das Lehramt für die Sekundarstufe II) der Universität Dortmund ist ein Anhang, der für den hier betrachteten Zusammenhang bedeutsam ist:

Beispiel für prädikative Modellierung (enthält Elemente der Unterrichtsvor- und Nachbereitung) [Schubert 2001a, S. 136–141]

Die Umsetzung wurde von den Schülerinnen zusammenfassend einer kritischen Reflexion unterzogen, die sich an einigen Leitfragen orientiert. Eine ausführliche Darstellung des Verfahrens und der Ergebnisse findet sich in Abschnitt 7.1 (S. 116).

Das zentrale Problem für diese Art der Modellierung im Anfangsunterricht besteht darin, dass den Schülerinnen die hinter der Auflösung der Anfragen stehenden Verfahren nicht ohne tiefere Einblicke in die von dem konkreten System benutzten Backtrackingalgorithmen deutlich werden. Das grundlegende Problem des Resolutionskalküls konnte den Schülerinnen nur unzureichend verdeutlicht werden. Dadurch bedingt, kommen Zweifel an der Tragfähigkeit der prädikativen Modellierung für den Anfangsunterricht auf. Wenn die hinter der Modellierung verborgene abstrakte Maschine nicht verstanden wird, kann vorgeblich erfolgreich unterrichtet werden – im Hintergrund werden jedoch Fehlvorstellungen aufgebaut. Ziel des Informatikunterrichts ist es, übersteigerten Erwartungen (die vor allem mit den Möglichkeiten im Rahmen der Künstlichen Intelligenz assoziiert werden) entgegenzuwirken.

---

Konkrete Gestaltungsvorschläge für schulinterne Stoffverteilungspläne sind als Anhänge F.1, S. 198ff und F.2, S. 210ff beigelegt. Diese Stoffverteilungspläne stellen eine erprobte Grundlage für die Kursgestaltung in der gymnasialen Oberstufe dar. Beide Vorschläge sind mit Material aus der Unterrichtsvorbereitung angereichert. In einigen Fällen wird die Unterrichtsplanung unter Zuhilfenahme eines aussagekräftigen Planungsrasters und Angabe der Stellung innerhalb der jeweiligen Unterrichtsreihe ausführlich dargestellt.

Diese Gruppen haben recht, wenn sie – wie wir – bekräftigen, daß man das Bewußtsein außerhalb der Praxis nicht verändern kann. Es muß jedoch hervorgehoben werden, daß die Praxis, die das Bewußtsein verändert, nicht nur aus Handeln, sondern aus Handeln und Reflexion besteht. Es gibt somit eine Einheit zwischen Theorie und Praxis, in der beide in einem ständigen Wechsel von der Praxis zur Theorie und zurück zu neuer Praxis aufgebaut, geformt und umgeformt werden.

[Freire 1977, S. 117]

## Kapitel 7

# Einschätzung der Informatik durch Lernende

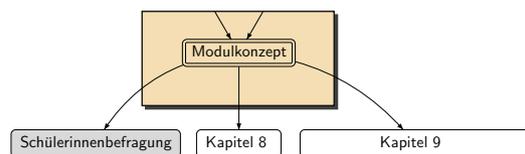


Abbildung 7.1: Struktur Kap. 6–7

In der Forschungsarbeit wurde über die Reflexion theoretisch orientierter Beiträge aus dem Umfeld der Schulinformatik in Kapitel 6 das Modulkonzept entwickelt. Dabei fanden die Ergebnisse aus Kapitel 5 (Interviewserie mit Expertinnen) Berücksichtigung. Im Folgenden werden Ergebnisse dargestellt, die sich auf die Einschätzung von Schülerinnen beziehen, die nach diesem Konzept unterrichtet wurden.

Im Zusammenhang mit den forschungsleitenden Fragestellungen<sup>271</sup> wird die Frage untersucht, was von dem Bild der Wissenschaft Informatik von den Lernenden angenommen wird. Dabei wird erwartet, dass durch den Informatikunterricht das Bild der Wissenschaft Informatik bei den Schülerinnen verändert wird.

Mit einer Kurzcharakterisierung der jeweiligen Unterrichtsreihe und ihrer Einordnung wird bei den Teiluntersuchungen die inhaltliche Dimension des Unterrichts ausgewiesen. Die Gesamtstudie muss auf die gymnasiale Sekundarstufe II beschränkt werden, da bisher nur dort bundesweit ein definierter Lernort für den Informatikunterricht und langjährige unterrichtliche Erfahrung vorliegen.

### Untersuchungsgestaltung und -darstellung

Bei den folgenden Untersuchungen handelt es sich nicht um repräsentative Studien. Vielmehr besteht das Ziel darin, bei Schülerinnen, die nach dem in Kapitel 6 vorgestellten Modulkonzept unterrichtet wurden, das Bild der Informatik herauszufinden. In einer Übersicht werden jeweils die untersuchten Gruppen und die Ergebnisse der Untersuchung vorgestellt. Bestandteil der Darstellung der Ergebnisse ist der Bezug zur Fragestellung. Die Untersuchungen werden in verschiedenen Informatikgrundkursen der gymnasialen Oberstufe<sup>272</sup> durchgeführt (Hinweise auf konkrete Schulen können aus Gründen des Persönlichkeitsschutzes nicht angegeben werden).

<sup>271</sup> vgl. Abschnitt 4.2.2, S. 78

<sup>272</sup> Alle Teiluntersuchungen wurden am Schultyp Gesamtschule vorgenommen, da es sich hier um eine Studie handelt, die als Grundlage zur Hypothesenbildung herangezogen wird.

Insgesamt werden vier Teiluntersuchungen (im Folgenden mit Teil 1... 4 bezeichnet) durchgeführt:

- Teil 1: Im Zusammenhang mit der prädikativen Modellierung wird ein explorativen Ansatz vorgestellt, der von Lehrerinnen für ihren Unterricht übernommen werden kann. Ausgewählte methodische und inhaltliche Elemente des Unterrichts werden von den Schülerinnen dargestellt. Dabei wird die Reflexionsfähigkeit der Schülerinnen gefordert und gestärkt – darüber hinaus werden der Lehrerin Hinweise auf den durchgeführten Unterricht gegeben.
- Teil 2: In Form einer schriftlichen Befragung in fünf Informatikgrundkursen im 11. Jahrgang soll Auskunft über das Bild der Informatik bei Schülerinnen gewonnen werden. Diese Studie realisiert einen breiteren Forschungsansatz, ohne jedoch repräsentativ sein zu können.
- Teil 3: Bei einer der an der schriftlichen Befragung beteiligten Gruppen werden (nach Teil 2) mit Hilfe eines Gruppeninterviews weitere Aussagen zu der Veränderung des Bildes der Informatik gewonnen.
- Teil 4: Schülerinnen die bereits schriftlich befragt wurden (vgl. Teil 2) und weiterhin einen Informatikgrundkurs (einer Schule) besuchen, werden im 12. Jahrgang erneut schriftlich bezüglich des Bildes der Informatik befragt. Damit soll für diese ausgewählten Schülerinnen die Entwicklung/Veränderung des Bildes der Informatik im Laufe eines Jahres deutlich werden.

Aus den vorgestellten Einzelergebnissen werden Hinweise abgeleitet, die einen Beitrag zur Klärung der Frage nach der Veränderung des Bildes der Informatik bei Schülerinnen liefern. Darüber hinaus werden Gestaltungshinweise für den Informatikunterricht ausgewiesen, der die Veränderung des Bildes der Informatik bei den Schülerinnen als Zieldimension berücksichtigt.

## 7.1 Erste Untersuchung – Einzelfallstudie

### Prädikative Modellierung in einem Informatik-Grundkurs im elften/zwölften Jahrgang

Die Schülerinnen setzten sich im Informatikunterricht<sup>273</sup> sowohl mit einer eher geschlossenen Modellierungsaufgabe (Familienbeziehungen) wie auch mit einer offenen Modellierungsaufgabe (Labyrinth<sup>274</sup>) auseinander. Zur Reflexion wurde im Anschluss an die Unterrichtsreihe zum prädikativen Modellieren eine Erhebung durchgeführt und ausgewertet.<sup>275</sup>

#### Dokumentation ausgewählter Antworten

1. Was ist Informatik?  
„Information + Automatik“, „Probleme lösen mithilfe des Computers“, „automatische Verarbeitung von Informationen“, „Informatik ist ... das Lösen und Bearbeiten von Problemen oder Daten mit Hilfe von EDV-Anlagen. Dieses Fach wird auch als Wissenschaft angesehen.“
2. Welche Inhalte des Unterrichts haben dir nicht gefallen?  
„Ahnentafel“<sup>276</sup> (zwei von sechs Gruppen), „Bevor Sie anfangen umfangreich über ... etc. zu erzählen, sollten Sie sich vergewissern, daß auch alle sinnvoll den Computer benutzen können. (Wie schalte ich ihn an?)“, „fachspezifischer Theorieunterricht (sometimes zu trocken)“, „Und bitte versuchen Sie nicht, uns für die Informatik zu begeistern!“ (Anmerkung: Der letzte Satz wurde durchgestrichen.)
3. Welche Inhalte des Unterrichts haben dir besonders gut gefallen?  
„Labyrinth“ (drei von sechs Gruppen), „eigene E-Mail-Adresse“ (drei von sechs Gruppen)

<sup>273</sup> Der Unterricht wurde im Zusammenhang mit der Gestaltung schulpraktischer Studien der ersten (universitären) Phase der Lehrerbildung durchgeführt.

<sup>274</sup> vgl. [Humbert 1999, S. 183]

<sup>275</sup> Die Befragung fand am Ende des ersten Unterrichtshalbjahrs in einem Koppelkurs 11.1 und 12.1 mit insgesamt 25 Schülerinnen statt. Die Fragen (vgl. Anhang B.1, S. 167) wurden den Schülerinnen zur Arbeit in Gruppen von je drei bis fünf Schülerinnen vorgelegt. Die Gruppen hatten die Aufgabe, zu einem Konsens zu gelangen, d. h. bezogen auf die Antworten sich jeweils möglichst auf eine Gruppenposition zu einigen. Jede Gruppe dokumentierte ihre Ergebnisse (Transkripte der Folien: siehe Anhang B.2, S. 167) und stellte sie anschließend im Kurs vor.

<sup>276</sup> Damit sind die o. g. Familienbeziehungen gemeint.

4. Welche Methoden des Unterrichts haben dir nicht gefallen?  
„Ihr Theorieunterricht ist manchmal zu fachspezifisch, dann heben Sie förmlich ab!“, „zu wenig Praxis“, „in der Einzelstunde häufig nur Theorie und keine Arbeit am Computer“, „Einzelarbeit“
5. Welche Methoden des Unterrichts haben dir besonders gut gefallen?  
„viel praktische Arbeit (am Computer)“ (zwei von sechs Gruppen), „Gruppenarbeit“, „... Aufgaben, die wir dann selbstständig lösen mußten“, „Projektarbeit“
6. Sonstiges – Wünsche, Hoffnungen, Fragen  
„Werden wir aus diesem Kurs gehen und mit dem Computer sinnvoll umgehen können?“, „Wir würden gerne noch etwas über das Internet erfahren.“, „Was macht ein Informatiker noch in der Praxis?“, „Wir möchten noch mehr über die Betriebssystemverwaltung wissen.“

Deutlich kristallisieren sich als Ergebnis der Befragung Schwerpunkte heraus: die überwiegende Anzahl der Schülerinnen drückt eine starke Affinität zu selbstständigen, gruppenorientierten Arbeitsformen aus, die nach ihrer Einschätzung praxisorientiert oder gar projektorientiert umgesetzt werden sollen.

Die Positionierung der Schülerinnen bezüglich der Themen aus dem prädikativen Modellierungszusammenhang ist kritisch zu bewerten, da die Beispiele *Ahmentafel* und *Labyrinth* nacheinander bearbeitet wurden und damit dem ersten auch die Rolle der Vermittlung grundlegender Elemente zukam. Verbreitet ist bei den Schülerinnen die Einschätzung, dass Informatikunterricht durchaus praktisch verwertbare Ergebnisse vermitteln soll.

Die Berücksichtigung des Themas „Nutzung verteilter Systeme“ in diesem Kurs, in das vor der Unterrichtsreihe zum prädikativen Modellieren eingeführt wurde, hat nachhaltig Befürwortung gefunden.

Es wird vor allem in den Ausführungen der Schülerinnen zu Punkt 6 (Sonstiges – Wünsche, Hoffnungen, Fragen) deutlich, dass ein Halbjahr Informatikunterricht in der gymnasialen Oberstufe für die Schülerinnen motivierenden Charakter hat und zu erweiterten Fragestellungen Anlass gibt. Dies konnte insbesondere bei den Schülerinnen des 12. Jahrgangs beobachtet werden, d. h. nach drei Halbjahren Informatikunterricht in der Oberstufe stellen die Schülerinnen differenziert konkrete Felder heraus, in denen sie schulisch Handlungsbedarf identifizieren.

Die Schülerinnen trennen in ihrer Reflexion (vgl. vor allem die Antworten zu den Fragen 2 und 4) nicht immer zwischen Inhalt, Methode und der Lehrperson; sie wird von den Schülerinnen gegenstandsüberdeckend gesehen.

## 7.2 Bild der Wissenschaft Informatik bei Schülerinnen

Damit eine über einen Kurs hinausgehende Basis für Aussagen<sup>277</sup> zum Bild der Informatik bei Schülerinnen gewonnen werden kann, werden fünf Informatikgrundkurse der gymnasialen Sekundarstufe II<sup>278</sup> schriftlich befragt. Die Befragung mittels Fragebogen<sup>279</sup> wurde im Schuljahr 1999/2000 bei Schülerinnen des 11. Jahrgangs durchgeführt. Sie wurde von den jeweiligen Kurslehrerinnen organisiert. Die Schülerinnen erhielten zwischen 15 und 20 Minuten, um den Fragebogen zu bearbeiten.

Quantitative Daten: Stichprobenumfang 73 Personen (38 w – weiblich, 35 m – männlich)

Im Folgenden werden Ergebnisse dieser Befragung vorgestellt.<sup>280</sup> Die Ergebnisse werden in der Reihenfolge der Befragung (vgl. Anhang C.1, S. 171) ausführlicher dokumentiert und kommentiert.

<sup>277</sup> Es wird hier nicht der Anspruch erhoben, repräsentative Daten anzugeben.

<sup>278</sup> an Gesamtschulen

<sup>279</sup> Der Fragebogen ist im Anhang C.1 (S. 171) dokumentiert.

<sup>280</sup> Die Detailergebnisse einer kompletten Gruppe sind in Anhang C.2 (S. 174) dokumentiert.

## Das Schulfach Informatik . . .

... fördert Kommunikation mit Hilfe des Schulintranets.	ja 79,5%	nein 20,5%
... fördert die Arbeit in Gruppen.	ja 76,7%	nein 23,3%
... erleichtert die Aktualisierung von Präsentationen.	ja 57,5%	nein 42,5%
... erzwingt das Erlernen komplizierter Sprachen.	ja 47,9%	nein 52,1%

Abbildung 7.2: Bild der Informatik bei Schülerinnen: Das Schulfach Informatik . . .<sup>281</sup>

Immerhin schreiben fast vier von fünf der befragten Schülerinnen dem Schulfach Informatik die Förderung der Kommunikation durch das schulische Intranet zu.<sup>282</sup> Mit einem ähnlich hohen Wert (76,7%) wird die Förderung der Unterrichtsform „Gruppenarbeit“ dem Informatikunterricht zugeordnet. Hingegen stimmen weniger als 60% der Aussage zu, dass die konkrete Kompetenz „Erleichterung der Aktualisierung von Präsentationen“ im Schulfach Informatik gefördert wird. Der Auffassung, dass im Schulfach Informatik „komplizierte Sprachen“ gelernt werden müssen, neigt weniger als die Hälfte der Schülerinnen zu.

Item 1

## Informatik ist. . .

... die Computerwissenschaft.	ja 82,2%	17,8%
... die Lehre von der Bedienung von Computern.	ja 80,8%	19,2%
... die W. zur Rationalisierung der geistigen Arbeit.	ja 46,6%	53,4%
... die Wissenschaft zur Entwicklung von KI.	ja 23,3%	76,7%
... die Wissenschaft für das Internet.	ja 16,4%	83,6%

Abbildung 7.3: Bild der Informatik bei Schülerinnen: Informatik ist. . .

Es wurde den Schülerinnen ermöglicht, neben den vorgeschlagenen Antworten auch eine freie Antwort einzutragen.<sup>283</sup> Davon machten 10 Schülerinnen (13,7%) Gebrauch. Es fällt auf, dass offenbar – aus der Sicht der überwiegenden Mehrheit der befragten Schülerinnen – Informatik mit Computerwissenschaft assoziiert wird. Überraschend ist die breite Akzeptanz der Aussage „Informatik ist die Lehre von der Bedienung von Computern“ mit immerhin über 80% auf Platz 2 (jede Schülerin hatte „drei Stimmen“). Immerhin fast 50% der Schülerinnen geben an, dass sich Informatik (auch) mit Rationalisierungsprozessen beschäftigt. Fast jede vierte Schülerin erkennt einen Zusammenhang zwischen Informatik und künstlicher Intelligenz. Dass jede sechste der befragten Schülerinnen Informatik als „Wissenschaft für das Internet“ charakterisiert, überrascht.

Item 2

Für die vollständige Transkription der Ergebnisse des Items 3: „Für mein persönliches Lernen (dies gilt vor allem für andere Fächer) hat sich durch die Informatische Bildung folgendes verändert . . .“ sei auf den Anhang C.2 (S. 175, 176f) verwiesen. Im Folgenden wird eine Zusammenfassung vorgenommen.

Von einem Großteil der Schülerinnen wurden Aussagen über die Veränderung des persönlichen Lernens, bedingt durch die Informatische Bildung, vorgenommen, wobei diese sich auf ein besseres allgemeines und konkretes Computerverständnis beziehen. Einige Aussagen beziehen sich im „positiven Sinne“ auf den verbesserten Umgang mit Computern und der dadurch erleichterten Arbeit und des Lernens für andere Fächer. Zudem wird der Umgang mit der Informatik als förderlich für die englische Sprache und strukturiertes Denken und Arbeiten, sowie zur Erschließung neuer Interessengebiete angesehen. Außerdem wird

<sup>281</sup> Innerhalb der Abbildung wird die Reihenfolge nach der Anzahl der Nennungen „ja“ absteigend vorgenommen.

<sup>282</sup> Für die folgenden Ausführungen vgl. Abbildung 7.2.

<sup>283</sup> vgl. Fragebogen: Anhang C.1 (S. 171)

von etlichen Schülerinnen auf einen höheren bzw. erweiterten Wissensstand bezüglich aktueller Themen auf Grund des Internets hingewiesen. Im „negativen Sinne“ gaben einige Schülerinnen an, dass keine Veränderung stattgefunden hat und Informatische Bildung für andere Fächer nichts beiträgt.

Überraschende Aussagen lassen aus der Analyse der mehrteiligen Antworten gewinnen. Nicht wenige Schülerinnen sehen durchaus (wenn auch teilweise marginale) Auswirkungen auf andere Fächer: „Komplizierter Denken; Allgemeinwissen verstärkt; *kann man nicht in anderen Fächern gebrauchen*“ wirft ein Licht auf die Einschätzung dieser Schülerin gegenüber den „anderen Fächern“.

	Befragungsergebnis		„zufällige“ Reihenfolge <sup>284</sup>	
	Anzahl	%	Anzahl	%
Ordnung korrekt angegeben	14	19,20	12,2	16,7
eine korrekte Zuordnung	29	39,70	36,5	50,0
alle Zuordnungen falsch	25	34,25	24,3	33,3
keine Angaben	5	6,85	0	0

Tabelle 7.1: Zuordnung der Sichten der Informatik<sup>285</sup> im zeitlichen Verlauf

Die Schülerinnen hatten die Aufgabe, drei (graphisch dargestellte) Sichten auf die Entwicklung der Informatik in eine zeitliche Reihenfolge zu bringen. Den Sichten wurden Veröffentlichungen von 1975, 1983 und 1992/5 zu Grunde gelegt. Die korrekte zeitliche Reihenfolge der drei Sichten auf die Informatik wird von jeder fünften Schülerin<sup>286</sup> angegeben. Der quantitativen Auswertung kann entnommen werden, dass sich die von den Schülerinnen erzielten Ergebnisse nicht wesentlich von einer Zufallsauswahl unterscheiden.

Mit Item 5 („Nehmen Sie Stellung zu den folgenden Behauptungen“) werden den Schülerinnen Entscheidungen über Aussagen zu der Durchdringung verschiedener Lebensbereiche mit Informatiksystemen abverlangt. Die Reihenfolge der ausgewerteten Daten in Abbildung 7.4 entspricht dem Grad der Zustimmung zu den Behauptungen des Fragebogens.

Die Durchdringung aller Lebensbereiche mit Informatiksystemen ist den befragten Schülerinnen deutlich. Bemerkenswert ist die große Übereinstimmung unter den Schülerinnen, dass ein Haus in der Zukunft die Polizei alarmiert, falls jemand einbrechen möchte. Über 90% der Schülerinnen gehen davon aus, dass das papierlose Büro innerhalb der nächsten Jahre realisiert wird. Mehr als  $\frac{3}{4}$  der Befragten ist die Tatsache bekannt, dass in Autos eine große Zahl von Informatiksystemen verbaut werden.<sup>287</sup> Die Einschätzung bezüglich der menschlichen Intelligenz und der möglichen Konkurrenz durch Informatiksysteme überrascht dagegen. Ist es im Informatikunterricht nicht gelungen, den Schülerinnen zu verdeutlichen, wo die grundlegenden Grenzen von Informatiksystemen liegen? Da die Ansätze der Schachcomputer inzwischen verfeinert wurden, ist die von fast  $\frac{4}{5}$  der Schülerinnen angegebene Auffassung, dass Schachcomputer in Zukunft jeden menschlichen Gegner schlagen, nicht verwunderlich. Das inzwischen klar sein sollte, dass „die Kapazität von fünf Computern für alle Berechnungen auf der ganzen Welt“ nicht ausreicht, ist offenbar nicht bis zu allen Schülerinnen vorgedrungen oder diese Behauptung wurde von einigen Schülerinnen nicht verstanden.

<sup>284</sup> Es gibt sechs Möglichkeiten der Anordnung der drei Sichten. Bei einer Zufallsauswahl ist Wahrscheinlichkeit des Ereignisses (vollständig) richtige Reihenfolge  $\frac{1}{6}$ . Es gibt genau zwei Möglichkeiten, in der Folge „keinen Treffer“ zu plazieren  $\frac{1}{3}$ .

<sup>285</sup> Die zugehörige Abbildung 4 findet sich im Fragebogen (vgl. Anhang C, S. 172). Die Sichten stammen von PETRI (siehe [Petri 1983]), BRAUER (nach dem Artikel [Brauer und Brauer 1995] vom Autor angefertigt) und GENRICH (siehe [Genrich 1975b]).

<sup>286</sup> vgl. Tabelle 7.1

<sup>287</sup> Hier soll kritisch angemerkt werden, dass „quizerfahrene“ Schülerinnen durch die Formulierung der Behauptung eher zu einer Zustimmung neigen.

Nehmen Sie Stellung zu den folgenden Behauptungen

Das Haus alarmiert die Polizei . . . .	ja 93,2%	6,8%
. . . wird es das papierlose Büro geben.	ja 91,8%	8,2%
Schachcomputer schlagen jeden . . .	ja 79,5%	20,5%
In einem Auto befinden sich ca. 70 Informatiksysteme.	ja 76,7%	16,4% 6,9
. . . Konkurrenz für die Intelligenz der Menschen.	ja 76,7%	23,3%
. . . fünf Computer reichen . . . völlig aus. . . .	ja 45,2%	50,7% 4,1

Abbildung 7.4: Bild der Informatik bei Schülerinnen: Position zu Behauptungen <sup>288</sup>

### 7.3 Schülergruppeninterview

#### Einschätzungen der Informatik

Dokumentation von Ergebnissen einer qualitativen Beobachtung (Schülergruppeninterview) im Februar 2000. Im Folgenden werden ausgewählte Antworten aus dem Transkript eines 30-minütigen Gruppeninterviews einer Mitarbeiterin der Informatikdidaktikgruppe der Universität Dortmund mit acht Schülern (darunter eine Schülerin) eines aus 18 Schülerinnen bestehenden Informatikgrundkurses dokumentiert. <sup>289</sup> Der Leitfaden für das Interview ist in Anhang D.1 (S. 179) dokumentiert. Im Folgenden werden für den hier betrachteten Zusammenhang Ausschnitte aus dem Transkript des Interviews <sup>290</sup> dokumentiert.

#### Warum haben Sie Informatik als Fach gewählt?

Ich habe mal ein Praktikum an der Berufsschule als Informationstechnischer Assistent gemacht, und ich fand das . . . interessant. Damit ich weiß, wie ich so einen Lebenslauf, so eine Tabelle . . . schreiben kann, . . . ich wollte früher Industriekaufmann werden . . . darum habe ich auch Informatik gewählt, damit ich später nicht auf die Nase falle.  
Mathe Leistungskurs ist gekoppelt an Informatik, . . . ich wollte lernen, richtig mit Computern umzugehen, . . . dass ich vielleicht einmal aus dieser Abhängigkeit von Computern rauskommen kann.

#### Was bedeutet Informatik für Ihre Eltern?

Wenn die Informatik hören, dann denken die eh nur Computer und Rechnen und Tippen . . . für meinen Vater, der braucht den Computer für den Beruf . . .  
Wir haben zuhause einen PC, und wenn ich mal Probleme damit habe, dann frage ich meinen Vater, „Ja Papa, kannst du mir da mal helfen?“ Da sagt er nur zu mir „Wieso, du hast doch Informatik!“ Ich sag’ „Ja, da machen wir aber was ganz anderes, das hilft mir hierbei nicht“ . . .  
Ich bin von klein auf mit Computern aufgewachsen, wir hatten immer mindestens zwei Stück zuhause, weil meine Mutter die für die Arbeit immer gebraucht hat, und mit Computern, bei uns kennt sich eigentlich jeder damit aus, nur ich mittlerweile besser als meine Mutter, weil die ganze Technik und so hat sie nicht so im Griff. . . .

#### Stellenwert/Bedeutung der Informatik

Auf eine Art könnte man Informatik als eine Erleichterung in beruflicher Hinsicht sehen, . . . Was wir jetzt lernen, da nützt uns später überhaupt nichts, weil die dann wieder so weit sein werden und dann können wir mit dem, was wir hier gelernt haben, überhaupt nichts mehr anfangen . . .  
Jeden Tag werden . . . bessere Computer erfunden. . . was man gerade denkt, der macht das dann sofort, führt das dann sofort aus, dann braucht man nicht dann erst stundenlang irgendwas eingeben, bevor das überhaupt alles läuft . . .  
Informatik ist . . . wenn man ein Problem hat, und man löst das Problem mithilfe von Computern. Also, Informatik hat immer was mit Computern zu tun, immer was mit ständigem Fortschritt, sodass man gar nicht mehr überblicken kann . . . das ist so ein richtiger Fortschritt und man kann sich dem gar nicht mehr richtig anpassen.

<sup>288</sup> Legende (in den Grafik von links nach rechts angeordnet):

-  kumulierter Anteil der Schülerinnen, die **eingetroffen** oder **möglich** angegeben haben (um die Trennschärfe zu erhöhen)
-  Schülerinnen (Angabe in Prozent) die angegeben haben: **unmöglich/falsch**
-  bezeichnet den Anteil der Schülerinnen, die keine Angabe zu der Frage gemacht haben (in Prozent)

<sup>289</sup> Die zweite Gruppe (die ausschließlich aus den Schülerinnen bestehen sollte), konnte aus organisatorischen Gründen nicht interviewt werden.

<sup>290</sup> Zusammenhängende Transkriptausschnitte sind in Anhang D.2 (S. 180) dokumentiert.

Informatik ist zu schnell, Informatik ist Schnelligkeit, Informatik ist Fortschritt. Aber ... das ist nicht unbedingt nützlich. ..., das wächst halt immer, Pentium I, II, III und das wird auch immer ständig wachsen, und das ist viel zu schnell für uns, das ist ... etwas Unkontrolliertes

#### **Informatikunterricht: Nutzen für andere Fächer?**

Ich kann jetzt schneller schreiben<sup>291</sup>. Aber sonst für andere Fächer hat der Informatikunterricht nichts gebracht ...

Information aus dem Internet holen. ... Früher, da war das viel aufwändiger: Bücherei, Buch ausleihen, lesen, „Das ist das falsche, geh' nochmal hin, hol' mir ein Neues!“

Wir gehen weiter in die Unmündigkeit rein.

In der Informatik, wenn man das nicht genauso schreibt, wie man es schreiben soll, dann funktioniert es halt nicht.

#### **Informatik und Zukunft – Gedankensplitter der Schülerinnen**

Dass Informatik nicht lebenswichtig ist, sondern das Leben ... erleichtert, aber es wird nie lebenswichtig sein.

Was wäre ein Atomkraftwerk ohne Computer?

Die nächste Zukunft, also die nächste Generation der Leute, die werden das ganz anders sehen, wir wissen jetzt noch, wie das geht.

... Man versucht immer alles so einfach wie möglich zu machen. ... Warum soll man dann nicht den leichten Weg gehen?

Stell' dir mal eine Zukunft vor, wo man nicht mehr arbeiten braucht, wo das die ganzen Computer und Roboter dann schon tun.

Es gibt doch jetzt World Wide Web, Netze, kleine Server, das ist ja nicht mehr ein Computer nur, der geht ja an viele andere. Das Problem ist dann dabei wieder, wenn man auf den Server zugreifen kann, kann man auch auf die anderen zugreifen. Und dann, kommt drauf an, wer davor sitzt, dann ist der halt das Ding, das denkt. ... Irgendwann machen die Computer was für sich alleine, und bestimmen uns und nicht wir sie.

Dass künstliche Intelligenz in dem Sinne uns überwuchert oder so – quasi wie so ein Virus oder so ...

Die Computer werden eh irgendwann die Welt beherrschen und uns auslöschen.

Die Schüler sprechen der Informatik Zukunftsbedeutung zu. Sie erwarten vom Informatikunterricht die Ausbildung konkreter Fertigkeiten „wie ich so einen Lebenslauf, so eine Tabelle ... schreiben kann; richtig mit Computern umzugehen“.

Sie trennen zwischen dem Teilgebiet „vernetzte Systeme“ – mit dem sie offenbar handelnd umzugehen gelernt haben – und allgemeinen Problemlöseprozessen, bei denen sie sich nicht in der Lage sehen, mit der weiteren Entwicklung Schritt zu halten.

Im Hintergrund einiger Antworten stehen Einschätzungen über die individuelle und gesellschaftliche Bedeutung der Informatik; sie sind durch verhaltene Hoffnungen und Unsicherheit geprägt.

## **7.4 Bild der Wissenschaft Informatik bei Schülerinnen – Längsschnitt**

Die im 11. Jahrgang durchgeführte Befragung (vgl. Abschnitt 7.2) wird bei Schülerinnen im 12. Jahrgang wiederholt. Die erneute Befragung soll dazu dienen, Änderungen bezüglich der Einstellungen zur Informatik herauszufinden.

Befragungszeitpunkte: Schuljahr 1999/2000 (11. Jahrgang) und Schuljahr 2000/2001 (12. Jahrgang)

Fach: Informatik

Kursart: Grundkurs

Stichprobenumfang: 16 Personen (6 w – weiblich, 10 m – männlich)

Für diese Schülerinnen werden Ergebnisse der Auswertung des Längsschnitts vorgestellt.<sup>292</sup>

Die Darstellung der Ergebnisse zur Einschätzung des Schulfachs Informatik (vgl. Abbildung 7.5) verdeutlicht in wesentlichen Teilen die Stabilität in der Einstellung der Schülerinnen. So werden die bereits in der Befragung im 11. Jahrgang positiv vermerkten Zuordnungen zur informatischen Bildung – bezogen auf das schulische Intranet, aber auch die Unterrichtsmethode „Gruppenarbeit“ – bestätigt, eine Schülerin (von 11 nach 12) entscheidet sich bezüglich der Zuordnung der Wertigkeit des Intranet anders.<sup>293</sup> Dennoch liegt der Anteil (nach dem 12. Jahrgang) über dem Durchschnitt in der breiteren Untersuchung (vgl. Abbildung 7.3).

<sup>291</sup> hier ist Tippen (engl. keyboarding) gemeint.

<sup>292</sup> Da die Anzahl der im Längsschnitt Befragten gering ist, werden in den folgenden Abbildungen über die prozentualen Angaben hinaus auch die absoluten Zahlen dokumentiert.

<sup>293</sup> Im 12. Jahrgang wurde dieser Bereich nicht explizit als Unterrichtsgegenstand thematisiert, sondern im Kontext konkreter Aufgabenstellungen in den Unterricht einbezogen.

## Das Schulfach Informatik ...

... fördert Kommunikation mit Hilfe des Schulintranets.	11. Jahrgang – 14 ja	87,5%	12,5%
	12. Jahrgang – 13 ja	81,25%	18,75%
... fördert die Arbeit in Gruppen.	11. Jahrgang – 12 ja	75%	25%
	12. Jahrgang – 12 ja	75%	25%
... erleichtert die Aktualisierung von Präsentationen.	11. Jahrgang – 8 ja	50%	50%
	12. Jahrgang – 7 ja	43,75%	56,25%
... erzwingt das Erlernen komplizierter Sprachen.	11. Jahrgang – 10 ja	62,5%	37,5%
	12. Jahrgang – 9 ja	56,25%	43,75%

Abbildung 7.5: Schülerinnen im Längsschnitt: Das Schulfach Informatik ...

Die Stabilität der Einschätzung zur Gruppenarbeit weist darauf hin, dass diese Unterrichtsform offenbar weiterhin häufig im Informatikunterricht eingesetzt wurde. Die mit Präsentationen und ihrer Aktualisierung zusammenhängenden Ergebnisse, werden von der Hälfte der Schülerinnen dieser Gruppe im 11. Jahrgangs dem Informatikunterricht zugeschrieben – am Ende des 12. Jahrgangs entscheidet sich hier eine Befragten anders.

## Informatik ist ...

... die Computerwissenschaft.	11. Jahrgang – 14 ja	87,5%	12,5%
	12. Jahrgang – 13 ja	81,25%	18,75%
... die Lehre von der Bedienung von Computern.	11. Jahrgang – 12 ja	75%	25%
	12. Jahrgang – 9 ja	56,25%	43,75%
... die Wissenschaft zur Entwicklung von KI.	11. Jg. 3 ja		81,25%
	12. Jahrgang – 7 ja	43,75%	56,25%
... die Wissenschaft für das Internet.	11. Jg. 5 ja	31,25%	68,75%
	12. Jg. – 6 ja	37,5%	62,5%
... die W. zur Rationalisierung der geistigen Arbeit.	11. Jg. – 6 ja	37,5%	62,5%
	12. Jg. 4 ja	25%	75%

Abbildung 7.6: Schülerinnen im Längsschnitt: Informatik ist ...

Die Ergebnisse zur Begriffsbestimmung/Definition der Informatik zeigen Änderungen, die in Abbildung 7.6 in vergleichender Form dargestellt sind. Die Reihenfolge entspricht der Anzahl der Nennungen am Ende des 12. Jahrgangs.

Bemerkenswert ist die Stabilität der von den Schülerinnen ausgewählten „Definitionen“. Dies trifft vor allem für das Item „Computerwissenschaft“ zu. Die größten Veränderungen stellt die Zunahme der Zustimmung (+4 Schülerinnen) zu der Definition „Informatik die Wissenschaft zur Entwicklung von KI“ und der Rückgang der Zustimmung zu „Informatik ist die Lehre von der Bedienung von Computern“ (-3 Schülerinnen) dar.

Gegenüber der im 11. Jahrgang befragten Gesamtgruppe (vgl. Abbildung 7.3) ist eine Schwerpunktverschiebung festzustellen. Dort fand sich eine Zustimmung von über 80% zu der Definition „Informatik ist die Lehre von der Bedienung von Computern“ – hier ist die Zustimmung (nach dem 12. Jahrgang) auf 56,25% zusammengeschmolzen.

	11. Jahrgang – Befragungsergebnis		12. Jahrgang – Befragungsergebnis		„zufällige“ Reihenfolge	
	Anzahl	%	Anzahl	%	Anzahl	%
Ordnung korrekt angegeben	3	18,75	5	31,25	2,67	16,7
eine korrekte Zuordnung	10	62,50	8	50,00	8,00	50,0
alle Zuordnungen falsch	3	18,75	3	18,75	5,33	33,3

Tabelle 7.2: Schülerinnen im Längsschnitt: Zuordnung der Sichten der Informatik im zeitlichen Verlauf

Bei der Ordnung der zeitlichen Abfolge der Sichten auf die Informatik durch die Schülerinnen kann eine geringfügige Verbesserung der Ergebnisse festgestellt werden (vgl. Tabelle 7.2).

Nehmen Sie Stellung zu den folgenden Behauptungen

Das Haus alarmiert die Polizei. . . .	11. Jahrgang – 16 ja	100%	
	12. Jahrgang – 16 ja	100%	
. . . wird es das papierlose Büro geben.	11. Jahrgang – 15 ja	93,8%	6,2%
	12. Jahrgang – 14 ja	87,5%	12,5%
In einem Auto befinden sich ca. 70 Informatiksysteme.	11. Jahrgang – 14 ja	87,5%	12,5%
	12. Jahrgang – 13 ja	81,25%	18,75%
. . . Konkurrenz für die Intelligenz der Menschen.	11. Jahrgang – 11 ja	68,75%	31,25%
	12. Jahrgang – 12 ja	75%	25%
Schachcomputer schlagen jeden . . . .	11. Jahrgang – 12 ja	75%	25%
	12. Jahrgang – 11 ja	68,75%	31,25%
. . . fünf Computer reichen . . . völlig aus . . . .	11. Jahrgang – 8 ja	50%	50%
	12. Jg. – 6 ja	37,5%	62,5%

Abbildung 7.7: Schülerinnen im Längsschnitt: Position zu Behauptungen. . .

Die Ergebnisse des Längsschnitts bezüglich der Behauptungen über den Einsatz und die Qualität von Informatiksystemen (vgl. Abbildung 7.7) zeigen kaum Veränderungen.

## 7.5 Auswertung und Zusammenfassung der Ergebnisse

Im Folgenden werden die Ergebnisse der Teiluntersuchungen zusammenfassend dargestellt.

### Bild der Informatik

Definitionen von Informatik lassen sich, wie im ersten Befragungsergebnis (vgl. Abschnitt 7.1) dokumentiert, direkt „abfragen“ oder indirekt erkunden. Ergebnis ist ein Bild der Informatik, das im ersten Fall eine „gelernte“ Definition wiedergibt, wohingegen in der offeneren Befragungssituation vielschichtiger Positionen erkennbar werden. Dies zeigt sich vor allem bei den Antworten zu der Fragestellung „Was bedeutet Informatik für Ihre Eltern?“ (vgl. Abschnitt 7.3, S. 120). Die Sicht der Schülerinnen auf die Informatik wird durch die mit Informatiksystemen arbeitenden Eltern sowie durch Massenmedien und eigene oder vermittelte Erfahrungen mit Computerspielen (Gameboy, . . .) beeinflusst.

### Inhalte

Die dokumentierten Ergebnisse machen deutlich, dass die Schülerinnen die handlungsermöglichenden Elemente des Informatikunterrichts, die ihnen einen unmittelbaren Nutzen bringen (z. B. schuleigene Mailadresse), positiv bewerten und „mehr davon“ haben möchten.

Sehr deutlich zeigt sich das Problem der nicht abzuschätzenden Grenzen der Informatik. Gerade im offenen Interview verleihen die Schülerinnen ihrer Befürchtung Ausdruck, den Prozess der Weiterentwicklung nicht verfolgen zu können, sondern dieser Entwicklung ausgeliefert zu sein.

### Methoden

Auf der methodischen Ebene wird deutlich, dass die in der Schule eher selten praktizierten Elemente<sup>294</sup>, die zum Erwerb und zur Verbesserung der Teamarbeit, Gruppenarbeit, Projektarbeit beitragen, von den Schülerinnen deutlich präferiert werden. Hier deutet sich – auch durch die Ergebnisse des Längsschnitts gestützt – an, dass die Gruppenarbeit einen wichtigen Platz im Informatikunterricht hat.

### Perspektiven

Auf dem Hintergrund der dargestellten Untersuchungen wird eine Verdichtung vorgenommen und auf erste Schlüsse für Gestaltungshinweise eines Ausbildungskonzepts hingewiesen. Grundlage der Untersuchungen war die Frage nach der Herausbildung eines Bildes der Informatik bei den Schülerinnen. Das Bild entwickelt sich sowohl durch den Informatikunterricht, aber auch auf dem Erfahrungshintergrund der Schülerinnen außerhalb des Informatikunterrichts. Dem Informatikunterricht kommt eine wichtige Bedeutung zu, indem er Korrektiv für falsche Vorstellungen, Erwartungen und Ängste ist. Es konnte herausgefunden werden, dass Schülerinnen z. B. Sorge vor der Zukunft zum Ausdruck bringen, weil sie sich nicht in der Lage sehen, abzuschätzen, dass es prinzipiell unlösbare Problemklassen gibt. Im Hintergrund einiger weniger Antworten stehen Einschätzungen über die individuelle und gesellschaftliche Zukunftsbedeutung der Informatik, die durch verhaltene Hoffnungen, aber auch durch Unsicherheit geprägt sind.<sup>295</sup>

Zusammen mit einem reflektierten Bild der Wissenschaft Informatik werden für die Schülerinnen zukünftige Entwicklungen einschätzbar. Damit erfährt die Herausbildung des Bildes eine Bedeutung für die individuelle Lebensgestaltung. Für die dokumentierten Ergebnisse wurde ein Grundkurs im Rahmen des Unterrichts um Rückmeldungen zum Unterricht gebeten, fünf Informatikgrundkurse schriftlich befragt – mit einem halben Grundkurs wurde ein leitfadengesteuertes Gruppeninterview durchgeführt und transkribiert. Im Längsschnitt wurde eine weitere schriftliche Befragung durchgeführt, in der Einstellungsänderungen bei den Schülerinnen untersucht wurden. Änderungen konnten nachgewiesen werden, wobei zu bemerken ist, dass diese nicht in dem erwarteten Umfang erzielt wurden. Bei der im Längsschnitt untersuchten Gruppe konturiert sich im Laufe der Zeit ein zunehmend reflektiertes Bild der Informatik. Darüber hinaus wird der Informatikunterricht im Verlauf der Zeit nach und nach anders eingeschätzt. Stabile Werte auf hohem Niveau – bezogen auf die Unterrichtsform Gruppenarbeit und die Unterstützung von Kommunikationsprozessen durch Informatiksysteme – geben Hinweise auf erfolgreiche Elemente dieses Informatikunterrichts. Bei den im Längsschnitt befragten Schülerinnen werden Probleme bei der Definition von Informatik deutlich: Die „Definition“ „Informatik ist die Lehre von der Bedienung des Computers“ findet sich nach zwei Jahren Informatikunterricht mit über 50% Zustimmung (nach „... Computerwissenschaft“) an der zweiten Stelle der angebotenen Definitionsvorschläge. Selbst die „Definition“ „Informatik ist die Wissenschaft für das Internet“ erreicht (mit 37,5% Zustimmung) noch den vierten Platz.

<sup>294</sup> Kommunikationsunterstützung (auch) durch Informatiksysteme; selbstständige, handelnde Erarbeitung und Vertiefung von Inhalten an Hand offener Problemstellungen (zur quantitativen Darstellung der verschiedenen Unterrichtsformen vgl. Abschnitt 3.5, S. 50).

<sup>295</sup> Hierbei sollte berücksichtigt werden, dass in dieser Altersstufe bei Jugendlichen/Heranwachsenden unabhängig von konkreten Unterrichtsfächern eine Orientierung auf ihre je individuelle sich entwickelnde „Weltsicht“ festgestellt werden kann (vgl. [Fischer u. a. 2000]).

## 7.6 Überlegungen zur Vermittlung des Bildes der Informatik

Bezogen auf das Bild der Informatik gilt es im Unterricht die auf das konkrete Problem bezogenen inhaltlichen Elemente der Informatik, wie auch die zum Einsatz kommenden Methoden der Informatik deutlich zu machen und in einen den Schülerinnen angemessenen historischen Kontext zu stellen.

Wie können die verschiedenen Ausprägungen des Bildes der Wissenschaft Informatik in unterrichtliche Konzepte integriert und für die Schülerinnen handelnd erfahrbar gemacht werden? Als zentrales methodisches Prinzip des Schulfachs Informatik ist die Problemorientierung anerkannt. Konkreter Unterricht im Schulfach Informatik zeichnet sich durch die geeignete Organisation der Lehr- und Lernprozesse in der Verzahnung der Ebenen „Theorie“ (= Erarbeitung, Problemgewinnung, Modellierungsphasen: Analyse, Problemlösung) und Umsetzung der entwickelten Lösungsmöglichkeiten (= Modellierungsphasen: Implementierung, Test, Verschrottung) an den schuleigenen Informatiksystemen (dies umfasst die dazu notwendige Intranetstruktur der Schule als Handlungsrückgrat) aus.

In den vorgestellten Ergebnissen wird deutlich, dass aus der Sicht der Schülerinnen der konkrete Nutzen eine hohe Wertschätzung erfährt. Aus dieser Motivation wählen nach wie vor viele Schülerinnen einen Informatikkurs im 11. Jahrgang. Die Verbindung der o. g. Ebenen sollte für die Schülerinnen transparent erfolgen, damit sie die Notwendigkeit der Theoriebildung erfahren und neue Einsichten gewinnen können.

In der konkreten Unterrichtsplanung und -durchführung finden sich häufig Methoden zur inhaltlichen Erarbeitung durch Partner- oder Gruppenarbeit. Diese Einschätzung wird in der Studie aus Sicht der Schülerinnen bestätigt und erweist sich im Längsschnitt als stabil. Da dem projektorientierten Informatikunterricht eine große Bedeutung zugemessen werden muss, sind projektorientierte Ansätze im Zusammenhang mit Unterrichtsreihen keine Ausnahme.

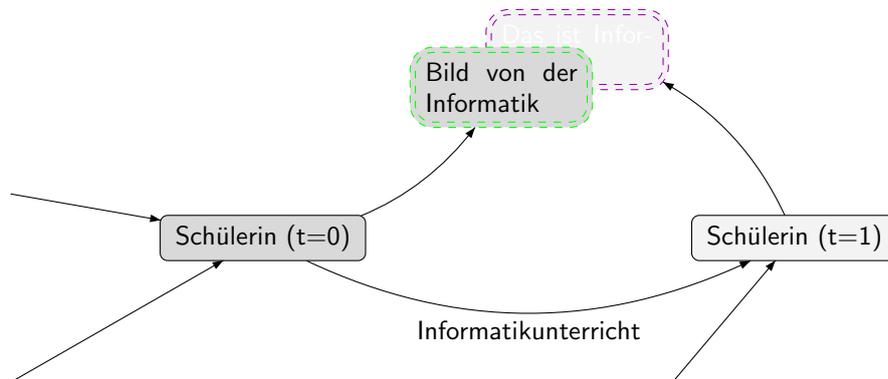


Abbildung 7.8: Änderung des Bildes der Informatik bei Schülerinnen (schematisch)

Bezogen auf die Änderung des Bildes der Informatik bei Schülerinnen zeigte sich eine begrenzte Einflussmöglichkeit des Informatikunterrichts. Das Ergebnis wird durch die Abbildung 7.8 illustriert. Die im 11. Jahrgang ( $t=0$ ) befragten Items werden als „Bild von der Informatik“ ausgewiesen. Durch die Teilnahme am Informatikunterricht über den Zeitraum von einem Schuljahr (12. Jahrgang –  $t=1$ ), ergab sich bei der erneuten Befragung mit den selben Items ein (wenn auch geringfügig) geändertes Bild der Informatik. In der Abbildung wird das geänderte Bild „hinter“ das zuerst erkundete Bild gelegt, da die Änderungen quantitativ gering ausfallen und die Vermutung angebracht erscheint, dass der Einfluss des Informatikunterrichts als relativ gering einzuschätzen ist. Durch die stärkere explizite Berücksichtigung dieses Themas sollte im Unterricht der Herausbildung eines begründeten Bildes der Informatik bei den Schülerinnen Rechnung getragen werden.



## Kapitel 8

# Evaluation des Informatikunterrichts

### 8.1 Einordnung in den Forschungsgang

Die Einführung des Modulkonzepts (vgl. Kapitel 6) impliziert eine neue Struktur des Informatikunterrichts, die mit Blick auf die allgemein bildende Sekundarstufe II entwickelt wurde. Die Durchdringung der Gesellschaft mit Informatiksystemen macht vor der Schule nicht halt. Damit sind den Schülerinnen vor Beginn des Informatikunterrichts Elemente der Informatik durch praktischen Umgang mit Informatiksystemen in anderen Unterrichtsfächern und außerhalb der Schule in Form von ([fachlich] unreflektierten) Handlungsroutinen bekannt. Dem Informatikunterricht kommt u. a. die Aufgabe zu, die fachlich fundierte Reflexion dieser Erfahrungen zu ermöglichen. Um das Konzept in der Sekundarstufe II umzusetzen, sind dazu innerhalb der zur Verfügung stehenden Unterrichtszeit viele Details zu berücksichtigen. Zunehmend findet der Einsatz von Informatiksystemen in vielen Fächern bereits in der Sekundarstufe I statt. Ein Problem besteht darin, dass die Unterrichtsfächer, in denen Informatiksysteme eingesetzt werden, bisher auf eine fachliche Basis für diesen Einsatz verzichten (müssen).

Ein Ausweg aus dieser Situation besteht darin, Informatik als verpflichtendes Unterrichtsfach in der Sekundarstufe I zu verankern. Einige Bundesländer haben damit begonnen, Überlegungen in dieser Richtung umzusetzen. Die Fachgruppe Didaktik der Informatik der Universität Dortmund hatte im Schuljahr 2000/2001 die Möglichkeit, verpflichtenden Informatikunterricht in der Sekundarstufe I im Bundesland Bayern (Schultyp: Gymnasium) im Auftrag der Bayerischen Staatsregierung zu untersuchen. Mit Blick auf das Modulkonzept eröffnet sich damit die Möglichkeit, ausgewählte Elemente auf ihre Übertragbarkeit in die Sekundarstufe I zu prüfen. Dazu ist die Frage zu untersuchen, welche Teile des Modulkonzepts geeignet sind, in der Sekundarstufe I grundlegend bearbeitet zu werden.

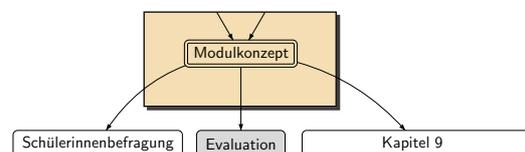


Abbildung 8.1: Struktur Kap. 6–8

Die Untersuchung der unterrichtlichen Umsetzung informatischer Konzepte (vgl. Kapitel 7) macht deutlich, dass in der Aufarbeitung der durch die veränderten Anforderungen an die Informatik in der modernen Gesellschaft und in der Schule Hinweise für geänderte Bildungsanforderungen gefunden werden können. Schülerinnen formulieren Anforderungen an dieses Fach, die nicht eingelöst werden [können], aber dennoch eine Herausforderung für die Unterrichtenden darstellen. Die Balance zwischen der Orientierung auf die von den Schülerinnen gewünschte Verwertbarkeit der unterrichtlichen Ergebnisse und allgemeinen ordnenden und strukturierenden Inhalten (von den Schülerinnen als „Theorie“ bezeichnet<sup>296</sup>) scheint nur schwer erreichbar zu sein; jedoch kann die Gestaltung der Unterrichtssequenzen so vorgenommen werden, dass neben der Erfüllung der institutionellen Zielvorgabe (Wissenschaftspropädeutik) auch die Schülerinteressen Berücksichtigung finden.

<sup>296</sup> vgl. exemplarisch: Kapitel 7, S. 116

Bei dieser Betrachtung zeichnet sich „guter Informatikunterricht“ durch die Verschränkung der in Abschnitt 3.3 (S. 51) beschriebenen Anforderungen aus.

## 8.2 Evaluation – zum Begriff<sup>297</sup>

Der Begriff Evaluation stammt aus dem Lateinischen und fand seinen Weg über den französischen und angloamerikanischen Sprachraum erst in der zweiten Hälfte des vorigen Jahrhunderts im Zuge der Autonomisierung und Ausdifferenzierung der Erziehungswissenschaft über die Vereinigten Staaten von Amerika nach Deutschland. Offensichtlich wird dieser Begriff in der theoriepolitischen Auseinandersetzung zwischen geisteswissenschaftlicher Pädagogik und kritischer Erziehungswissenschaft auf der einen Seite und der empirischen Erziehungswissenschaft auf der anderen Seite instrumentalisiert (vgl. [Wulf 1972], [Tenorth 2000]). Da über Sprache immer auch Herrschaft ausgeübt wird, scheint es hilfreich zu sein, sich dieser theoriepolitischen Implikationen zu vergewissern und die in diesem Kontext getroffenen Differenzierungen auf ihren möglichen Nutzen zu überprüfen. Die Rechenschaft über die sinnhafte und zielgerichtete Verwendung gesellschaftlicher Ressourcen wurzelt in der kalvinistischen Tradition. Gepaart mit den Entwicklungen der empirischen Sozialforschung wurde so in Europa<sup>298</sup> eine Tradition der Evaluation bezogen auf Lehr-/Lernprozesse entwickelt (vgl. [Daxner 1996]).

Im Folgenden wird Evaluation im Sinne von „sach- und fachgerechter Einschätzung, Bewertung, Beurteilung von Forschungsvorhaben und -programmen ... (durch eine beauftragte Behörde, eine unabhängige Kommission o. ä.) ...“ (vgl. [Nortmeyer 2002]) verwendet. Es ist nicht beabsichtigt, einzelne unterrichtliche Situationen zum Anlass der Evaluation der individuellen Arbeit von Informatiklehrerinnen heranzuziehen. Vielmehr besteht das Ziel in der Untersuchung der Tragfähigkeit eines Konzepts. Im Rahmen dieser Arbeit werden darüber hinaus Verbindungen zwischen dem Konzept des bayerischen Schulversuchs zum Pflichtfach Informatik und dem in dieser Arbeit entwickelten Modulkonzept (vgl. Kapitel 6) untersucht.

## 8.3 Begleitende Untersuchungen zur Einführung des Pflichtfachs Informatik in Bayern

Die Einführung des Pflichtfachs Informatik mit zwei Unterrichtsstunden für alle Gymnasien in Bayern in der 6. Jahrgangsstufe beginnt 2003 (vgl. [Hubwieser 1999]). Grundlage der unterrichtlichen Gestaltung stellen Erfahrungen dar, die in Schulversuchen an 20 Gymnasien gewonnen wurden. Diese Erfahrungen wurden mit der Umsetzung im Europäischen Gymnasium (Typ III) [Hechenleitner 2001] gesammelt. Die bayerische Staatsregierung beauftragte die Fachgruppe Didaktik der Informatik der Universität Dortmund mit der externen Evaluation des Informatikunterrichts in diesem Schulversuch.

Ziel der externen Evaluation ist es, den Unterrichtsprozess und seine Rahmenbedingungen daraufhin zu untersuchen, ob die durch die Gesamtkonzeption intendierten Ziele für konkrete Lerngruppen umsetzbar sind. Es ist nicht Ziel dieser Evaluation, die Vielgestaltigkeit des Unterrichtsgeschehens abzubilden. Vielmehr gilt es herauszuarbeiten, welcher Gestaltungsrahmen fachlicher Art pädagogisch und unter Zuhilfenahme lerngruppenbezogener, thematisch angemessener und unterstützender Medien umgesetzt wird. Als Grundlage des Unterrichts der beobachteten Lerngruppen des 6. Jahrgangs wird unter der Überschrift „Dokumente und Objekte“ zusammenfassend in [Frey u. a. 2001, S. 20f] formuliert:

„Die Schülerinnen [...] arbeiten vor allem mit elektronischen Dokumenten, die sie mit Hilfe einer Reihe verschiedener Informatiksysteme erstellen und bearbeiten. Sie gestalten Grafiken und Texte, verwalten Dokumente mit Hilfe von Ordnersystemen, transportieren Dokumente mit elektronischer Post, verknüpfen sie zu Hypertextstrukturen [...] und erhalten erste Einblicke in die automatische Informationsverarbeitung [...]. So lernen die Kinder einerseits

<sup>297</sup> zur Geschichte des Begriffs Evaluation vgl. [Karbach 1998], [Nortmeyer 2002]

<sup>298</sup> genauer in den Niederlanden und im Norden Englands

die Funktionsweise wichtiger Informatiksysteme, andererseits Grundprinzipien der Digitalisierung, Vernetzung und automatischen Verarbeitung von Information kennen. Die Analyse (Modellierung) der bearbeiteten Dokumente zeigt dabei, dass diese jeweils aus bestimmten typischen Objekten mit charakteristischen Eigenschaften zusammengesetzt sind. Ein Textdokument enthält beispielsweise Absätze mit einem bestimmten Rand und Zeilenabstand, diese wiederum Zeichen mit bestimmter Größe und Schriftart. Mit dieser Sichtweise können die Schülerinnen [...] unabhängig von der jeweils verwendeten Software eine Vielzahl von Phänomenen im Zusammenhang mit Informatiksystemen verstehen und systematisieren. Insbesondere ermöglicht sie es den Schülerinnen [...], sich bei der Manipulation der Dokumente unabhängig von der (schnellebigen und produktspezifischen) Bedieneroberfläche des jeweiligen Werkzeugs an einer "inneren" Struktur der Dokumente zu orientieren.“

**Darstellung von Information mit Hilfe von Dokumenten**

- ▷ Datenstrukturen von Vektorgrafik- und Textdokumenten
- ▷ Grundlegende Operationen auf Objekten: Markieren, Ausschneiden, Einfügen
- ▷ Kombination von Graphik und Text

**Verwaltung von Dokumenten**

- ▷ Dokumente als Dateien, Löschen und Kopieren von Dateien
- ▷ Ordner zur hierarchischen Verwaltung von Dateien, beispielhafte Ordnersysteme, Verschieben und Kopieren von Dateien und Unterordnern

**Versand von Dokumenten**

- ▷ Struktur elektronischer Postsysteme
- ▷ Anreicherung von elektronischer Post mit anderen Dokumenten

**Information in vernetzten Umgebungen**

- ▷ Hypertext als Vernetzungsprinzip
- ▷ Erzeugung und Verwaltung von Dokumentenstrukturen
- ▷ Systematische Suche nach Informationen in vernetzten Umgebungen

**Automatische Verarbeitung von Information**

- ▷ Umgangssprachliche Formulierung von Verarbeitungsvorschriften
- ▷ Bausteine von Algorithmen
- ▷ Prinzipielle Funktionsweise von Rechenanlagen

aus [Frey u. a. 2001, S. 20], vgl. [Bayerisches Kultusministerium 2000]

Tabelle 8.1: Informatik – Bayern – Lerninhalte der 6. Jahrgangsstufe laut Lehrplan

Die Forschungsfrage für die Evaluation lautet:

- Stellt dieses Konzept der objektorientierten Modellierung als Grundlage für die Organisation eines „roten Fadens“ im konkreten Vermittlungszusammenhang für die Schülerinnen eine brauchbare Metapher bereit, die ihnen hilft „Neues“ mit Bekanntem so zu verbinden, dass eine sachgerechte, von konkreten Produkten unabhängige Sicht auf Informatiksysteme im Unterricht erkennbar wird?

Ergebnis der Forschungsfrage ist die folgende Untersuchungshypothese:

- Die provozierte Verbindung zwischen dem bereits Gelernten und dem Neuen, die durch die „künstliche Brücke“ der objektorientierten Modellierung zusammengebracht werden soll, ist für die Lehrenden nützlich, bleibt aber im Lehr-/Lernprozess für die Schülerinnen unsichtbar.

## Untersuchungsgestaltung

Zur konkreten Beobachtung des Unterrichts wurden – unter Berücksichtigung der o. g. Forschungshypothese – erste Konturen einer fachdidaktischen Evaluation entwickelt. Dazu wurden folgende Fragestellungen entwickelt:

- Welche Unterrichtsmittel (Medien und Informatikraum) kommen zum Einsatz?
- Welche Fachkonzepte werden im Unterricht thematisiert?
- Welche fachdidaktischen Konzepte werden im Unterricht umgesetzt?
- Welche Tätigkeiten führen die Schülerinnen aus? (Schüleraktivität)

In der 8. Kalenderwoche 2001 wurde an fünf Tagen jeweils in einem anderen Ort der Informatikunterricht von Lehrerinnen hospitiert. Dabei konnten Erfahrungen aus dem Projekt „Multimediale Evaluation in der Informatiklehrausbildung“ (MUE)<sup>299</sup> genutzt werden. Dieses Projekt wurde von 1999 bis 2001 vom Ministerium für Schule, Wissenschaft und Forschung (MSWF) des Landes Nordrhein-Westfalen gefördert, um die Ausbildung der beruflichen Kompetenzen für das Lehramt Informatik in erster und zweiter Phase zu verbessern. Das Projekt wurde von den Partnern an den Universitäten Dortmund und Paderborn und an den Studienseminaren Hamm und Paderborn gemeinsam realisiert. Der Unterricht wurde in zwei Sichten aufgezeichnet. Die Videodokumentation aus dem bayerischen Schulversuch diente außerdem zur weiteren Qualifizierung von Multiplikatoren innerhalb Bayerns. Die Beschreibung des unterrichtlichen Konzepts, der Unterrichtsvorbereitung, ausgewählter Beobachtungen und evaluativer Elemente ist in [Frey u. a. 2001] dargestellt und wurde anlässlich der INFOS 2001 in Paderborn in einem Workshop fachdidaktisch Interessierten vorgestellt (vgl. [Hubwieser u. a. 2001b]).

Im Zusammenhang mit der vorliegenden Arbeit wird der Fokus auf die fachdidaktische Fragen und Erfahrungen gelegt, die in der Vorbereitung und der Auswertung der Hospitationserfahrung qualitative Ergebnisse begründen.

Thema	Hospitationsprotokoll – Anhang
Ordnerbäume – objektorientierte Beschreibung	E.1, S. 187
Textverarbeitung – objektorientiert betrachtet	E.2, S. 189
E-Mail – objektorientierte Beschreibung	E.3, S. 191
Verweisstruktur – objektorientiert beschreiben und modellieren	E.4, S. 193
E-Mail – Aufbau der Adresse, Transport im Netz	E.5, S. 195

Tabelle 8.2: Hospitationen im Bayerischen Schulversuch – Themenübersicht

Die Zielsetzung des beobachteten Informatikunterrichts wird sehr deutlich von einer konzeptionellen Sicht auf die Wirkprinzipien von Informatiksystemen getragen (vgl. [Frey u. a. 2001, S. 20f]). Die Schülerinnen lernen im informatischen Kontext, so dass die Umsetzung grundlegender Informatikmethoden und -inhalte exemplarisch thematisiert werden.

Ausformulierte Dokumentationen der Beobachtungsergebnisse der Hospitationen wurden nach Reflexion durch die Evaluatoren und der unterrichtenden Informatiklehrerinnen veröffentlicht (vgl. [Frey u. a. 2001]). Zum Thema „Verweisstruktur – objektorientiert beschreiben und modellieren“ (Hypertexte) wird im Folgenden (in Anlehnung an das Hospitationsprotokoll) eine Auswertung dargestellt.

<sup>299</sup> siehe [Humbert u. a. 2000], [Magenheim u. a. 2000]

## Verweisstruktur – objektorientiert beschreiben und modellieren (Hypertexte)

### 1. Medieneinsatz – Informatiksystem

Zur Übertragung des Lehrerbildschirms auf die Schülerbildschirme wurde eine hardwaremäßige Videovernetzung verwendet. Mit dem schulischen Intranet (Fileserver/Printserver – Novell Version 4.x) steht jedem Arbeitsplatz ein dezidiertes Verzeichnis zur Verfügung. Jeder Rechner hat außerdem über einen Kommunikationsserver (Linux) Zugang zum Internet. Zudem verfügt die Schule über zwei fahrbare Beamer-/Rechnereinheiten, die primär in den Klassenzimmern/Klassenräumen eingesetzt werden.

Als Standardanwendungssoftware wurde StarOffice™ 5.2 zur Dokumentenerstellung, InternetExplorer™ zur Ansicht verwendet.

Das Whiteboard wurde für die Haftobjekte (Arbeitsblatt), das Kinderlexikon [Würmli 1992] für die einleitende Motivation benutzt.

Präsentation mit Hilfe des Overheadprojektors

- Auszüge aus dem Kinderlexikon [Würmli 1992]
- Benutzerhefteintrag
- Arbeitsblatt zur Vertiefung
- Arbeitsblätter (Objektdiagramm Lexikon)
- Klasse Verweis

Die Arbeit an dem neuen Stoff wurde durch Haftbilder an der Tafel handelnd unterstützt. So wurde die Verknüpfungsstruktur der Dokumente mit Verweisen anschaulich demonstriert. Die Demonstration durch die Lehrerin zur Gestaltung eines Verweises erfolgte durch die Nutzung der Präsentationstechnik im Informatikraum (Beamerpräsentation der Erstellung eines neuen Dokuments mit eingebundenen Verweisen).

Die Arbeit der Schülerinnen wurde durch zwei Anwendungen unterstützt. Für Erstellung und Nutzung wurden verschiedene Programme eingesetzt, um so die beiden Sichten voneinander zu trennen.

### 2. Informatisches Fachkonzept

Der Übergang von den „traditionellen“ Verweisstrukturen<sup>300</sup> eines Lexikons zur informatischen Verweisstruktur in Hypertexten konnte nur gelingen, weil die Schülerinnen den Bezug zur Textverarbeitung und Dateiorganisation aus objektorientierter Sicht bereits soweit verinnerlicht hatten, dass ihnen die Übertragung (Transfer) der dort entfalteten Fachsicht auf diesen neuen Gegenstand keine prinzipiellen Schwierigkeiten bereitete.

Die notwendigen Vorkenntnisse (Klassen- und Objektbegriff, Baumstruktur, insbesondere Datei-Ordner-Baumstruktur; relativer und absoluter Pfadbegriff) und Fertigkeiten (aktive Änderungen von Attributwerten an Objekten) wurden vor Beginn dieser Unterrichtsreihe erarbeitet.

Damit konnten Strukturähnlichkeiten auf die neue Situation übertragen werden: Das Textdokument wird erweitert zum Hypertextdokument mit neuen Dateiattributen.

Die Nachricht an ein Objekt, verstanden als Methodenaufruf zur Änderung von Dateiattributen, wird zur Kaskade von Nachrichten verallgemeinert. So löst das Setzen der Zieladresse Attributwertänderungen des zugehörigen Zeichens oder die Verweisaktivierung das Schließen bzw. Öffnen der entsprechenden Dateien aus.

Bemerkenswert ist die von den Schülerinnen verwendete Fachsprache: Begriffe, wie Klasse, Objekt, Attribut, Methode, Nachricht werden von den Schülerinnen korrekt verwendet. Das Arbeiten mit zwei verschiedenen Sichten bietet die Möglichkeit, sowohl eine konstruktive Sicht, wie auch eine Nutzersicht auf die Verweisstruktur zu erhalten.

<sup>300</sup> wie sie durch Verweise in Lexikonbeiträgen [siehe auch →[Elektrizität](#)] realisiert sind

### 3. Fachdidaktische Konzepte

Als Eingangsbeispiel wurde ein traditionelles Lexikon gewählt, um den Fachinhalt (Verweisstrukturen) zu motivieren. Daran schloss ein Unterrichtsgespräch mit Wiederholung der vorherigen Fachinhalte Informatiksysteme (Ordner, Textverarbeitung, ...) und objektorientiertes Modellieren an.

Der Unterrichtsprozess wurde so gestaltet, dass unerwarteten Aktivitäten der Schülerinnen bewußt Raum gegeben wird. Alle Schülerinnen hatten in der anschließenden Phase ausreichend Zeit, sich mit den gestellten vielschichtigen Arbeitsaufträgen angemessen in praktischer Tätigkeit auseinanderzusetzen.

In diesem Prozess wurden sie von der Lehrerin beraten, unterstützt – nach und nach auch durch die Mitschülerinnen, die die Aufgaben bereits erfolgreich bearbeitet hatten. Die zielgerichtete Arbeit der Schülerinnen verläuft unter Beachtung der zur Verfügung stehenden Zeit, der Komplexität der Arbeitsaufträge und nicht zuletzt der Abstraktionsanforderungen erfolgreich (Kooperation, Kollaboration, Teamarbeit).

Es folgte ein fragend-entwickelndes Unterrichtsgespräch zur Vertiefung. Dabei zeigte sich der Erfolg des durchgehend objektorientierten Ansatzes: Die Schülerinnen waren in der Lage, Objektbeziehungen auf einer übergreifenden Abstraktionsebene zu erkennen. Dies wurde durch die Lehrerin einerseits durch geeignete „Prozessverunschärfung“ (unscharfes Bild nach Abnehmen der Brille), andererseits durch das Verdecken der enthaltenen Objekte (Umdrehen der Haftbilder) am konkreten Beispiel veranschaulicht.

Die Thematisierung von Nachrichtenaustausch und -weiterleitung ermöglichte einerseits die Rückbindung an die zu Beginn der Einführung (Demonstration) erkannte automatische Attributänderung, andererseits konnte den Schülerinnen das durch die Aktion einer Benutzerin ausgelöste Öffnen und Schließen der jeweiligen Datei durch ein verbales Erklärungsmuster deutlich gemacht werden.

### 4. Gestaltung des Lehr-/Lernprozesses

Der Vergleich traditioneller Verweise mit informatischen Verweisstrukturen wurde am konkreten Beispiel (Lexikon), der Aufbau einer angemessenen Verweisstruktur und der Abbildung in objektorientierte Beschreibungen im Klassengespräch mit sorgfältigen Impulsen untersetzt. Durch einen daran anschließenden Arbeitsauftrag (Stillarbeit) mit dem Vergleich der Ergebnisse wurde eine „theoretische“ Zwischensicherung auf Papier realisiert (die Schülerinnen verschriftlichten die Ergebnisse für ihre Unterlagen).

Die Umsetzung der gewonnenen Ergebnisse auf die konkrete Handlungsebene ist wichtiger Bestandteil des erfolgreichen Informatikunterrichts in dieser Lernalterstufe. Diese Umsetzung muss im unmittelbaren Zusammenhang mit der Erarbeitung der abstrakten Begrifflichkeiten erfolgen. Sie kann nicht auf einen späteren Zeitpunkt verschoben werden.

Somit wurde den Schülerinnen Raum für die Umsetzung der konzeptionellen Elemente am konkreten Beispiel mit der bekannten Infrastruktur (den in den Unterrichtsreihen zuvor auch benutzten Informatiksystemen) gegeben. Hier mussten die Schülerinnen ihr Wissen reorganisieren, um damit zu dem neuen Punkt vorzudringen. Damit wurde ein integrativer Übungsteil realisiert.

Die beiden Sichten (StarOffice™ zur Erstellung [Produktionsmodus] – InternetExplorer™ zur Ansicht [Nutzungsmodus]) wurden dadurch auseinander gehalten, dass (durch die Vorgabe der Lehrerin) der Produktionsmodus vom Nutzungsmodus vollständig getrennt wurde. Dies wurde durch die Nutzung verschiedener Programme realisiert – einerseits StarOffice™, auf der anderen Seite InternetExplorer™. Damit wurde es den Schülerinnen ermöglicht, die beiden Sichten unterscheiden zu können.

Die Lehrerin ließ den Schülerinnen Raum für eigene Lernwege, die nicht in jedem Fall sofort erfolgreich waren.<sup>301</sup> Die daran anschließende Vertiefung zum Modellierungsprozess im fragend-entwickelnden Unterrichtsgespräch führte zu einer nochmaligen Vergewisserung über das Erreichte

<sup>301</sup> Die in der Demonstration durch die Lehrerin präsentierte Neugestaltung eines Dokuments mit Verweisen brachte einige Schülerinnen im nachfolgenden Arbeitsprozess trotz anderslautender Arbeitsanweisung auf die Idee, ebenfalls ein neues Dokument zu erstellen.

und thematisierte die erzielten Ergebnisse, aber auch die Hindernisse, die einzelne Schülerinnen (genauer: Paare von Schülerinnen) konkret zu überwinden hatten, um die gestellten Aufgaben lösen zu können.

## **Ergebnis der Reflexion der Unterrichtsbeobachtung**

### **Medieneinsatz**

Um praktisches Handeln mit den schulischen Informatiksystemen zu ermöglichen, sind einige (auf den ersten Blick nicht „sichtbare“) Voraussetzungen zu erfüllen.

- Ein funktionierendes schulisches Intranet (Ablage und Austausch der erstellten Dateien)
- Beamer (zur Demonstration) mit entsprechender Einbindung in die Infrastruktur
- Verfügbarkeit der für die beiden „Sichten“ notwendigen getrennten Anwendungen für die Schülerarbeitsplätze

An der Tafel wurde mit Haftelementen die Struktur veranschaulicht.

Lerngruppenbezug: Für die konkrete Arbeit der Schülerinnen mit den schulischen Informatiksystemen muss in der Partnerarbeit u. a. geklärt werden, wie die Arbeit aufgeteilt wird, welche der beiden Schülerinnen sich einloggt (unter welchem Account gearbeitet wird), wie die (elektronischen) Ergebnisse der Partnerin zur Verfügung gestellt werden.

### **Informatisches Fachkonzept**

Die offensichtliche Wirksamkeit der gewählten Beispiele zur Vermittlung des informatischen Fachkonzepts war für die Beobachter überraschend, da die Schülerinnen in der Argumentation und der Umsetzung deutlich machten, dass die Fachsprache nicht „aufgesetzt“ wurde, sondern mit Hilfe der Fachsprache über die Gegenstände kommuniziert werden konnte.

Die Einnahme der Produzenten- aber auch der Konsumentensicht in der konkreten Situation verdeutlicht die Mächtigkeit der Abstraktionsfähigkeit, die die Schülerinnen durch die frühzeitige Auseinandersetzung mit diesem mächtigen Fachkonzept erreichen können.

### **Fachdidaktische Konzepte**

Die Verbindung herkömmlicher Verweisstrukturen und die Anknüpfung an die objektorientierte Fachsprache als konzeptionelle Grundlage des Unterrichts erwies sich als Anker, der es den Schülerinnen ermöglichte, die neuen Elemente in die Struktur einzubeziehen. Das „Neue“ wird damit auf das „Bekannte“ zurückgeführt. Die Erweiterung wird auf ein neues Attribut und zwei neue Methoden beschränkt.

### **Schülerinnenaktivität und -verhalten**

Die anspruchsvollen Inhalte bedürfen der adäquaten schülerorientierten Umsetzung, um ihre Wirksamkeit entfalten zu können.

Die Lehrerin ermöglichte den Schülerinnen die Erkundung eigener Lernwege und liess fehlerhafte Wege in der Partnerarbeit zu.

Beobachtungen, die in den anderen Hospitationen gesammelt wurden, sind stichwortartig in Anhang E (ab S. 187) dokumentiert. Zusammenfassend ist festzustellen, dass die oben angegebene Untersuchungshypothese (vgl. Abschnitt 8.3, S. 129) auf Grund der Evaluation nicht aufrecht erhalten werden kann, zeigte sich doch, dass die Schülerinnen im Unterrichtsprozess objektorientierte Begriffe nutzen, z. B. die Erweiterung der Funktionalitäten mit Hilfe weiterer Attribute und darauf anzuwendender Methoden beschreiben; darüber hinaus in der Umsetzung die Modellierung praktisch umzusetzen in der Lage sind. Damit stellt die objektorientierte Sicht einen für diese Schülerinnen der 6. Jahrgangsstufe grundlegenden Begriffsapparat bereit, der als durchgängige Fachsprache zur unterrichtlichen Umsetzung in dieser Alterstufe als geeignet angesehen werden kann.

## 8.4 Überlegungen zur Anwendung des Modulkonzepts in der Sekundarstufe I

Die Ergebnisse der Evaluation des bayerischen Schulversuchs werden im Folgenden für eine kritische Analyse des in Kapitel 6 entwickelten Modulkonzepts herangezogen. Dazu wird in Tabelle 8.3 der Zusammenhang (im Sinne der Übereinstimmung) zwischen den für alle Schülerinnen verpflichtenden Lerninhalten des Schulfachs Informatik der 6. Jahrgangsstufe in bayerischen Gymnasien<sup>302</sup> und dem Modulkonzept<sup>303</sup> vorgestellt. Daran schließt eine Diskussion der Überlegungen an, welche Elemente des Modulkonzepts in der Sekundarstufe I verankert werden können und welche Elemente möglicherweise nicht altersangemessen vermittelt werden können.

Modul	Informatiksysteme verantwortlich nutzen	Elemente der theoretischen Informatik	informatische Modellierung (OOM)
Lerninhalte			
Darstellung von Information mit Hilfe von Dokumenten	(—)	(—)	(×)
Verwaltung von Dokumenten	×	(—)	(×)
Versand von Dokumenten	×	(—)	(—)
Information in vernetzten Umgebungen	×	(—)	(×)
Automatische Verarbeitung von Information	(—)	(×)	×

Tabelle 8.3: Lerninhalte Sekundarstufe I × Module<sup>304</sup>

Im Folgenden wird in der Reihenfolge der Module eine Bewertung und Einordnung der inhaltlichen Schwerpunkte des Lehrplans für die 6. Jahrgangsstufe vorgenommen.

### Informatiksysteme verantwortlich nutzen

In der Konkretion der Lerninhalte des Lehrplans finden sich Elemente des Moduls „Informatiksysteme verantwortlich nutzen“ an exponierter Position. Vor allem die Lerninhalte „Information in vernetzten Umgebungen“ und „Versand von Dokumenten“ zeichnet eine hohe Übereinstimmung mit den im Modulkonzept explizit als prioritär ausgewiesenen Modul „Informatiksysteme verstehen und verantwortlich nutzen“ aus.<sup>305</sup> Bei genauerem Studium der in Tabelle 8.1 differenzierter angegebenen Lerninhalte ist festzustellen, dass einige Elemente des Moduls nicht aufgeführt werden. Accounting und Netiquette werden im

<sup>302</sup> vgl. Tabelle 8.1

<sup>303</sup> vgl. Abschnitt 6.2.1, S. 106ff

<sup>304</sup> Legende: Lerninhalt lt. Plan für die Sekundarstufe I in bayerischen Gymnasien Übereinstimmung mit der Zielsetzung eines Moduls

× hohe Übereinstimmung

(×) Übereinstimmung ohne „Implementierung“ im Sinne des Modulkonzepts

(—) nicht berücksichtigt

<sup>305</sup> Es ist darauf hinzuweisen, dass hier der in Kapitel 4, Abschnitt 4.2.1, S. 67 dargestellte Informationsbegriff zu Grunde gelegt wird. Folgerichtig könnte als Überschrift „Kommunikation in vernetzten Umgebungen“ statt „Versand von Dokumenten“ Verwendung finden.

Lehrplan nicht ausdrücklich genannt. Im Modulkonzept stellen diese Elemente eine wichtige Grundlage bereit. Diesen konkreten Punkten wird in dem bayerischen Konzept im Vergleich zum Modulkonzept weniger Aufmerksamkeit gewidmet.<sup>306</sup>

Darüber hinaus wurde bei der Evaluation vor Ort festgestellt, dass ein nicht unerheblicher Aufwand betrieben wird, um die handlungsorientierte Umsetzung der diesem Modul zuzuordnenden Lerninhalte für die Schülerinnen zu ermöglichen. Dazu schafft das Handlungsrückgrat die notwendigen Voraussetzungen (vgl. Abschnitt 6.1, S. 104).

### **Elemente der theoretischen Informatik**

Bei der Betrachtung des Moduls „Elemente der theoretischen Informatik“ ist zu berücksichtigen, dass eine altersgemäße integrative Vermittlung bei den gewählten Lerninhalten kaum möglich scheint. Mit dem Lerninhalt „Automatische Verarbeitung von Information“ werden algorithmische Grundmuster unterrichtlich verankert.

Ideen für eine altersgemäße Thematisierung von Elementen der Informatik, die einen theoretisch anspruchsvollen Hintergrund aufweisen, wurden in [Schwill 2001] dokumentiert. Es fehlen unterrichtliche Umsetzungen für die dort vorgetragenen Beispiele.

Bei den im bayerischen Schulversuch thematisierten Inhalten ist eine integrative Vermittlung dieser Aspekte aus Zeitgründen kaum umzusetzen. Damit wird deutlich, dass im Rahmen dieses Vorschlags Elemente des Moduls „Elemente der theoretischen Informatik“ kaum Berücksichtigung finden können. Dennoch sind Anknüpfungspunkte vorhanden,<sup>307</sup> die beispielsweise im Rahmen der Begabungsförderung in den Jahrgangsstufen 7 und 8 vertieft werden könnten.

### **Informatische Modellierung**

In der in Abschnitt 8.3 diskutierten Evaluation des bayerischen Schulversuchs, der die Grundlagen für die Einführung des Pflichtfachs Informatik in Gymnasien ab Jahrgangsstufe 6 darstellt, wurde dokumentiert, dass Lehrerinnen und Schülerinnen Fachbegriffe aus der objektorientierten Modellierung verwenden. Damit wird eine durchgängige Fachsprache verwendet, die für diesen Anwendungsfall als tragfähig angesehen wird. Der Bezug zu dem Modulkonzept – konkret zu dem Modul „informatische Modellierung“ in der Ausprägung objektorientierte Modellierung – ist offensichtlich. Die für den 6. Jahrgang geplante und im Schulversuch erprobte objektorientierte Modellierung benutzt die objektorientierte Fachsprache. Er sieht keine Implementierung auf der Ebene der Programmierung (abgesehen von dem Bereich „Automatische Verarbeitung von Information“) vor.

### **Zusammenfassung**

Im Rahmen der Sekundarstufe I wurden in dem bayerischen Schulversuch Elemente der Module „Informatiksysteme verstehen und verantwortlich nutzen“ und „Modellierung – zentrales Feld informatischer Arbeit“<sup>308</sup> erfolgreich umgesetzt. Die beobachteten Umsetzungen zeigen, dass es möglich ist, bereits in der Sekundarstufe I erste fachlich fundierte Inhalte eines allgemein bildenden Informatikunterrichts lerngruppenbezogen und altersgemäß im Unterricht zu verankern.

Das Modulkonzept wurde in Kapitel 6 entwickelt, um für Schülerinnen in der gymnasialen Sekundarstufe II die Verbindung zwischen den informatischen Fachkonzepten und der angemessenen Strukturierung

<sup>306</sup> Den ausführlichen Materialien zur Umsetzung (erreichbar über die Webseite: <http://ddi.in.tum.de/unterricht/egy3.html>) kann entnommen werden, dass diese Elemente z. Tl. in den Unterricht integriert werden (vgl. exemplarisch <http://ddi.in.tum.de/unterricht/hefteintraege/01-lq.jpg>).

<sup>307</sup> Hier sei exemplarisch auf Routing verwiesen – dies bietet im Zusammenhang mit „Versand von Dokumenten“ einen interessanten Anknüpfungspunkt – ein Vorschlag für die unterrichtliche Umsetzung mit Hilfe eines Rollenspiels findet sich in [Perrochon 1996, S. 123].

<sup>308</sup> realisiert mit Hilfe der objektorientierten Modellierung (OOM)

der Lernprozesse Rechnung zu tragen. Durch die Schülerinnenbefragung (vgl. Kapitel 7) konnte eine besondere Fassade – der Einfluss des Informatikunterrichts auf das Bild der Informatik bei Schülerinnen – näher betrachtet werden und ergab in der Auswertung methodische und inhaltliche Hinweise für den Informatikunterricht. In der externen Evaluation des bayerischen Schulversuchs wird deutlich, dass sich das Modulkonzept eignet, um Informatikunterricht in der Sekundarstufe I qualitativ zu untersuchen. Die Eignung des Modulkonzepts zur Entwicklung curricularer Elemente im Sinne einer allgemeinen konstruktionsleitenden Grundlage für den Informatikunterricht ist hingegen ein Schritt, der aussteht. Da vermehrt curriculare Aktivitäten bzgl. des Schulfachs Informatik entfaltet werden<sup>309</sup>, kann das Modulkonzept für eine konkrete Altersstufe so verfeinert werden, dass es nutzbringend für die Unterstützung der Entwicklung von Informatikcurricula eingesetzt wird. Damit kann der Versuch, das Modulkonzept zu einer tragenden Säule / Klammer des Informatikunterrichts weiterzuentwickeln, nach dieser ersten (erfolgreichen) Prüfung in Betracht gezogen werden.

---

<sup>309</sup> aktuell trifft dies für den Hauptschullehrplan in Nordrhein-Westfalen (Informatik als abschlussrelevantes Wahlfach), aber auch für die Überarbeitung der Einheitlichen Prüfungsanforderungen in der Abiturprüfung (EPA) Informatik zu

In welche Umwelt wird das Stadtkind geboren? In ein Ensemble von komplexen Systemen, die für diejenigen, die sie konzipieren, eine bestimmte Bedeutung, für diejenigen, die sie benutzen, eine ganz andere Bedeutung haben. Im Kontakt mit Tausenden von Systemen – an ihren Endstationen stehend – kann der Stadtmensch Apparate bedienen. Nur, durch solche Bedienung lernt er nichts.

[Illich 1986, S. 108]

## Kapitel 9

# Informatik als Gegenstand und Mittel des Lernens

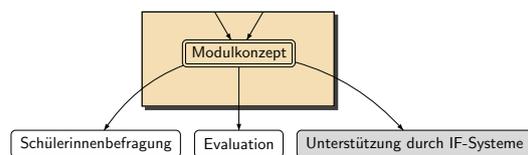


Abbildung 9.1: Struktur Kap. 6–9

Die Entwicklung von Informatiksystemen, die im Unterricht eingesetzt werden [sollen], stellt hohe Ansprüche sowohl an den fachwissenschaftlichen Hintergrund der Entwickler, aber auch an die [Fach-]Didaktik. Insbesondere, wenn die zu entwickelnden Informatiksysteme besonderen didaktischen Anforderungen entsprechen sollen, wird „die Luft sehr dünn“.

Viele der Informatiksysteme, die entwickelt wurden, um Lernprozesse zu unterstützen, sind nicht für den Anwendungsfall „Schule“, sondern für individuelles Lernen erstellt worden und eignen sich daher nur bedingt für den Einsatz im Unterricht. Zu Problemen, die sich im Zusammenhang mit der der Entwicklung und dem Einsatz von Informatiksystem ergeben, sei exemplarisch auf vorliegende Untersuchungen<sup>310</sup> verwiesen.

Die Überlegungen in diesem Kapitel konzentrieren sich auf Entwicklung und Darstellung der Gestaltungsanforderungen, die bei der Konstruktion von Informatiksystemen zum Einsatz im Informatikunterricht berücksichtigt werden [sollten]. Die formulierten Anforderungen konnten als konstruktionsleitende Ideen bei der Entwicklung einiger Informatiksysteme Eingang finden. Erste Einsatzberichte<sup>311</sup> zeigen, dass mit Hilfe der formulierten Anforderungen zielgerichtet Informatiksysteme gestaltet werden können, die den Einsatz im Informatikunterricht erlauben.

Die Fachwissenschaft Informatik ist für schulische Lehr- und Lernprozesse in zweifacher Hinsicht bedeutsam: einerseits Gegenstand der Erkenntnis, aber andererseits mit ihren Systemen Mittel der Erkenntnis im Informatikunterricht zu sein.

<sup>310</sup> exemplarisch aus

- pädagogischer Sicht: [Schulmeister 2002]
- didaktischer Sicht: [Baumgartner 1997] – vor allem die differenzierte und grundlegende Kritik an den Kriterienkatalogen ist überzeugend; das vorgeschlagene Vorgehensmodell weist darüber hinaus darauf hin, dass die Lehrerin die Verantwortung für die Gestaltung der Unterrichts nur dann wahrnehmen kann, wenn sie die Kompetenz besitzt, die unterschiedlichen Dimensionen bewerten zu können
- informatischer Sicht: Exkurs zu Lernobjekten, vgl. Kapitel 3, S. 33

<sup>311</sup> vgl. Abschnitt 9.3

Die Verzahnung des vorliegenden Kapitels mit den bisherigen Teilen dieser Arbeit wird in der nebenstehenden Übersicht verdeutlicht.

Aus allen Teilen der vorliegenden Arbeit finden Hinweise in das vorliegende Kapitel Eingang. Die Hintergründe liegen auf der Hand: die Entwicklung der Informatik läßt sich nicht ohne einen Bezug auf konkrete Informatiksysteme darstellen – die Lerntheorie wurde mit Blickrichtung auf die Didaktik der Informatik dargestellt.

Wichtige Hinweise lassen sich in der Darstellung der geschichtlichen Entwicklung der Anforderungen an die Schulinformatik sowohl von Fachwissenschaftlern, aber auch aus fachdidaktisch orientierten Beiträgen ableiten.

Die Lehrexpertise (vgl. Kapitel 5) umfasste die beiden Untersuchungsbereiche „Methodisch-didaktische Aspekte“ und „[Software-]Technische Unterstützung des Informatikunterrichts“. In der Auswertung dieser Untersuchung finden sich eine Reihe von Aussagen, die in dem vorliegenden Kapitel berücksichtigt werden.

Bei der Entwicklung des Modulkonzepts gehen Konzepte der Projektorientierung und der Handlungsorientierung gestaltungsleitend in die Konstruktion ein – damit ist deutlich, dass Informatiksystemen – nicht nur als theoretischer Gegenstand – eine wichtige Funktion für den Lernprozess zukommt. Da die Orientierung an den Interessen der Schülerinnen in diesem Konzept eine maßgebliche Rolle erhält, ist die Benutzung von „real existierenden“ Informatiksystemen Bestandteil eines nach dem Modulkonzept zu strukturierenden Unterrichts.

Die Befragung der Schülerinnen ergab (im 11. Jahrgang) eine Zustimmung von 80,8% zu der „Definition“ *Informatik ist die Lehre von der Bedienung von Computer*. Der Umgang mit Informatiksystemen im Unterricht beeinflusst diese Vorstellung. Das Ergebnis verweist auf die Wichtigkeit einer kritischen Auseinandersetzung mit dem Bild der Informatik der Schülerinnen als Bestandteil des Informatikunterrichts. Die Verantwortung der Lehrenden besteht u. a. darin, zu entscheiden, für welche unterrichtlichen Phasen der Einsatz der Informatiksysteme ausgeschlossen werden muss/soll. Der Einsatz von Informatiksystemen im Informatikunterricht zur Unterstützung der Lernprozesse ist auf diesem Hintergrund nicht einfach zu beantworten. Wird über die Notwendigkeit des Einsatzes für die Implementierung der von den Schülerinnen selbst entwickelten Teile von Informatiksystemen hinaus das System als „Medium“ im Lernprozess eingesetzt, verstärkt dies den Eindruck bei Schülerinnen, dass Informatikunterricht (und damit Informatik als Fach/Wissenschaft) immer etwas mit dem Benutzen von Informatiksystemen zu tun hat. Viele Schülerinnen (der gymnasialen Oberstufe) erwarten vom Informatikunterricht auch, dass sie anschließend ihr konkretes Informatiksystem zu Hause effektiver bedienen können.<sup>313</sup>

Die in der Evaluation des Informatikunterrichts in bayerischen Gymnasien (vgl. Kapitel 8) gesammelten Ergebnisse weisen darauf hin, dass den Informatiksystemen, die im Unterricht eingesetzt werden, die Funktion zentraler Hilfsmittel für den Unterricht zukommt. Allerdings wurde der Aufwand, der notwendig ist, um diese Mittel konkret zur Verfügung zu stellen, bei der Evaluation nicht weiter untersucht. Deutlich ist, dass in den Schulen administrative Aufwändungen notwendig sind, die nicht ohne weiteres von Lehrerinnen neben ihrer Unterrichtstätigkeit erwartet werden können.<sup>314</sup>

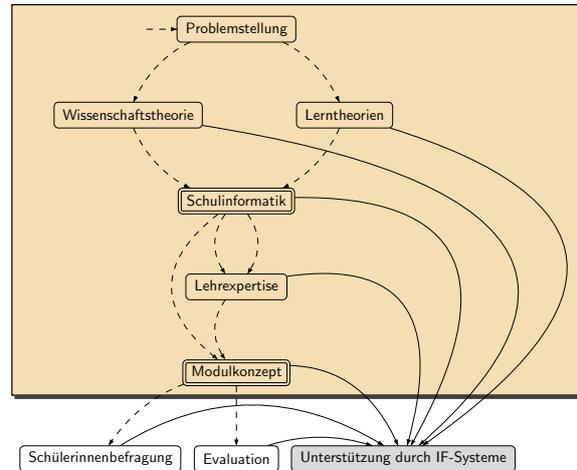


Abbildung 9.2: Einflussfaktoren Kapitel 9<sup>312</sup>

<sup>312</sup> In der Übersicht werden direkte Hinweise in den jeweiligen Kapiteln auf die vorliegenden Ausführungen mit durchgezogenen Linien dargestellt, die *gestrichelten Linien* geben die Gesamtlinie in der Arbeit an.

<sup>313</sup> vgl. exemplarisch Kapitel 7, S. 115

<sup>314</sup> vgl. [Breiter und Kubicek 1998] – zur Position der GI vgl. [Hubwieser u. a. 2001a]

## 9.1 Beitrag der Fachdidaktik zur Nutzung von Informatiksystemen für die Lehr-/Lernprozessgestaltung

Informatiksystemen kommt aus fachdidaktischen Gründen eine wichtige Funktion im Informatikunterricht zu. Schülerinnen benutzen Informatiksysteme, analysieren, konstruieren (Teile von) Informatiksysteme(n) und reflektieren über deren Grenzen. Die Gefahr einer Aufspaltung der Fachdidaktik in einen Teil, der sich den „real existierenden“ Systemen widmet und einen Zweig, der sich mit den Fachkonzepten beschäftigt, kann nicht ausgeschlossen werden.<sup>315</sup> Es erscheint nicht sinnvoll, Systeme, die heute möglicherweise dem Stand der Technik (typischerweise mit gewissen Einschränkungen) entsprechen, auch für die nächsten zehn (oder mehr) Jahre zu empfehlen. Zwischen der Gültigkeit wissenschaftlich fundierter Untersuchungen und Beschreibungen, die auf der Grundlage der aktuellen Restriktionen gewonnen werden, und den jeweils aktuellen technischen Möglichkeiten besteht damit ein Ungleichgewicht.

Die Untersuchungen zu den von den Expertinnen formulierten Anforderungen in Kapitel 5 machen deutlich, dass die Unsicherheit bzgl. möglicher Entwicklungen auch bei Expertinnen erkennbar ist. Die aktuellen Möglichkeiten der Softwaretechnik geben eine gewisse Hoffnung, dass durch softwaretechnische Trennung der Ebenen der Darstellung von den „dahinter liegenden Konzepten“ die Chance besteht, überdauernden Konzepten einen Platz zu geben, der dem jeweils aktuellen „Look & Feel“ konkreter Benutzungsoberflächen insofern Rechnung trägt, als eine klare Schnittstelle zwischen dem jeweiligen Fachkonzept und der graphischen Repräsentanz dafür sorgt, dass eine Portierung der Software ohne Änderung der Kernelemente möglich ist. Damit wird es möglich, eine Entkopplung zwischen der „Erscheinungs- und Darstellungsebene“ und den dahinter liegenden Fachkonzepten umzusetzen. Diese gilt es fachdidaktisch auf die je aktuelle Ausprägung der konkreten Strukturen anzuwenden und unterrichtlich verfügbar zu gestalten.

Diesen Anforderungen wird im Modulkonzept Rechnung getragen. Bei der objektorientierten Modellierung kann in der Analysephase die Trennung der Ebenen z. B. nach dem MVC-Konzept berücksichtigt werden. Die Orientierung an den invarianten, „hinter der Benutzungsoberfläche verborgenen“ Fachkonzepten erlaubt es, unabhängig von aktuellen Ausprägungen einen Fachhintergrund in den Mittelpunkt der Betrachtungen zu rücken. In der vorliegenden Arbeit wird mit dem Modulkonzept eine fachdidaktisch reflektierte Basis zur Verfügung gestellt, die nicht in erster Linie auf die Umsetzung in speziellen Informatiksystemen abzielt, sondern Fachkonzepte zum Ausgangspunkt der fachdidaktischen Gestaltung erklärt. Im Anhang F werden prototypische Umsetzungsvorschläge vorgestellt. Diese sind auf die Ebene des konkreten Unterrichts bezogen – in dem konkrete Informatiksysteme eingesetzt werden. So wird durch die Anwendung des Konzepts – bezogen auf konkrete Lerngruppen und konkrete Rahmenbedingungen – gezeigt, dass es möglich ist, die Ebenen miteinander zu verbinden, so dass das Konzept für die Umsetzung als Leitlinie tragfähig gestaltet werden kann.

## 9.2 Gestaltungsanforderungen

Um Entscheidungen zur Gestaltung von Informatiksystemen für den Einsatz im Informatikunterricht zu treffen, sind Anforderungen zu berücksichtigen. Dazu können zwei Anforderungsbereiche unterschieden werden:

1. Erkenntnisse der Informatik zur Konstruktion von Informatiksystemen.<sup>316</sup> Sie umfassen u. a. folgende Dimensionen
  - (a) Rechtlicher Rahmen für die Konstruktion von Informatiksystemen – [inter-]nationale Normen<sup>317</sup>

<sup>315</sup> Die Gefahr einer solchen Spaltung ist immer vorhanden, wie das Konzept des ECDL (vgl. Abschnitt 4.2.2, S. 76) zeigt, in dem Bedienwissen in den Vordergrund gerückt und zertifiziert wird – nicht etwa die Kenntnis der Fachkonzepte, oder der Grenzen der Informatik.

<sup>316</sup> Aufgabe der Informatik ist es, Informatiksysteme unter Berücksichtigung des Kontextes und ihrer Beziehung zur menschlichen geistigen Tätigkeit herzustellen und einzusetzen (vgl. Abschnitt 2.2, S. 13).

<sup>317</sup> vgl. zusammenfassend [Gorny 1998, S. 143f]

- (b) Technisch-organisatorische Randbedingungen
  - (c) Menschengerechte und aufgabenangemessene Gestaltung von Informatiksystemen<sup>318</sup>
2. Fachdidaktische und administrative Überlegungen, die in besonderer Weise die Rahmenbedingungen für den Einsatz im Informatikunterricht berücksichtigen. Dabei erweist sich das Kriterium der Plattformunabhängigkeit<sup>319</sup> als übergreifende Kategorie
- (a) Netzwerkfähigkeit – Einsatz im Schulintranet muss einfach möglich sein:  
Installation auf dem Schulserver – sofortige Nutzung auf allen Klienten, ohne z. B. Lizenzserver aufsetzen zu müssen und an den Klienten manuell eingreifen zu müssen.<sup>320</sup> Diese Eigenschaft wird als Handlungsrückgrat bezeichnet.
  - (b) Quelloffen und dokumentierte Schnittstellen [Application Program(ming) Interface (API)], damit Lehrerinnen die eingesetzte Software an ihre konkreten Bedingungen (Lerngruppe, etc.) anpassen können.<sup>321</sup>  
Programmiersprachenüberschreitend, d. h. Nutzung erstellter Software durch andere Programmiersprachen sollte über sorgfältig definierte Zwischenformen auch ohne Kenntnis des Quellcodes, der das Zwischenergebnis produziert hat, möglich sein.  
Zukunftsoffen: Anpassungsfähig für unterschiedliche Klienten – beispielsweise Personal Digital Assistants (PDA), Handhelds, Tablett-PCs – mit denen im [Informatik-]Unterricht zukünftig verstärkt gearbeitet wird.
  - (c) Portable Dokumentenformate, die auch ohne eine spezielle Produktionsumgebung „lesbar“ und änderbar sind (z. B. HTML, XML, T<sub>E</sub>X). Es ist anzustreben, dass Ergebnisse in offenen, möglichst standardisierten, nicht proprietären Formaten zur Verfügung gestellt werden – bei Grafiken z. B. Scalable Vector Graphics (SVG) – offenes Vektorgrafikformat des World Wide Web Consortium (W3C)<sup>322</sup>.

Es scheint kaum realistisch, dass im Dienst befindliche Lehrerinnen neben ihrem normalen Dienst Informatiksysteme gestalten, die den softwaretechnischen und fachdidaktischen Anforderungen gerecht werden [können]. In den zurückliegenden Jahrzehnten wurden dennoch immer wieder solche Systeme von Lehrerinnen entwickelt. Die Ursache für den Erfolg solcher Systeme liegt darin begründet, dass bis vor ca. zehn Jahren Informatiksysteme im schulischen Zusammenhang häufig als „stand alone“-Systeme betrieben wurden und damit die Anforderungen an Systementwickler erheblich geringer schienen.

Einheitliche Schulausstattungen (bezogen auf Hardware und Software), wie sie zeitweise für Stadtstaaten<sup>323</sup> durchgesetzt wurden, sind als überholt anzusehen. Heutzutage gibt es für Schulen völlig unterschiedliche Ausstattungsempfehlungen, die je nach Bundesland verschiedenen „Federn“ entstammen. Darüber hinaus nehmen die Träger der allgemein bildenden Schulen – in der Mehrzahl der Fälle die Kommunen – Einfluss auf konkrete Beschaffungsmaßnahmen. Versuche mit Laptopklassen [Issing und Schaumburg 2002], Überlegungen zur Ausstattung von Lerngruppen mit PDAs lassen erwarten, dass auch zukünftig keine einheitliche Ausstattung bzgl. der Endgeräte für die Schülerinnen zu erwarten ist. Vielmehr gilt es, Konzepte umzusetzen, die unabhängig von aktuellen Endgeräten tragfähig sind. Für die Softwareausstattung bieten sich plattformunabhängige, international vereinbarte Standards als Entscheidungsgrundlage zur Gestaltung an. Alle Versuche, einheitliche Software in die bundesdeutsche Schullandschaft zu tragen, sind bis

<sup>318</sup> Human Computer Interaction (HCI), Softwareergonomie – unter Berücksichtigung von Ergebnissen der Wahrnehmungspsychologie

<sup>319</sup> bezüglich der Infrastruktur, der Endgeräte, der Betriebssysteme und der Windowmanager, die den Schülerinnen zur Verfügung gestellt werden

<sup>320</sup> vgl. dazu [Rittershofer 2001]

<sup>321</sup> Als wünschenswert gelten darüber hinaus: kommandozeilenfähig, skriptfähig, sorgfältig modularisiert, so dass z. B. eine andere Grafikbibliothek verwendet werden kann, ohne den kompletten Quellcode ändern zu müssen.

<sup>322</sup> darüber hinaus zur Dokumentation Formate, die vollständig dokumentiert sind, wie PostScript © (PS) und Portable Document Format © (PDF)

<sup>323</sup> Unix<sup>TM</sup> in Berlin, Macintosh<sup>TM</sup> in Bremen

heute gescheitert.<sup>324</sup>

Hinweise zur Auswahl von Informatiksystemen im Zusammenhang mit Lehr-/Lernprozessen finden sich zur Zeit in einer überbordenden Anzahl. Es lassen sich nur wenige Vorschläge finden, die Fragen der Gestaltung und der Auswahl von Informatiksystemen auf das Unterrichtsfach Informatik beziehen.<sup>325</sup>

Für den Einsatz von Informatiksystemen im Informatikunterricht werden hier exemplarisch die Bereiche informatische Modellierung und Konstruktionsunterstützung für endliche Automaten einer kurzen Bewertung unterzogen.

- Objektorientierte Modellierung

Zur Erhöhung der Effizienz der Entwicklung von Software durch Expertinnen steht eine Vielzahl professioneller, integrierter Entwicklungsumgebungen [Integrated Development Environment (IDE)] zur Verfügung. Ihre Eignung zur Unterstützung des Informatikunterrichts wurde bisher nur selten untersucht, bevor solche integrierten Entwicklungsumgebungen im Informatikunterricht eingesetzt werden.<sup>326</sup>

- Hilfsmittel zur Konstruktion von endlichen Automaten

Es liegt eine Reihe von Werkzeugen für unterschiedliche Bedarfe und Zielgruppen vor.<sup>327</sup> Der Bereich kann fachlich erfolgreich eingegrenzt werden und eignet sich, um mit einfachen Werkzeugen eine unterstützende Funktion zu realisieren.

Im Zusammenhang mit erweiterter Suche in Texten können Werkzeuge, die reguläre Ausdrücke<sup>328</sup> benutzen, erheblich zur qualitativen Verbesserung der Suche beitragen. Allerdings ist es typischerweise nicht möglich, solche als sinnvoll erkannten Möglichkeiten in ein bestehendes Informatiksystem so zu integrieren, dass sie als zusätzliche Funktionalität nutzbar sind, da die Systeme entweder nicht für „normale“ Benutzerinnen erweiterbar oder änderbar sind, oder weil nach einer Änderung Seiteneffekte die grundlegenden Funktionalitäten des jeweiligen Systems verändern, so dass sie nicht mehr so funktionieren, wie vor der Änderung.

<sup>324</sup> Nur den Bundesländern, die mittels Zentralabitur über längere Zeit einheitliche Anforderungen formulieren konnten, ist es gelungen, softwaretechnische Anforderungen für die Informatische Bildung festzuschreiben. Das dies nicht unbedingt zukunftsweisend ist, kann am Beispiel des thüringischen Zentralabiturs gezeigt werden – obwohl der Autor die prinzipielle Entscheidung für das Betriebssystem und die Programmiersprache Oberon hier ausdrücklich ausnehmen möchte.

<sup>325</sup> Für die Entwicklung und Auswahl von Informatiksystemen für den Informatikunterricht können [Baumgartner 1997], [Gorny 1998], [Steinmetz 1999] und [Schulmeister 2002] mit Gewinn herangezogen werden. Die Orientierung an dem Erfahrungshintergrund, den Schülerinnen „mitbringen“ sollte nicht außer Acht gelassen werden, wie Mark GUZDIAL und Elliot SOLOWAY betonen: „Today’s technology produces the media that “kids these days” are consuming. These same kids can produce their kind of media using today’s technology. In fact, they want to. And they’ll learn programming to do it“ [Guzdial und Soloway 2002, S. 20].

<sup>326</sup> Beispiele für Entwicklungsumgebungen, bei denen dem Autor bekannt ist, dass sie im Informatikunterricht Verwendung finden: Delphi (nach dem Katalog [Rittershofer 2001] ist Delphi nicht für den Einsatz in vernetzten Umgebungen geeignet), Together – <http://www.togethersoft.com/>, ArgoUML [Robbins und Mayer 1998], ObjectPlant [Arctadius 1996] – siehe auch 5.2.4, S. 97ff). In [Ortmann und Koch 2001] werden OTW (Objekttechnologie Werkbank), Rational Rose, TogetherJ und ArgoUML vergleichend untersucht – allerdings war zu dem Zeitpunkt der Untersuchung bereits eine erheblich verbesserte Version von ArgoUML verfügbar, so dass ein Gutteil der Kritik schon zum Veröffentlichungszeitpunkt nicht mehr aufrecht erhalten werden kann. Die dargestellten Kriterien werden nicht weiter fachdidaktisch begründet, so dass der Eindruck der Beliebigkeit entsteht. Dem Kriterium der Plattformunabhängigkeit wird keine große Beachtung geschenkt (z. B. ist es mit ArgoUML möglich, Modelle aus Rose (wenn sie dort nach XMI exportiert wurden) zu importieren; darüber hinaus können entwickelte Modelle in das offene Format XML Metadata Interchange (XMI) und Grafiken nach SVG exportiert werden).

<sup>327</sup> – [Reichert u. a. 2000], [Hartmann und Nievergelt 2002] – Einsatzbereich: Sekundarstufen innerhalb einer Projektwoche, kurze Reihe zur Einführung in die Möglichkeiten der Modellierung mit Hilfe endlicher Automaten und Turingmaschinen (plattformunabhängig)

„Theorie [eignet sich] als Ausgangspunkt für Betrachtungen im Informatikunterricht [...]. Endliche Automaten und Turingmaschinen sind zwei Beispiele [...]. Die Lernumgebung Kara zeigt, dass Theorie nicht a priori abstrakt und schwer sein muss. Im Gegenteil: Kara hat geradezu einen spielerischen Charakter [...].“ [Hartmann und Nievergelt 2002, S. 474].

– [Häfel und Lamprecht 2001] – Einsatzbereich: Universität und Sekundarstufe II (plattformunabhängig)

– [Spohrer 2002] – Einsatzbereich: Mittelstufe (Jahrgangsstufen 8-10) des Gymnasiums (Idee interessant, Umsetzung kritikwürdig, da nicht plattformunabhängig – sollte nach m. E. unbedingt überarbeitet werden)

<sup>328</sup> die von endlichen Automaten erkannt werden

## Untersuchungsschwerpunkte

In der vorliegenden Arbeit kann dem Bereich der Informatiksysteme für den Einsatz im Informatikunterricht nur ein kleiner Teil gewidmet werden. Der Themenbereich bedarf über diese Arbeit hinaus grundlegender Studien. Dennoch können einige Punkte vorgestellt werden, die im Zusammenhang mit dem Modulkonzept bedeutsam sind<sup>329</sup>.

Folgende Fragestellungen und Aufgaben finden Eingang in diese Forschungsarbeit:

- Unterstützung der objektorientierten Modellierung im Informatikunterricht durch Informatiksysteme [Professionelle] Entwicklungsumgebungen setzen voraus, dass die Benutzerin bereits mit der objektorientierten Modellierung vertraut ist. Diese Voraussetzung muss im Informatikunterricht erst geschaffen werden. Einige der mit UML genormten graphischen Darstellungsformen bieten die Möglichkeit, den Modellbildungsprozess sowohl bezogen auf die statischen Strukturen (Klassen- und Objektdiagramme) als auch bezogen auf die dynamischen Strukturen (mit Hilfe von Anwendungsfallanalysen und Sequenzdiagrammen) adäquat im Zusammenhang von Lehr-/Lernprozessen zu visualisieren. Wie können die Sichten – eingeschränkt auf ein konkretes Szenario, so visualisiert werden, ohne dass den Schülerinnen die Möglichkeit genommen wird, Fehler zu begehen, deren Konsequenz sich erst im weiteren Verlauf der Modellierung erweisen? Mit der Arbeit einer Projektgruppe an der Universität Dortmund konnten diese Fragen untersucht und zu einem Ergebnis verdichtet werden, das zur Zeit im schulischen Einsatz erprobt wird.
- Eignung von Programmiersprachen für den Informatikunterricht  
Welche Programmiersprachen sind aus fachdidaktischen Gründen für den Einsatz im Informatikunterricht zu empfehlen? Der Einsatz von Programmiersprachen im Informatikunterricht (zur Implementierung der entwickelten informatischen Modelle) wird selten in Abrede gestellt. Überlegungen zur Eignung von Programmiersprachen für den Informatikunterricht haben fachdidaktisch eine lange Tradition. Bereits 1976 wurde von Klaus BRUNNSTEIN [Brunnstein u. a. 1976] eine Teilmenge von Pascal zur Vermittlung grundlegender Informatikkonzepte im Informatikunterricht (in der Sekundarstufe I) empfohlen.<sup>330</sup> Ungeachtet dieser Vorschläge – setzte sich (aus Kostengründen) – Turbo Pascal (in den jeweils aktuellen Versionen) für die Umsetzung im Informatikunterricht durch. Inzwischen findet die Programmiersprache Java weite Verbreitung, ohne dass der Einsatz einer grundlegenden Evaluation unterzogen worden wäre. Auf dem Hintergrund der frühen Entwicklung muss dies als Rückschritt bezeichnet werden, da offenbar die Fachdidaktik „ihre Hausaufgaben nicht gemacht hat“, da sie ohne grundlegende Kritik und damit m. E. didaktisch unreflektiert die Programmiersprache Java akzeptiert.<sup>331</sup> In einer Diplomarbeit wird die Fragestellung „Eignet sich die Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess?“ bearbeitet und um die Berücksichtigung fachdidaktischer Fragen erweitert.
- Unterstützung der Arbeit von Informatiklehrerinnen
  1. Die pädagogische Zielsetzung der Aufgabenstellung besteht darin, dass Schülerinnen möglichst schnell eine Rückmeldung über ihren aktuellen Lernstand erhalten sollen, um eigenverantwortlich ihre Lernfortschritte und Optimierungsmöglichkeiten zu erkennen. Die prinzipielle Eignung der im Informatikunterricht eingesetzten Infrastruktur (vernetzte Systeme, alle Schülerinnen verfügen über personenbezogene Accounts) zur Durchführung von Befragungen liegt auf der Hand. Aus diesem Grund soll mit Hilfe der Techniken der objektorientierten Modellierung ein netzbasiertes Werkzeug entwickelt werden, das ausschliesslich auf offenen Standards (inkl. der benutzen Programmiersprache) beruht. Dieses Werkzeug soll die Möglichkeit bieten, [Teil-]Ergebnisse der Auswertung von Schülerantworten zu visualisieren und notwendige

<sup>329</sup> Der Autor hatte die Möglichkeit, einige der offenen Fragestellungen im Zusammenhang mit der Betreuung einer Projektgruppe, einer Diplomarbeit und einer Abschlussarbeit zur Erlangung des Abschlusses als Fachinformatiker näher zu untersuchen. Alle drei Arbeiten wurden im Fachgebiet Didaktik der Informatik am Fachbereich Informatik (FBI) der Universität Dortmund angefertigt.

<sup>330</sup> Unterstützt wurde diese Arbeit durch das Bundesbildungsministerium und organisierte sich in einer Gruppe mit dem Namen Rechner-Gestützter Unterricht (RGU) (vgl. [Brunnstein u. a. 1974])

<sup>331</sup> abgesehen von der Veröffentlichung der Übersetzung von [Böszörményi 1998] durch Andreas Schwill (vgl. [Böszörményi 2001])

Berechnungen halbautomatisch durchführen zu lassen. Die Schülerinnen beantworten interaktiv (mit Hilfe automatisch generierter Formulare) Fragen – die [Zwischen-]Ergebnisse werden serverbasiert gespeichert und „nach Abgabe“ wird der schreibende Zugriff ausgeschaltet. Das Ziel besteht in der Vereinfachung der Erstellung, der Bearbeitung und der Auswertung von Befragungen zum Lernstand und damit zur unterrichtsbegleitenden Evaluation.

2. Da Informatiklehrerinnen typischerweise grundlegende Entscheidungen über die an einer Schule eingesetzten Informatiksysteme begleiten, fällt ihnen häufig die Ausgabe zu, ein (zunehmend) komplexes, heterogenes Netzwerk zu verwalten. Um diese Tätigkeit zu unterstützen, soll ihnen mit einem sorgfältig nach modernen Softwareentwicklungsmethoden erstelltes Informatiksystem zur Seite gestellt werden, das es ermöglicht, sowohl die statische Pflege der Daten über eine Webschnittstelle zu realisieren, aber auch die Erreichbarkeit der Klienten serverbasiert darzustellen. Die Realisierung einer serverbasierten Lösung, die ohne die Installation einer Datenbank erfolgen soll, ist plattformunabhängig zu gestalten, um den Einsatz in heterogenen Umgebungen zu erlauben. Diese Randbedingung bezieht sich auch auf die Server, auf denen das System zum Einsatz gebracht wird.

### 9.3 Beispielhafte Umsetzung ausgewählter Anforderungen

Ein konkretes Problem bei der Umsetzung der Gestaltungsanforderungen (vgl. Abschnitt 9.2) besteht darin, dass die entwickelten Kriterien nicht immer Eingang in die Überlegungen finden. Die Ursachen dafür sind vielfältig. In aktuellen Softwareprojekten<sup>332</sup> konnte beobachtet werden, dass von Informatikerinnen (insbesondere in der Ausbildung) der Wunsch formuliert und umgesetzt wird, möglichst aktuelle Versionen von Entwicklungswerkzeugen, Klassenbibliotheken, etc. Eingang in die Entwicklung finden zu lassen. Der zentrale Nachteil dieses Vorgehens besteht darin, dass aktuelle, teilweise nicht vollständig in ihren Seiteneffekten einschätzbare Varianten von Softwarepaketen zum Einsatz kommen, die später ihren Tribut fordern. Darüber hinaus muss festgestellt werden, dass damit dem Prinzip der Plattformunabhängigkeit nicht vollständig Rechnung getragen wird, da die benutzten Bibliotheken (in den jeweils aktuellen Versionen<sup>333</sup>) nicht zeitgleich für alle Plattformen zur Verfügung stehen. Das zuletzt genannte Problem wird mit großer Sicherheit bereits gelöst sein, wenn diese Arbeit erscheint, macht allerdings deutlich, wie eng der Entscheidungsspielraum für die Entwicklungen bezüglich solcher Detailentscheidungen wird, wenn aktuelle Entwicklungen als unverzichtbar für die eigene Arbeit angesehen werden (von Horst OBERQUELLE wurde dies im November 2002 anlässlich des Informatiktages in Bad Schussenried als „Featuritis“ bezeichnet).

#### Lernumgebung für objektorientiertes Modellieren im Informatikunterricht

Über zwei Semester (Wintersemester 2001/2002 und Sommersemester 2002) erstellte eine Projektgruppe (Nr. 403) an der Universität Dortmund (bestehend aus elf Informatikstudierenden des Hauptstudiums) eine Lernumgebung für objektorientiertes Modellieren im Informatikunterricht (LEO). Die Ergebnisse dieser Arbeiten umfassen die Lernumgebung mit einigen Szenarios und sind in dem (internen) Endbericht [Alex u. a. 2002] dokumentiert, der darüber hinaus Einblicke in die Arbeitsweise der Projektgruppe erlaubt. Im Juli 2002 stellte die Projektgruppe ihre Arbeitsergebnisse im Rahmen einer internationalen Konferenz (SEC III – Open IFIP-GI-Conference on Social, Ethical and Cognitive Issues of Informatics and ICT) ausgewählten Informatiklehrerinnen aus Nordrhein-Westfalen vor. Die Resonanz war sehr ermutigend. Inzwischen wurde den Betreuern der Projektgruppe mitgeteilt, dass ein Lehrbuch für den Informatikunterricht für die berufsbildende Sekundarstufe II geschrieben wird, in dem mit der Lernumgebung die grundlegenden Konzepte der Modellierung eingeführt werden.<sup>334</sup> Eines der von den Studierenden entwickelten Szenarios

<sup>332</sup> die vom Autor beratend begleitet wurden – seien sie nun von Entwicklerinnengruppen (wie im Folgenden beispielhaft an der Projektgruppe zur Erstellung einer Lernumgebung für objektorientiertes Modellieren im Informatikunterricht (LEO) dargestellt wird) aber auch von Einzelentwicklern (wie im Zusammenhang mit der Diplomarbeit zur Untersuchung der Eignung der Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess gezeigt wird)

<sup>333</sup> LEO benötigt Java in der Version 1.4 – PyNassi benötigt die Qt-Bibliothek <http://www.trolltech.no/products/qt/Version3>

<sup>334</sup> Folgendes teilte mir einer der Autoren in einer privaten E-Mail mit:

„Autoren: Ralf SAGORNY-SCHWARZ; Karsten MIELKE – Verlag: EINS“ [Sagorny-Schwarz, E-Mail vom 17. Oktober 2002].

ist an dem Konzept Stifte und Mäuse [Czischke u. a. 1999] orientiert. Damit kann mit Unterstützung durch die Lernumgebung für dieses verbreitete Konzept ein Fundament für die Modellierung geschaffen werden. Die entwickelte Lernumgebung stellt einen Baustein für die Überbrückung des fehlenden Bindeglieds zwischen der Modellierung und der Umsetzung in eine konkrete Programmiersprache dar. Gerade in diesem Feld zeichnet die entwickelte Umgebung aus, dass durch die fast vollständige Überdeckung der Begriff der Welt der Objektorientierung durch geeignete und schülerorientierte Szenarios der Zugang zur Modellierung angemessen unterstützt werden kann [Brinda u. a. 2002]. Darüber hinaus ermöglicht die Wahl eines offenen Formats für die Szenarios<sup>335</sup> interessierten Lehrerinnen und Schülerinnen, eigene Szenarios zu erstellen.

### **Eignet sich die Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess?**

Der Autor hatte Gelegenheit, dieses Thema für eine Diplomarbeit auszuschreiben und Ingo LINKWEILER bei der Bearbeitung zu betreuen. Als Ergebnis wurde die Untersuchung der Programmiersprache Python im softwaretechnischen und fachdidaktischen Kontext durchgeführt und in [Linkweiler 2002] dokumentiert. Im Rahmen dieser Diplomarbeit wurde mit Hilfe von Fallstudien die Tragfähigkeit der entwickelten Kriterien gezeigt:

PySuM realisiert das Konzepts Stifte und Mäuse in der Programmiersprache Python. Um zu verdeutlichen, dass auf Hochsprachenebene die Möglichkeit realisiert werden kann, performant interaktive Grafikausgaben zu realisieren, wurde das Konzept um Sprites erweitert.

PyNassi realisiert einen Struktogrammeditor<sup>336</sup>, der es erlaubt, die erstellten Struktogramme interaktiv „ablaufen zu lassen“ – aber auch, sie als lauffähige Python-Skripte abzuspeichern. Die Erweiterung des Editors um die Möglichkeit der Nutzung von Klassen<sup>337</sup>, erlaubt die Verbindung objektorientierter Modellierung mit Algorithmen und Datenstrukturen. Diese Fallstudie erfreut sich großer Beliebtheit, wie an der Anzahl der Dateizugriffe abgelesen werden kann.<sup>338</sup> Die Speicherung der Struktogramme wird mit dem Python-Werkzeug „pickle“ realisiert. Diese Entscheidung führt zu Einschränkungen in der Portabilität, da je nach Plattform unterschiedliche Zeilenendezeichen in den gespeicherten Daten abgelegt werden.

### **Werkzeug zu webgestützten Befragungen und ihrer Auswertung**

Im Rahmen der Ausbildung zum Fachinformatiker absolvierte Beytullah SEZGEN sein Betriebspraktikum an der Universität Dortmund<sup>339</sup> und erstellte im Zusammenhang mit der Einarbeitung in Problemstellungen aus dem Bereich der Didaktik ein Werkzeug zur Generierung und Auswertung von Befragungen. Das in der Programmiersprache Python entwickelte Werkzeug (PyLZK) arbeitet serverbasiert<sup>340</sup> und kann mit Hilfe üblicher Webbrowser von den Schülerinnen benutzt werden. Voraussetzung für die Nutzung sind personenbezogene Accounts, die zusammen mit einem „Einmalpasswort“ die Bearbeitung der Fragebögen erlauben. Alle Daten werden im XML-Format gespeichert, so dass an jeder Stelle des Prozesses mit einfachen Mitteln Änderungen möglich sind. Durch diese Eigenschaft kann das Werkzeug um weitere Funktionen (die auch in anderen Programmiersprachen entwickelt werden können) ergänzt werden. Prototypische Einsätze (in den Schultypen Gesamtschule und Berufskolleg) zeigen, dass die Funktionsweise des entwickelten Werkzeugs im Intranet einer Schule den Anforderungen vollauf gerecht wurde. Das Werkzeug PyLZK ist in [Sezgen 2001] dokumentiert.

<sup>335</sup> XML mit eigener Document Type Definition (DTD)

<sup>336</sup> Struktogramme gemäß [Norm DIN 66 261 ], [Nassi und Shneiderman 1973]

<sup>337</sup> Methoden können mit dem Struktogrammeditor ausgefüllt werden

<sup>338</sup> „[...] im Zeitraum 20.10.2002 bis 2.12.2002 ergibt die Auswertung des Log-Files durch ein kleines Python-Skript 613 Dateizugriffe. (ca. 10-20 kann man davon für Testzwecke noch abziehen) [...] Bisher hatte ich mit 5 Lehrern ausführlichen persönlichen Kontakt, [davon] 2 [...] telefonisch. Eine Berliner Lehrer-Gruppe hat eine große Python-Fraktion, und testet gerade PyNassi“ [Linkweiler, E-Mail vom 3. Dezember 2002].

<sup>339</sup> am Fachbereich Informatik, Fachgebiet Didaktik der Informatik

<sup>340</sup> für die Plattformen Linux, Mac OS X und Windows

## Klientendokumentation und Netzwerkmonitor

Zur Administration eines [schulischen] Intranets ist es häufig angezeigt, Details über einzelne Systeme verfügbar zu machen (dies gilt vor allem, wenn die Klienten nicht baugleich sind oder Software und Hardware klientenseitig installiert werden muss/soll). Um die Administration eines solchen Netzwerks zu erleichtern, wurde ein System geplant und erstellt, das diese Arbeit durch eine serverbasierte Webschnittstelle unterstützt. Es umfasst ein Modul, durch das auch die Aufstellung (von Klienten) an Hand einer webgestützten graphischen Darstellung (Raum/Etagenplan) unterstützt wird.

Über das Pflichtenheft hinaus wurde die Möglichkeit, eine Konfiguration zu kopieren<sup>341</sup> und die Änderung der Raumzuordnung für Klienten zusätzlich implementiert und dokumentiert und in das Werkzeug (PyNetzwerkmonitor) integriert. Das Ergebnis zeigt, dass durch einen gezielten Auftrag ein Informatiksystem entwickelt werden konnte, die eine wertvolle Hilfe für die Administratorinnen eines [schulischen] Netzwerks zur Verfügung stellt.<sup>342</sup>

In Tabelle 9.1 werden die im Abschnitt 9.2 dargestellten Kriterien auf die Beispiele bezogen.

Gestaltungsbeispiel	Gestaltungsanforderung (Stichwort) <sup>343</sup>					
	fachwissenschaftlich			fachdidaktisch		
	(a)	(b)	(c)	(a)	(b)	(c)
	Rechtlicher Rahmen für die Konstruktion	Technisch-organisatorische Randbedingungen	Menschengerechte und aufgabenangemessene Gestaltung	Netzwerkfähigkeit	Quelloffen, dokumentierte Schnittstellen	Portable Dokumentenformate
Lernumgebung für objektorientiertes Modellieren im Informatikunterricht	×	×	(×)	(×)	×	×
Eignet sich die Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess? Fallstudien	×	×	(×)	(×)	×	(×)
PyLZK, PyNetzwerkmonitor	(×)	×	(×)	×	×	×

Tabelle 9.1: Anforderungen und Beispiele – Kreuzreferenzen<sup>344</sup>

<sup>341</sup> um so nach einem Muster mehrere Klienten, die sich nur in Details unterscheiden, anzulegen

<sup>342</sup> vgl. Bericht zur betrieblichen Projektarbeit zur Erlangung des Abschlusses als Fachinformatiker – Fachrichtung Anwendungsentwicklung [Sezgen 2002]

<sup>343</sup> Die Gestaltungsanforderungen sind im Abschnitt 9.2 dokumentiert.

<sup>344</sup> Legende:

Berücksichtigung der Anforderungen in den Beispielen

× wurde berücksichtigt

(×) teilweise berücksichtigt

## 9.4 Überlegungen zur Gestaltung von Informatiksystemen als Lernmittel für den Informatikunterricht

Wie in den vorgelegten Untersuchungen deutlich geworden ist, kann durch explizit fachdidaktische Anforderungen eine Entwicklung angestoßen werden, die zu Informatiksystemen führt, die den für den Einsatz im Informatikunterricht ausgewiesenen Kriterien Rechnung tragen. Die Nutzung der Ergebnisse der vorgestellten Systeme steht am Anfang, so dass erste Rückmeldungen nur ein Schlaglicht auf den Einsatz zulassen. Die Rückmeldungen zeigen, dass es einen großen Bedarf gibt, dem offenbar bisher nicht zufriedenstellend Rechnung getragen wurde.

Die vorgestellten Überlegungen sollten Anlass sein, auf diesem Feld weiterzuarbeiten.

Sowohl Studierende, als auch Schülerinnen und Lehrerinnen sollen von der hier dokumentierten Arbeit profitieren, wie die folgenden Ausschnitte aus dem Abschlussbericht der Projektgruppe deutlich machen (das zu Grunde liegende Kapitel 16 des Endberichts enthält die persönlichen Erfahrungen und Eindrücke der Projektgruppenteilnehmerinnen).

Ich habe die [Projektgruppe] PG als sehr lehrreiche Veranstaltung erlebt. Ich möchte behaupten, dass die PG die wichtigste Lehrveranstaltung des Hauptstudiums hinsichtlich der Praxisbezogenheit ist. Im Verlauf der PG habe ich mich mit vielen Konzepten und Technologien (XML, CRC-Karten, verschiedene Vorgehensmodelle, ...) vertraut gemacht. [...]

So gab es didaktische Konzepte zu erarbeiten und dafür eine technische Umsetzung zu finden, Berichte und Vorträge anzufertigen und eine Unmenge von anderen Aufgaben über den Tellerrand des Informatikstudiums hinaus. [...]

Bei der Ideenfindung stand für mich deshalb meist nicht der Lerneffekt oder der pädagogische Nutzen im Vordergrund, sondern in erster Linie der nötige Aufwand, um eine bestimmte Funktion des Programms zu realisieren. Dabei ging es nicht nur um den primären Aufwand bei der Herstellung der Lernumgebung an sich, sondern auch darum, die Schnittstelle zwischen unserer Lernumgebung und den nachladbaren Modellwelten, den Szenarien, möglichst kompakt und verständlich zu halten, da früh abzusehen war, dass unsere PG nur die Zeit hatte, eine beispielhafte Auswahl von möglichen Modellwelten zu implementieren, um die Fähigkeiten des Gesamtsystems unter Beweis zu stellen und Beispiele für die Verwendung der Schnittstelle zur Verfügung zu stellen. [...]

[...] waren wir entschlossen, in der gewählten Programmiersprache Java das aktuelle [Software Development Kit] SDK 1.4 zu benutzen, welches zeitgleich mit der Implementierungsphase veröffentlicht wurde. Somit ist unser geistiges Werk auf aktuellem Stand [...]

Nicht zuletzt hat die Projektgruppe als Ergebnis der Entwicklungsarbeit ein vorzeigbares Produkt aus dem Zylinder hervorgeholt. Alles in allem hat sich die Teilnahme an dieser Projektgruppe in jeder Hinsicht gelohnt. [...]

Nach ca. 10 Monaten oftmals intensiver Arbeit liegt nun ein Produkt vor, was die Welt und den Informatikunterricht in der Schule hoffentlich bereichert. [...]

Mich reizten ein fächerübergreifendes Projekt, das die Informatik und ihre Didaktik verband, wie es an der Universität leider nur selten geschieht, und die Idee, ein Ergebnis nicht als reinen Selbstzweck oder bloßen Zwischenschritt eines übergeordneten Forschungsplans zu erstellen, sondern ein Produkt zu entwickeln, das mit der nötigen Arbeit und etwas Glück auch in der Praxis Anwendung finden sollte. [...]

Im Gegensatz zu manchen Vorerfahrungen stellte sich die Gruppenarbeit als sehr produktiv heraus. [...]

[Alex u. a. 2002, S. 120–126 – Ausschnitte aus verschiedenen persönlichen Beschreibungen]

Ziel der Überlegungen ist es, unter Zuhilfenahme ausgewählter Kriterien zu verdeutlichen, dass mit prototypischen Umsetzungen diese Kriterien erfüllt werden können. Mit den dokumentierten Umsetzungen wurde gezeigt, dass die aufgestellten Kriterien geeignet sind, um gestaltungsleitende Entscheidungen vorzubereiten und Informatiksysteme für den Einsatz im Informatikunterricht auf dieser Basis zu entwickeln.<sup>345</sup>

Der Ansatz erweist sich als tragfähig, sollte aber weiterentwickelt werden. Zentrale Kriterien fachlicher und fachdidaktischer Art ergänzen einander. Da (zumindestens zur Zeit) kein ökonomisches Interesse an der Entwicklung solcher Informatiksysteme besteht, die den hier genannten Kriterien gerecht wird, sollte bei Arbeiten der Didaktik der Informatik explizit auf die Entwicklung offener, plattformübergreifender Systeme gedrängt werden, die damit nach Abschluss der jeweiligen Arbeit von Interessierten weiterentwickelt werden können.

---

<sup>345</sup> In diesem Kapitel wurde eine Einschränkung auf Informatiksysteme als Unterrichtshilfen vorgenommen. Damit ist nicht impliziert, dass der Autor den Einsatz anderer Unterrichtshilfen geringerschätzt.



Großen Kindern dagegen würde ich sagen, daß das eine überaus schwierige Frage sei, die ich ihnen erst beantworten könne, wenn sie hübsch fleißig Biologie, Psychoanalyse und Soziologie von Grund auf gelernt hätten. Denn große Kinder sind sehr große Dummköpfe in allen Fragen der Aufklärung.

[Bernfeld 1981, S. 92]

## Kapitel 10

# Zusammenfassung

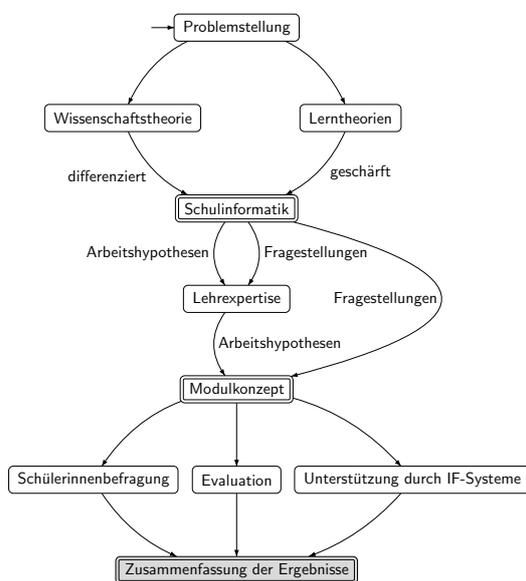


Abbildung 10.1: Überblick Forschungsgang

des Konzepts erwies sich die Untersuchung der Fragestellung nach der Veränderung des Bildes der Informatik bei Schülerinnen als aussagefähig. Die Ergebnisse zeigen eine tendenzielle Veränderung des Bildes der Informatik bei den Schülerinnen, wobei die Änderung quantitativ nicht so deutlich ausfällt, dass die mit dem Konzept verbundenen Anforderungen in ihrer Vielgestaltigkeit als eingelöst bezeichnet werden können. Als Konsequenz dieser Ergebnisse sollten Gewichtsverschiebungen vorgenommen werden. Eine mögliche Gestaltungsalternative besteht darin, den verpflichtenden Informatikunterrichts in der Sekundarstufe I einzuführen.

Das entwickelte Modulkonzept wurde in einer weiteren Untersuchung Grundlage zur Analyse des Informatikunterrichts herangezogen. Ergebnisse von Unterrichtsbeobachtungen im bayerischen Schulversuch zur Einführung von Informatik als Pflichtfach in der Sekundarstufe I (Gymnasium, Jahrgangsstufe 6) wurden auf das Modulkonzept bezogen. Es wurde deutlich, dass eine Zuordnung des beobachteten Unterrichts zu den Elementen des Modulkonzepts möglich ist. Dies ist insofern bedeutsam, als das Konzept mit Blick auf die Sekundarstufe II entwickelt wurde und mit der Untersuchung der stufenübergreifende Charakter für die Module „Informatiksysteme verantwortlich nutzen“ und „informatische Modellierung“ gezeigt werden konnte. Damit scheint es möglich, konkrete Elemente des Modulkonzepts bereits in der Sekundarstufe I frühzeitig zu fundamentieren, um sie zu einem späteren Zeitpunkt der allgemeinen informatischen Bildung

wieder aufzunehmen (spiraliges Curriculum).

Eine dritte Evaluationsstudie zielte darauf ab, Informatiksysteme zur Unterstützung der Lernprozesse zu erstellen. Dazu wurden Gestaltungsanforderungen aus Sicht der Fachwissenschaft und der Fachdidaktik zusammengetragen, die für einige Arbeiten gestaltungsleitend als Kriterien herangezogen werden konnten. Die Ergebnisse sind ermutigend, da gezeigt wurde, dass die Kriterien Hinweise zur Gestaltung geben, die über die (ausschließlich) fachliche Dimension hinausgehen. Die differenzierte Evaluation der entwickelten Informatiksysteme im Informatikunterricht wurde nicht als Gegenstand dieser Arbeit vorgenommen. Hier sind weitere Untersuchungen unverzichtbar. Rückmeldungen von Lehrerinnen zeigen, dass die entwickelten Informatiksysteme im Unterrichtszusammenhang eingesetzt werden.

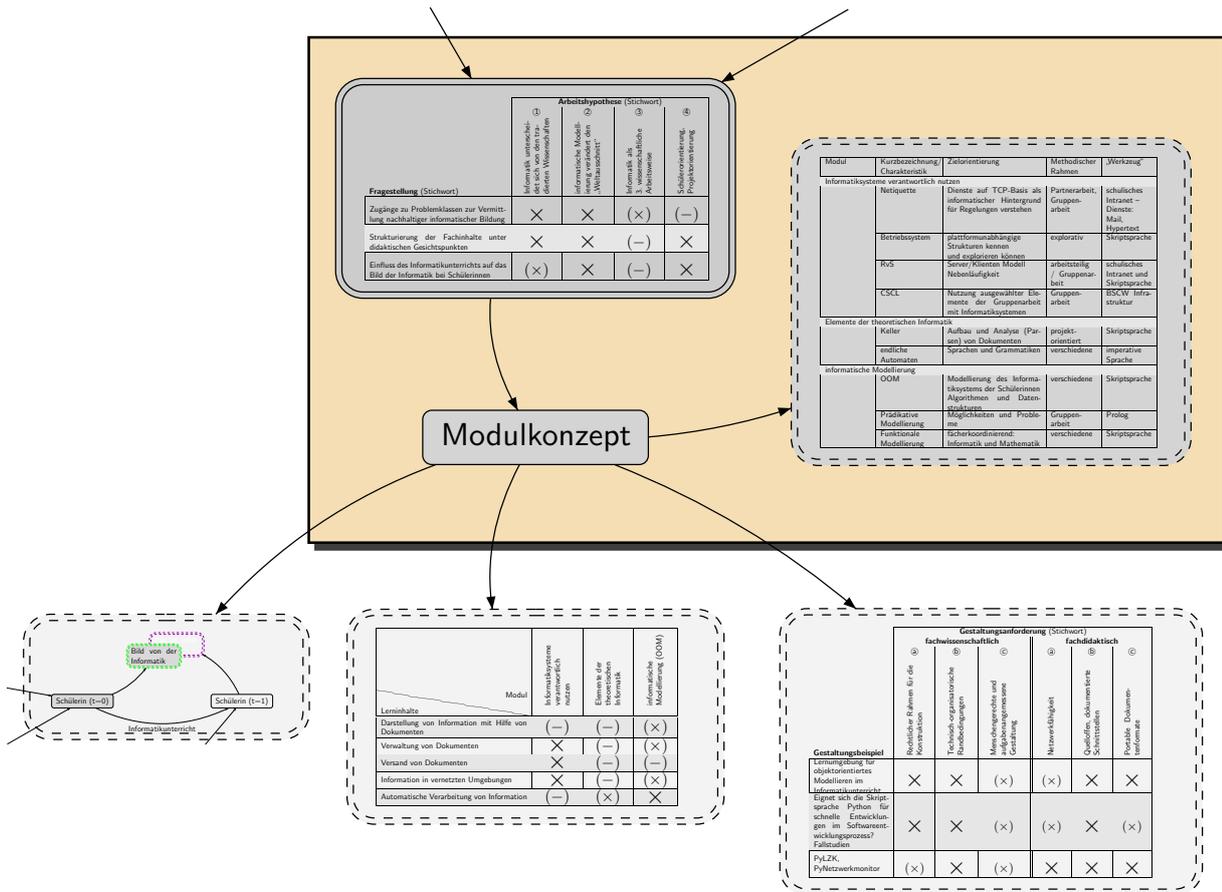
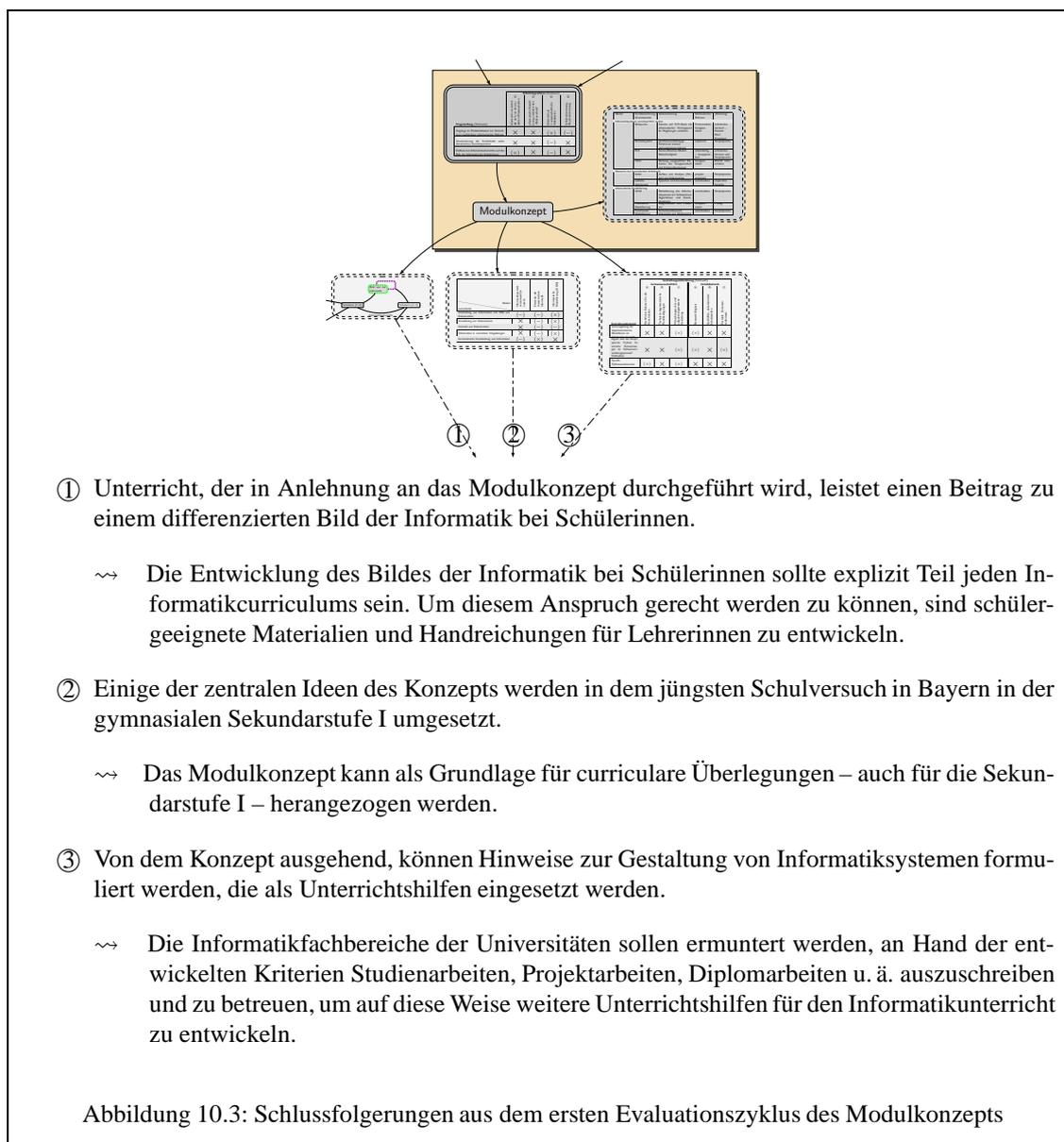


Abbildung 10.2: Übersicht zur Entwicklung und Evaluation des Modulkonzepts<sup>346</sup>

<sup>346</sup> Zuordnung der zusammenfassenden Dokumente zum Modulkonzept

Kapitel	Bezeichnung	Tabelle/Abbildung	Seite
Das Modulkonzept	Arbeitshypothesen und Fragestellungen Kreuzreferenzen	Tabelle 6.1	104
	Modulkonzept Beispiellübersicht	Tabelle 6.2	110
Einschätzung der Informatik durch Lernende	Änderung des Bildes der Informatik bei Schülerinnen (schematisch)	Abbildung 7.8	125
Informatikunterricht evaluieren	Lerninhalte Sekundarstufe I x Module	Tabelle 8.3	134
Informatik als Gegenstand und Mittel des Lernens	Anforderungen und Beispiele – Kreuzreferenzen	Tabelle 9.1	145

In der Abbildung 10.2 wird die Entwicklung und Evaluation des Modulkonzepts an Hand der begleitend erstellten Dokumente zusammenfassend dargestellt. In der Abbildung 10.3 sind eine Zusammenfassung der Ergebnisse und einige Schlussfolgerungen des ersten Evaluationszyklus des Modulkonzepts dargestellt.



Zum „Bild der Informatik“ bei Schülerinnen geben über die vorgestellten Untersuchungen hinaus weitere Quellen Hinweise. In der Untersuchung zur Analyse und Evaluation der Softwareentwicklung in Deutschland<sup>347</sup> wird festgestellt: „Es gibt wohl in keinem Bereich ähnlich verzerrte Vorstellungen über die Berufsbilder wie in der Informatik. In der breiten Öffentlichkeit werden mit dem Berufsbild Informatiker oder Softwareentwickler immer noch technikbegeisterte Programmierer verstanden. Unter diesem Eindruck kann es nicht verwundern, dass insbesondere Frauen sich nur sehr begrenzt für ein derartiges Studium interessieren“ [BMBF 2000, S. 187]. Das dazu Lehrerinnen qualifiziert sein müssen, die selbst über ein entwickeltes, differenziertes Bild der Wissenschaft Informatik verfügen, wird in den Untersuchungen von Peter BERGER deutlich: „Über ein differenziertes und mehr oder weniger elaboriertes Bild von der

<sup>347</sup> für das Bundesministerium für Bildung und Forschung (BMBF)

Wissenschaft Informatik verfügen naturgemäß vorwiegend [... Lehrerinnen] mit einem informatischen Hochschulstudium“ [Berger 2001, S. 257].

Die Entwicklung des vorgestellten Modulkonzepts war ausdrücklich auf die allgemein bildende Sekundarstufe II gerichtet. Die Anwendbarkeit auf curriculare Ausprägungen der Sekundarstufe I stellt eine Erweiterung dar, die nicht überschätzt werden sollte. Dass ein curriculärer Rahmen für grundlegende informatische Elemente in der Sekundarstufe I unabdingbar ist, zeigt eine soeben vorgelegte empirische Studie, ohne jedoch auf diese Lösungsmöglichkeit hinzuweisen. „Angesichts der beachtlichen Menge an Daten, die die Schüler im Laufe des Projekts auf ihren Computern gesammelt hätten, stellte sich für sie heraus, wie wichtig es sei, sich ein System anzulegen, in dem die Daten auch zu einem späteren Zeitpunkt noch wiedergefunden werden könnten. In verschiedenen Projekten sei mit den Schülern geübt worden, wie Informationen archiviert und Datenstrukturen angelegt werden können. [...] So stellte ein Lehrer fest, dass einige Schüler nach wie vor Probleme damit hätten, Dateistrukturen zu organisieren. Der Laptop böte hier zwar theoretisch einen Vorteil, praktisch sei es aber häufig schwierig, diese Kompetenzen im Unterricht auch allen Schülern zu vermitteln“ [Issing und Schaumburg 2002, S. 55].

Diese Kompetenzen sollten m. E. unter Verwendung eines Konzepts und der Verankerung im allgemein bildenden Schulfach Informatik ihren Platz in der Schule finden. Die Forderung nach Berücksichtigung informatischer Themen als Bestandteil der allgemeinen Bildung wird in der Studie [BMBF 2000] für das Bundesministerium für Bildung und Forschung (BMBF) erhoben: „Bereits frühzeitig muss die Bereitschaft zur Auseinandersetzung mit Themen der Informationstechnologie und ein selbstverständlicher Umgang mit diesen Kommunikationsmedien gefördert werden. Diese Grundlagen müssen bereits in den Schulen flächendeckend vermittelt werden“ [BMBF 2000, S. 187]. Die Ziele des Unterrichts eines Schulfachs Informatik unterscheiden sich in ihrer Qualität nicht von den Zielen anderer Unterrichtsfächer, wie nicht zuletzt in den Ausführungen von Werner HARTMANN und Jürg NIEVERGELT deutlich wird: „Erstens werden die für den Alltag unabdingbaren Fertigkeiten vermittelt. Zweitens erhalten Schülerinnen und Schüler einen Einblick in das Wesen und die intellektuellen Errungenschaften des Fachs. Und drittens wird ein Bewusstsein geschaffen für die Bedeutung und Rolle des Fachs in einem größeren gesellschaftlichen Rahmen. Im Informatikunterricht ist es bis heute noch nicht in ausreichendem Maß gelungen, diesen drei Ansprüchen gerecht zu werden“ [Hartmann und Nievergelt 2002, S. 476].

Die im Zusammenhang mit dieser Arbeit begonnene Entwicklung von Konzepten zur inhaltlich-organisatorischen Strukturierung von Lernprozessen auf fachlicher und lerntheoretischer Basis von Informatik (als Kulturtechnik) legt die Neugestaltung/Umgestaltung grundlegender Gestaltungsprinzipien didaktischen Handelns auf der Basis eines problem- und projektorientierten Unterrichts nahe. Dabei wird der Fähigkeit der Schülerinnen, Probleme zu identifizieren, eine besondere Bedeutung zugemessen.

In jeder Ausprägung von der schulischen Bildung über die betriebliche Ausbildung bis hin zur Hochschulausbildung ist informatische Bildung ein wichtiger Orientierungspunkt des allgemein bildenden Anspruchs. Dabei sind folgende Kriterien zu berücksichtigen:

- verantwortlicher Umgang mit Werkzeugen der Informatik,
- strukturgeleitete Planungsfähigkeit,
- Überprüfung an realen Maschinen,
- Infragestellung von (Teil-)Ergebnissen,
- Kommunizierbarkeit von Ergebnissen.

In der vorliegenden Arbeit wurde gezeigt, dass die Schulinformatik einen notwendigen Beitrag zur Allgemeinbildung leisten muss. Dies ergibt sich zum Einen aus der Analyse der Ergebnisse der fachwissenschaftlichen Diskussion und zum Anderen aus der besonderen Arbeitsweise der Informatik. Damit ist es notwendig, den pädagogischen Prozess an dem Paradigma Berufswelt auszurichten<sup>348</sup>. Die Gestaltungsanforderungen wurden konzeptionell zu dem Modulkonzept verdichtet, das auch über den Einzelfall hinaus eine qualitative Dimension des Informatikunterrichts deutlich macht.

<sup>348</sup> Leitkonzepte Projekt, Produkt, Team, Diskussion, beraten, delegieren, mitbestimmen und kooperieren, vgl. Abschnitt 4.2.1, S. 68

# Kapitel 11

## Offene Fragen

Haben die Systemtheoretiker, Konstruktivisten, Poststrukturalisten, Cultural Studies-Wissenschaftler Recht? Sie behaupten, in »unseren hochkomplexen, pluralistischen Gesellschaften« dürfe keiner sich für besser halten als andere. Toleranz [aber] schließe Qualitätsurteile aus. [...]  
 Oder haben die kritischen Theoretiker Recht? Sie argumentieren, dass Toleranz in Interesselosigkeit und in den Meinungsterror der political correctness umschlägt, wenn man die Qualität von Gesellschaften, Medien und Medienprodukten nicht analysiert. Sie sagen, dass wir die Freiheit haben, Qualität und Nicht-Qualität, Gutes und Schlechtes, Wahres und Falsches zu benennen und Qualitäten zu erkämpfen: demokratische Qualität, Lebensqualität, ästhetische Qualität.

[Prokop 2002, S. 10]

In der vorliegenden Arbeit wird Informatik in das Zentrum von Bildungsbemühungen gestellt. Eine solide fachwissenschaftliche Basis stellt Anker und Gestaltungshintergrund für fachdidaktische Prozesse dar. So sehr die modernen Pädagoginnen sich auch reformpädagogischen, ganzheitlichen Auffassungen des Lehr-/Lernprozesses verpflichtet fühlen, die Fachwissenschaft bleibt eine unabdingbare Grundlage für verantwortliches pädagogisches Handeln.

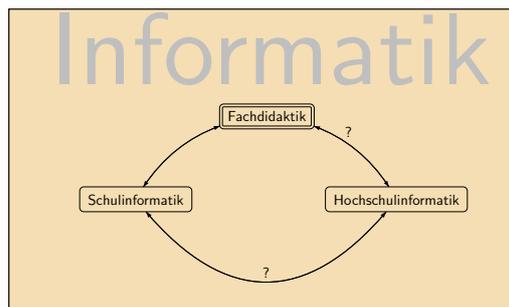


Abbildung 11.1: Fachdidaktik, Schulinformatik und Hochschulinformatik

Die Beziehungen zwischen der Fachdidaktik und der Schulinformatik wurden in der vorliegenden Arbeit herausgearbeitet – aber welche Beiträge leistet die Fachdidaktik Informatik zur Hochschuldidaktik?

Auf Fragen der Modellierung, der Programmierung und des Erwerbs der dazu notwendigen qualifizierenden Grundlagen wird organisatorisch geantwortet – durch Veranstaltungsformen, die kaum didaktisch reflektiert werden.

Überlegungen, die eher pragmatischen Charakter besitzen, wurden in Bezug auf die Gestaltung von Informatikstudiengängen bereits früh umgesetzt (exemplarisch: Praktika, Projektgruppen).

Vereinzelt finden sich Ansätze zur Bearbeitung fachdidaktischer Fragestellungen der Hochschulinformatik.

Zu didaktischen Modellen zur Umsetzung der Modellierung – insbesondere des Programmierens – wurden internationale Studien vorgelegt, die nach [Kaasbøll 1998a, S. 196f] drei verschiedenen didaktischen Ansätzen zugeordnet werden können. In Tabelle 11.1 werden die von Jens J. KAASBØLL identifizierten Ansätze auf die zu Grunde liegenden didaktischen Modelle bezogen und durch charakteristische Elemente beschrieben.

Ansatz (lt. KAASBØLL)	Lerntheoretischer Hintergrund	Charakteristik
Semiotic ladder	Fachlogik $\implies$ Lehr-/Lernlogik (Abbilddidaktik)	Syntax, Semantik, Pragmatik
Cognitive objectives taxonomy	Bloomsche Taxonomie (Kognitivismus)	Programm benutzen, lesen, ändern und ggf. erstellen
Problem solving	pragmatischer Konstruktivismus	Lernerorientiert – problemorientiert

Tabelle 11.1: Didaktische Ansätze zur Programmierung in der Hochschulinformatik (international)

Mit [Harrer und Schneider 2002], [Schneider 2002] wurden Forschungsbeiträge zu einer sich entwickelnden Hochschuldidaktik der Informatik vorgestellt. Es steht zu erwarten, dass die fachdidaktischen For-

schungsaktivitäten in den Bereichen Schulinformatik und Hochschulinformatik miteinander verzahnt und aufeinander bezogen werden.

In der Fachwissenschaft Informatik zeichnet sich ein erweitertes Verantwortungsbewusstsein für die Ergebnisse und Auswirkungen der informatischen Gestaltungsprozesse ab, das dazu führt, dass die menschengerechte Gestaltung von Informatiksystemen ein wesentliches Element im Qualifikationsprozess von (Diplom-)Informatikerinnen wird. Vereinzelt liegen Erfahrungsberichte zu Fragestellungen in diesem Umfeld vor. Hier können fachdidaktisch geleitete Untersuchungen die Entscheidungsfindung unterstützen und die Effizienz erhöhen, wie beispielsweise von KAASBØLL [Kaasbøll 1998b] dokumentiert wird.

Eine [fach-]didaktisch geleitete Reflexion der Strukturen, Inhalte und Methoden der Hochschulinformatik kann dazu beitragen, die (zu) hohe Studienabbrecherquote zu verringern. Didaktisch reflektierte Gestaltungshinweise können darüber hinaus Kompetenzen bei Lehrenden und Studierenden optimieren.

Mehr und mehr Nichtinformatikerinnen benötigen ein informatisches Begriffsrepertoire, damit sie die Prozesse, die durch Informatiksysteme beeinflusst werden, angemessen beschreiben können. Eine Form der Integration von Elementen der Fachwissenschaft Informatik in andere Wissenschaften besteht in dem Aufbau von „Bindestrich-Informatiken“ (exemplarisch: Bioinformatik, Geoinformatik). Hier zeigt sich die Sinnhaftigkeit von Bildungsvoraussetzungen, die durch die allgemeine Bildung zur Verfügung gestellt werden.

Die Aufgabe, für berufliche Qualifikationen die notwendigen schulischen Voraussetzungen anzugeben, führt zu der Überlegung, welche Elemente informatischer Bildung in der Sekundarstufe I allgemeinbildenden Charakter haben. Die damit verbundenen konkreten Fragen sind bisher nicht hinreichend von der Fachdidaktik geklärt.<sup>349</sup>

In der Tabelle 11.2 werden wichtige Untersuchungsdimensionen des Informatikunterrichts zusammengestellt. Zu jeder der angegebenen Dimensionen wird der Beitrag der vorliegenden Arbeit qualifiziert. Damit wird deutlich, dass in den beiden Bereichen Lernmittel<sup>350</sup> und Lebensweltbezug Forschungsbedarf besteht.

Informatikunterricht			
Fachwissenschaft	Lerntheorien	Lernmittel	Lebensweltbezug
✓	✓ <sup>351</sup>	(✓) prototypisch	–

Tabelle 11.2: Untersuchungsdimensionen des Informatikunterrichts<sup>352</sup>

Die Abbildung 11.2 verdeutlicht Elemente, die bezogen auf die Schülerinnen im Zusammenhang mit dem Informatikunterricht bedeutsam sind. Von diesen wurde nur ein kleiner Teil in dieser Arbeit untersucht. Offen bleiben Fragen, die mit den Schlagworten „mentale Modelle“ und Fehlvorstellungen (misconceptions) bezeichnet werden.<sup>353</sup> Der Aufbau von Wissensnetzen in einem mehrstufigen Prozess wurde zur Erstellung

<sup>349</sup> Der für diese Arbeit dokumentierte Informatikunterricht der Jahrgangsstufe 6 findet an Gymnasien statt und ist damit nicht primär auf die Schaffung von Voraussetzungen für eine berufliche Ausbildung nach dem 10. Schuljahr orientiert.

<sup>350</sup> Die in der Evaluation des Informatikunterrichts in bayerischen Gymnasien (vgl. Kapitel 8) gesammelten Ergebnisse weisen darauf hin, dass den Informatiksystemen, die im Unterricht eingesetzt werden, die Funktion zentraler Hilfsmittel für den Informatikunterricht zukommt. Allerdings wurde der Aufwand, der notwendig ist, um diese Mittel inklusiv der Infrastruktur (Handlungsrückgrat, vgl. Abschnitt 6.1, S. 104) konkret zur Verfügung zu stellen, bei der Evaluation nicht weiter untersucht.

<sup>351</sup> Zur Umsetzung des Modulkonzepts kann festgestellt werden:

Die Zugänge zu informatisch zu bearbeitenden Problemklassen wurden primär mit Blick auf die Fachwissenschaft strukturiert, wobei der informatischen Modellierung eine Schlüsselfunktion zukommt. Die Frage der lerntheoretischen Orientierung wurde unter dieser Fragestellung nicht beleuchtet.

<sup>352</sup> im Rahmen dieser Arbeit

✓ betrachtet  
(✓) exemplarisch untersucht  
– nicht explizit untersucht

<sup>353</sup> Um sich mit einer detaillierten Sicht auf Aneignungsprozesse zu beschäftigen, bedarf es zum Einen einer klaren inhaltlich strukturierten Ebene und darüber hinaus einer Festlegung auf eine Theorie des Lernens. Wird auf dieser Basis Forschung betrieben, wird m. E. das Gebiet der Informatikdidaktik zu früh auf Details eingeschränkt.

von Informatiksystemen zur Unterstützung des Informatikunterrichts vorgenommen. Wissensnetze eignen sich zur Vorbereitung der fachlichen Struktur bei der Vorbereitung der Unterrichtsgestaltung durch die Lehrerin. Auf *Kernideen* und *Fundamentale Ideen* wurde allgemein im Zusammenhang mit der Diskussion der Entwicklung von Lerntheorien hingewiesen. Der Ansatz der Fundamentalen Ideen der Informatik kann mit Hilfe der Überlegungen zu Kernideen eine Weiterentwicklung erfahren.

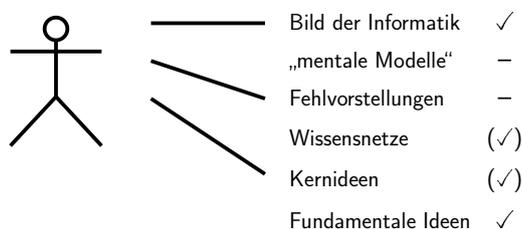


Abbildung 11.2: Schülerin und Bezüge zum Informatikunterricht

Findet Informatik als allgemein bildendes Fach Eingang in den Fächerkanon, so werden alle Schülerinnen von der informatischen Bildung „betroffen“. Zu geschlechtsspezifischen Zugangsweisen sind einige wissenschaftliche Arbeiten verfügbar, die einen grossen Fundus an Ideen darstellen (vgl. [Funken u. a. 1996], [Faulstich-Wieland und Nyssen 1998], [Westram 1999]). Es sind Konzepte zu entwickeln und zu evaluieren, die den besonderen Zugangsweisen von Mädchen und Jungen zur Informatik Rechnung tragen.<sup>354</sup>

<sup>354</sup> Überlegungen zur zeitweiligen Aufhebung der Koedukation sollen dabei nicht ausgespart werden.



# **Anhänge**



- A Expertise durch Interviews
- B Befragung zur Prädikativen Modellierung
- C Bild der Schülerinnen von der Informatik
- D Gruppeninterview
- E Hospitationsprotokolle zur Evaluation
- F Exemplarische Unterrichtseinheiten
- G Schemata zum Problemlösen

Literaturverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Liste der Algorithmen

Namens- und Organisationsindex

Sachindex

Abkürzungsverzeichnis



# Anhang A

## Expertise durch Interviews – Übersicht und Ergebnisse

### A.1 Anfrage mit der Bitte um Mitarbeit

Universität Dortmund  
Lehrstuhl Informatik XII  
Didaktik der Informatik  
Ludger Humbert

44221 Dortmund

Tel: 0231-755 6142 / Fax: 0231-755 6116

E-Mail: [humbert@is12.cs.uni-dortmund.de](mailto:humbert@is12.cs.uni-dortmund.de)

---

Dortmund, im Oktober 2000

---

Liebe Kollegin,

im Rahmen meiner Forschungsarbeiten möchte ich mit Ihnen ein Interview zu Konzepten des Informatikunterrichts in der gymnasialen Oberstufe und ihrer unterrichtlichen Umsetzung führen.

Sie können als kompetente Gesprächspartnerin für meine Forschungen eine wichtige Quelle für die Diskussion konzeptioneller Überlegungen sein.

Auf diesem Weg möchte ich Sie bitten, sich mit mir in Verbindung zu setzen. Die Planung eines Interviewtermins möchte ich gerne frühzeitig in meine Terminplanung einbauen.

In der Hoffnung, dass Sie durch Ihre Mitarbeit mein Forschungsvorhaben befördern möchten, verbleibt

mit freundlichen Grüßen

Ludger Humbert

## A.2 Leitfaden für das Expertinneninterview

### A.2.1 Abschnitt I – Daten und Fragen zur Person

Wann sind Sie geboren?	
Seit wann beschäftigen Sie sich mit Themen aus der Informatik?	
Seit wann unterrichten Sie Informatik?	
Welche Fächer haben Sie primär für das Lehramt studiert?	
An welcher Universität?	
Wann haben Sie Ihr erstes Staatsexamen erworben?	
Warum gerade Informatik?	offen
Wann das zweite Staatsexamen?	
Schultyp?	
warum?	offen
Weitere Qualifikationen?	
Zweiter Bildungsweg?	
Wieviele Schulwechsel?	
Position in der Schule?	
Mitglied in einer der Vereinigungen für Informatiklehrerinnen?	
regelmässige Leserin von Zeitschriften aus dem Bereich Informatik und Schule?	
	LogIn
	IBU
	C&U
<b>aktuelle</b> Mitarbeit	
	KM - Lehrplanentwicklung
	Mittelbehörde - Lehrerfortbildung IF
	Bund - Schulen ans Netz
	Königstein - fachdidaktische Gespräche
	lokale Initiative(n)
Betreuung Webangebot URL:	

### A.2.2 Abschnitt II – Interviewthesen

These	ja	nein
Es gibt keine Didaktik der Informatik.		
Informatik in der Schule ist überflüssig.		
Informatikunterricht ist immer auch Programmierunterricht.		
Es gibt gute Konzepte, nur keine Lehrer, die sie umsetzen.		
Die Informatikwerkzeuge determinieren den konkreten Unterricht.		
Die Schule kann nicht mithalten, deshalb wird Informatikunterricht immer unattraktiver werden.		
Informatik wird Biologie ablösen und Physik und Chemie weiter verdrängen		

### A.2.3 Abschnitt III – Konzepte

Welches Konzept vertritt die Interviewpartnerin für die Schule (Sekundarstufe II, allgemeinbildend)?

Basis	nein	eher weniger	eher mehr	ja
Objektorientierung				
Algorithmen und Datenstrukturen				
funktionale Modellierung				
wissensbasierte Modellierung				
theoretisch orientierte Ausrichtung				
...				

### A.2.4 Abschnitt IV – Methodisch-didaktische Aspekte

Wie setzt die Interviewpartnerin ihr Konzept in der Schule um?

Schemata zur Unterstützung des Problemlösungsprozesses	-	+
PAP		
Struktogramme		
UML		
ER-Diagramme		
...		

### A.2.5 Abschnitt V – [Software-] Technische Unterstützung

Unterstützung der Modellierung	-	+
Entwicklungsumgebungen im Unterricht?		
professionelle CASE-Werkzeug(e)?		

### A.3 Auswertungen

#### A.3.1 Abschnitt I – Daten und Fragen zur Person

Frage	Antworten – Auswertung	I
Wann sind Sie geboren?	$\bar{x} = 1958$	1
Seit wann beschäftigen Sie sich mit Themen aus der Informatik?	$\bar{x} = 1979$	2
Seit wann unterrichten Sie Informatik?	$\bar{x} = 1986$	3
Welche Fächer haben Sie primär für das Lehramt studiert?	5 I, 12 M, 9 P <sup>355</sup>	4
An welcher Universität?	kein Trend	5
Wann haben Sie Ihr erstes Staatsexamen erworben?	$\bar{x} = 1983$	6
Warum gerade Informatik?	offen	7
Wann das zweite Staatsexamen?	$\bar{x} = 1986$	8
Schultyp?	13 Gy/GE <sup>356</sup>	9
Weitere Qualifikationen?	9	11
Zweiter Bildungsweg?	2	12
Wieviele Schulwechsel?	$\bar{x} = 1$	13
Position in der Schule?	50% (8) FK IF <sup>357</sup>	14
Mitglied in einer der Vereinigungen für Informatiklehrerinnen?	88%	15
regelmässige Leserin der Zeitschrift ...		
LOG IN	100%	16
IBU <sup>358</sup>	38%	17
C&U <sup>359</sup>	38%	18
sonstiges	44% c't	19
<b>aktuelle Mitarbeit</b>		
KM - Lehrplanentwicklung	50%	20
Mittelbehörde - Lehrerfortbildung IF	63%	21
Bund - Schulen ans Netz	25%	22
Königstein - fachdidaktische Gespräche	75%	23
lokale Initiative(n)	81%	24
Betreuung Webangebot URL	75%	25

<sup>355</sup> I – Informatik, M – Mathematik, P – Physik

<sup>356</sup> Gy – Gymnasium, GE – Gesamtschule

<sup>357</sup> FK IF – Fachkonferenz Informatik – Vorsitz

<sup>358</sup> IBU – Informatik betrifft uns (Erscheinen eingestellt)

<sup>359</sup> C&U – Computer und Unterricht

### A.3.2 Abschnitt II – Interviewthesen

These	ja	nein	k.A.	II
Es gibt keine Didaktik der Informatik.	1	15	-	1
Informatik in der Schule ist überflüssig.	0	16	-	2
Informatikunterricht ist immer auch Programmierunterricht.	10	6	-	3
Es gibt gute Konzepte, nur keine Lehrer, die sie umsetzen.	4	11	1	4
Die Informatikwerkzeuge determinieren den konkreten Unterricht.	2	14	-	5
Die Schule kann nicht mithalten, deshalb wird Informatikunterricht immer unattraktiver werden.	0	16	-	6
Informatik wird Biologie ablösen und Physik und Chemie weiter verdrängen.	2	13	1	7

### A.3.3 Abschnitt III – Konzepte

Welches **Konzept** vertritt die Interviewpartnerin für die Schule (Sekundarstufe II, allgemeinbildend)?

Basis	nein	eher weniger	eher mehr	ja	III
<b>Objektorientierung</b>	-	-	5	11	1
<b>Algorithmen und Datenstrukturen</b>	-	-	9	7	2
funktionale Modellierung	3	9	2	2	3
<i>wissensbasierte Modellierung</i>	-	7	7	2	4
<i>theoretisch orientierte Ausrichtung</i>	2	5	5	4	5
<i>Anwendungsbereiche - Bewertung - Hintergrund</i> <sup>360</sup>					6
<i>Informatik und Gesellschaft</i> <sup>361</sup>					7
technische Informatik / Wirkprinzipien <sup>362</sup>					8
Rechnernetze und verteilte Systeme <sup>363</sup>					9
informatisches Modellieren <sup>364</sup>					10
Teamarbeit – Projektarbeit <sup>365</sup>					11
sieben weitere einzelne Nennungen ...					
programmiersprachenorientierter Unterricht <sup>366</sup>					19
mathematisch orientierter Informatikunterricht					20

<sup>360</sup> III 6: hier wird das Berufsverständnis von Informatiklehrerinnen und -lehrern deutlich: die Verschränkung der inhaltlichen Ebene über die Thematisierung des Zusammenhangs ist eine originäre Aufgabe des Informatikunterrichts – wird für wichtiger erachtet, als die funktionale Modellierung

<sup>361</sup> III 7: Berufsbilder, Geschichte der Informatik wurden ebenfalls explizit genannt

<sup>362</sup> III 8: hier scheiden sich die Geister: dreimal wurde dieser Bereich explizit aufgenommen, aber als Gegenstand ausgeschlossen; zwei Nennungen äußern die absolute Notwendigkeit und drei Nennungen erachten diesen Bereich eher mehr für wichtig

<sup>363</sup> III 9: vier Nennungen wollen diesen Bereich explizit aufgenommen wissen (mindestens mit eher mehr)

<sup>364</sup> III 10: Prinzipien der informatischen Modellierung explizit im Informatikunterricht einen Platz zu geben halten immerhin drei Lehrerinnen für unabdingbar

<sup>365</sup> III 11: siehe vor

<sup>366</sup> III 19, III 20: diese Orientierungen werden explizit benannt – aber abgelehnt

### A.3.4 Abschnitt IV – Methodisch-didaktische Aspekte

Wie setzt die Interviewpartnerin ihr Konzept in der Schule um?

Schemata zur Unterstützung des Problemlösungsprozesses	-	+	IV
PAP	8	7	1
Struktogramme	5	10	2
UML	4	11	3
ER-Diagramme	6	8	4
Pseudo-Code	1	2	5
Und/Oder Bäume	-	1	6
CRC	-	1	7
Literate Programming	-	1	8
Petrinetzdarstellung	-	1	9

### A.3.5 Abschnitt V – [Software-] Technische Unterstützung

	-	didaktisch gestaltet	+	V
Entwicklungsumgebungen im Unterricht?	1	1	7	1
professionelle CASE-Werkzeug(e)?	6	4	2	2

Die vollständige Dokumentation der Transkripte findet sich in einem eigenständigen Dokument, das nicht vollständig öffentlich zugänglich gemacht wird, da es wegen der darin enthaltenen Vielzahl von personenbezogenen Angaben Rückschlüsse auf konkrete Personen zulässt.

## Anhang B

# Befragung zur Prädikativen Modellierung

Unterrichtsreihe zur prädikativen Modellierung: durchgeführt im Zusammenhang mit dem Tagespraktikum der Didaktik der Informatik des Fachbereichs Informatik an der Universität Dortmund im Wintersemester 1998/1999.

- Dokumentation der Materialien zum Tagespraktikum,
- Erhebung zum Abschluss der Unterrichtsreihe und damit zum Abschluss des ersten Unterrichtshalbjahrs in einem Koppelkurs 11.1 und 12.1 mit insgesamt 25 Schülerinnen und Schülern.

### B.1 Dokumentation der Fragen

1. Was ist Informatik?
2. Welche Inhalte des Unterrichts haben dir nicht gefallen?
3. Welche Inhalte des Unterrichts haben dir besonders gut gefallen?
4. Welche Methoden des Unterrichts haben dir nicht gefallen?
5. Welche Methoden des Unterrichts haben dir besonders gut gefallen?

---

Die vorgenannten Fragen wurden den Schülerinnen zur Arbeit in Gruppen von je 3 bis 5 Schülerinnen vorgelegt. Jede Gruppe erhielt eine Folie zur Dokumentation der Ergebnisse.

### B.2 Dokumentation der Ergebnisse

Die Ergebnisse wurden in einer Unterrichtsstunde erarbeitet und und anschliessend diskutiert. Diese Unterrichtsstunde fand am Freitag, 29. Januar 1999 statt.

Im Folgenden werden alle schriftlich festgehaltenen Antworten auf die Fragen angegeben.<sup>367</sup>

---

<sup>367</sup> Transkripte der Folien, die zum Vortrag benutzt wurden

**zu 1. Was ist Informatik?**

1. Informatik ist ... die Computer Wissenschaft
2. Informatik ist ...
  - (a) Information + Automatik? Frag' ich mich manchmal auch!
  - (b) Das Lösen und Bearbeiten von Problemen oder Daten mit Hilfe von EDV-Anlagen,
3. Informatik ist ...
  - (a) automatische Verarbeitung von Informationen
  - (b) Information + Automatik
  - (c) Probleme lösen mit Hilfe des Computers
4.
  - (a) Der Begriff Informatik setzt sich aus den Wörtern Information und Automatik zusammen.
  - (b) Informatik ist die automatische Verarbeitung von Informationen.
5. Informatik ist ... Problemlösende, Informations- und Datenverarbeitende Automatisierung.
6. Informatik ist ... das Lösen und Bearbeiten von Problemen oder Daten mit Hilfe von EDV-Anlagen. Dieses Fach wird auch als Wissenschaft angesehen.

**zu 2. Welche Inhalte des Unterrichts haben dir nicht gefallen?**

1.
  - (a) unverständlicher Unterricht (teilweise auch für Könner)
  - (b) Kurs war zu groß - keinen guten Bezug zu den Schülern
  - (c) Themen angefangen, nicht beendet
2.
  - (a) Python
  - (b) fachspezifischer Theorieunterricht (sometimes zu trocken)
  - (c) Gruppenarbeiten
  - (d) Ahnentafel
3. zu viel Theorie
4.
  - (a) 11er und 12er waren zusammen in einem Kurs
  - (b) etwas zuviel Theorie
  - (c) zu langes Aufhalten an den Stammbäumen
5.
  - (a) die komplizierten Erklärungen
  - (b) Einzelarbeit
  - (c) oberflächliches Anschneiden von Programmiersprachen
6.
  - (a) zu wenig Internetarbeit
  - (b) sehr mäßige Klärung von Prolog bzw. Labyrinth

**zu 3. Welche Inhalte des Unterrichts haben dir besonders gut gefallen?**

1. (a) selbstständig Arbeiten  
(b) liberaler Lehrer  
(c) E-Mail-Adresse für jeden
2. (a) Labyrinth  
(b) praxisorientiertes Arbeiten  
(c) E-Mail
3. Labyrinth
4. (a) zwischendurch selbstständiges Arbeiten am Computer  
(b) eigene E-Mail Adresse  
(c) Labyrinth
5. (a) Gruppenarbeit  
(b) Beispiele  
(c) Atmosphäre
6. Internetarbeit (HTML)

**zu 4. Welche Methoden des Unterrichts haben dir nicht gefallen?**

1. siehe oben
2. (a) Ihr Tafelbild ist manchmal undurchschaubar.  
(b) Hin und wieder machen Sie einen genervten Eindruck, leider können nicht alle sofort den Durchblick für Informatik gewinnen.  
(c) Bevor Sie anfangen umfangreich über Python etc. . . . zu erzählen, sollten Sie sich vergewissern, dass auch alle sinnvoll den Computer benutzen können. (Wie schalte ich ihn an?)  
(d) Ihr Theorieunterricht ist manchmal zu fachspezifisch, dann heben Sie förmlich ab!  
(e) Und bitte versuchen Sie nicht, uns für die Informatik zu begeistern!  
(f) Aber wir müssen auch ein gutes Haar an Ihnen lassen, durch Ihre netten Folien versuchen Sie uns einiges wieder zu verdeutlichen.
3. (a) zu wenig Erklärungen bei den einzelnen Themen,  
(b) zu wenig Praxis (unnötige Themen)  
(c) unübersichtliche Tafelbilder
4. in der Einzelstunde häufig nur Theorie und keine Arbeit am Computer
5. Einzelarbeit
6. *keine Angabe*

**zu 5. Welche Methoden des Unterrichts haben dir besonders gut gefallen?**

1. siehe oben
2. *keine Angabe*
3. (a) viel praktische Arbeit (am Computer)  
(b) Gruppenarbeit
4. (a) praktische Arbeit am Computer  
(b) Unser Lehrer gab oft Aufgaben, die wir dann selbstständig lösen mussten
5. (a) Gruppenarbeit  
(b) Projektarbeit
6. Arbeit mit den 12ern

**zu 6. Sonstiges**

1. *keine Angabe*
2. Werden wir aus diesem Kurs gehen und mit dem Computer sinnvoll umgehen können?
3. *keine Angabe*
4. (a) Wir würden gerne noch etwas über das Internet erfahren  
(b) Was macht ein Informatiker noch in der Praxis?
5. Wir möchten noch mehr über die Betriebssystemverwaltung wissen.
6. *keine Angabe*

# Anhang C

## Bild der Schülerinnen von der Informatik

### C.1 Dokumentation des Fragebogens

Fragebogen für Schülerinnen und Schüler des 11. Jahrgangs, die einen Informatikkurs gewählt haben. lh

---

Fragebogenkennung	Geschlecht (w/m)	Bearbeitungsdatum

Als vierstellige **Fragebogenkennung** verwenden Sie jeweils den **ersten** und **letzten** Buchstaben der **Vornamen** ihrer Eltern.

---

1. Nehmen Sie Stellung zu den folgenden Behauptungen, indem Sie jeweils ein Kreuz in jeder Zeile machen:

	trifft voll zu	trifft zu	trifft nicht zu	trifft überhaupt nicht zu
Das Schulfach Informatik fördert Kommunikation mit Hilfe des Schulintranets.				
Das Schulfach Informatik fördert die Arbeit in Gruppen.				
Das Schulfach Informatik erzwingt das Erlernen komplizierter Sprachen.				
Das Schulfach Informatik erleichtert die Aktualisierung von Präsentationen.				

**2. Was ist Informatik? - Bitte höchstens drei Aussagen ankreuzen.**

Nutzen Sie die freie Zeile für eine eigene Erklärung.

Informatik ist	
die Computerwissenschaft.	<input type="checkbox"/>
die Wissenschaft zur Entwicklung von künstlichen Intelligenzen.	<input type="checkbox"/>
die Wissenschaft zur Rationalisierung der geistigen Arbeit.	<input type="checkbox"/>
die Lehre von der Bedienung von Computern.	<input type="checkbox"/>
die Wissenschaft für das Internet.	<input type="checkbox"/>
geschaffen worden, um Kriege effektiver planen zu können.	<input type="checkbox"/>

**3. Für mein persönliches Lernen (dies gilt vor allem für andere Fächer) hat sich durch die Informatische Bildung folgendes verändert:**

...

**4. Sichten auf die Informatik. Bitte ordnen Sie die folgenden drei Sichten auf die Informatik zeitlich. Dazu schreiben Sie die Zahlen 1 bis 3 rechts neben die jeweilige Darstellung (1 soll die älteste, 3 die aktuellste Sicht bezeichnen).**

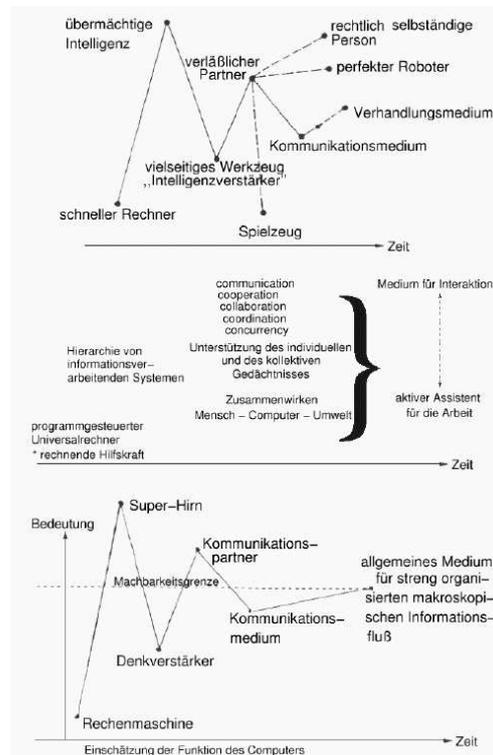


Abbildung C.1: Fragebogen – Leitbilder der Informatik

5. Bitte nehmen Sie Stellung zu den folgenden Behauptungen  
(je Aussage nur *ein* Kreuz):

	eingetroffen	möglich	unmöglich/falsch
In wenigen Jahren wird es das <b>papierlose Büro</b> geben.			
Die Kapazität von <b>fünf Computern reicht</b> für alle Berechnungen auf der ganzen Welt völlig aus.			
In einem <b>Auto</b> befinden sich ca. 70 Informatiksysteme.			
Das <b>Haus</b> alarmiert die Polizei, weil jemand einbrechen will.			
<b>Schachcomputer schlagen jeden</b> menschlichen Spielpartner.			
Informatiksysteme stellen eine <b>Konkurrenz für die Intelligenz</b> der Menschen dar.			

---

Die Ergebnisse dieses Fragebogens werden **nicht für die Leistungsbewertung** benutzt.  
Danke schön.

## C.2 Dokumentation der Ergebnisse

Auswertung:

tabellarische Auflistung der Ergebnisse.

Geschlecht (w/m)	Bearbeitungsdatum
9/9	

1. Nehmen Sie Stellung zu den folgenden Behauptungen, indem Sie jeweils ein Kreuz in jeder Zeile machen:

Das Schulfach Informatik ...	trifft voll zu	trifft zu	ja $\Sigma$	trifft nicht zu	trifft überhaupt nicht zu	nein $\Sigma$
fördert Kommunikation mit Hilfe des Schulintranets.	3	13	<b>16</b>	2	0	<b>2</b>
fördert die Arbeit in Gruppen.	4	11	<b>15</b>	3	0	<b>3</b>
erleichtert die Aktualisierung von Präsentationen.	1	10	<b>11</b>	6	1	<b>7</b>
erzwingt das Erlernen komplizierter Sprachen.	4	5	<b>9</b>	7	2	<b>9</b>

2. Was ist Informatik? - Bitte *höchstens drei* Aussagen ankreuzen.

Informatik ist ...

<b>die Lehre von der Bedienung von Computern.</b>	15
<b>die Computerwissenschaft.</b>	13
<i>freie Antworten ...</i>	8
die Wissenschaft zur Entwicklung von künstlichen Intelligenzen.	4
die Wissenschaft zur Rationalisierung der geistigen Arbeit.	4
die Wissenschaft für das Internet.	2
geschaffen worden, um Kriege effektiver planen zu können.	1

Die folgenden acht *freien Antworten* wurden (je einmal) angegeben:

- ... Problemlösung mit Informatiksystemen
- ... sehr riskant
- ... das Programmieren von Computerprogrammen
- ... ein Fach, wo man lernt, mit Computern umzugehen
- ... kann gefährlich werden.
- ... wichtig im jetzigen Leben
- ... die Lehre der Computerwissenschaft u. des Bedienens

3. **Für mein persönliches Lernen (dies gilt vor allem für andere Fächer) hat sich durch die Informatische Bildung folgendes verändert:**<sup>368</sup>

- Das Denken bei der Mathematik hat sich leicht gesteigert.
- Ich bin ordentlicher geworden: Weil in der Informatik nur ordentlich strukturierte Programme Erfolg bringen,
- Die Informatische Bildung hat mir in anderen Fächern nicht viel genutzt (ist nicht aufgefallen). Außer vielleicht ein bisschen in Mathematik. Im Umgang mit dem PC zuhause hat mir die Informatische Bildung jedoch geholfen.
- In anderen Fächern brauche ich informatisches Wissen eigentlich überhaupt nicht, jedoch ist es nützlich, wenn man einen PC besitzt, um die Arbeitsweise des PC's kennen zu lernen und um mit ihm umgehen zu können.
- Ich komme schneller an Informationen ran. Meine PC-Kenntnisse sind besser geworden.
- Mein Englisch-Vokabelschatz hat sich ein bisschen erweitert. Ich komme mit dem Computer besser klar. Ich kann die Leistungskraft eines Computers voll ausschöpfen.
- Komplizierter Denken; Allgemeinwissen verstärkt; kann man nicht in anderen Fächern gebrauchen.
- Durch ab und zu vergleichbare Denkschritte erleichtert sich das Fach Mathematik (natürlich nicht immer).
- Gar nichts. Vielleicht nur, dass ich mehr Spass an diesen Rechnern habe, aber deshalb habe ich Informatik ja auch gewählt, weil ich mehr wissen wollte.
- Man lernt effektiver Schreiben und man kann durch das Internet mehr Informationen für bestimmte Themen finden.
- Es hat sich eigentlich überhaupt nichts verändert.
- Leichteres Benutzen des Computers. Mehr Wissen über Computer, was in anderen Fächern angewendet werden kann.
- Bessere Kenntnisse und Vorstellungen darüber.
- Ich kann mir Sachen aus dem Internet holen, um Infomaterial zu bekommen - Texte abtippen für Referate ist einfacher geworden (nicht mehr per Hand).
- Für viele andere Fächer ist das Internet sehr hilfreich geworden. Ich kann mir Informationen herausuchen, die für ein Unterrichtsthema bzw. Referat sehr nützlich sind.

4. **Sichten auf die Informatik. Bitte ordnen Sie die folgenden drei Sichten auf die Informatik zeitlich.**

Die zugehörige Abbildung 4 findet sich auf S. 172.

**Ergebnisse:** alle richtig 6 (33 %); einen (und damit automatisch zwei) Fehler/ ein Treffer 7 (39 %); alle drei falsch 5 (28 %) — **zufällige Auswahl: 16,7 %; 50 %; 33 %**

<sup>368</sup> zwei Antworten lauteten: „nicht“, rsp. „gar nichts“

5. Bitte nehmen Sie Stellung zu den folgenden Behauptungen  
(je Aussage nur *ein* Kreuz):

	eingetroffen	möglich	ja $\Sigma$	unmöglich/falsch
In wenigen Jahren wird es das <b>papierlose Büro</b> geben.	1	14	15	3
Das <b>Haus</b> alarmiert die Polizei, weil jemand einbrechen will.	13	3	16	2
In einem <b>Auto</b> befinden sich ca. 70 Informatiksysteme.	5	8	13	5
Informatiksysteme stellen eine <b>Konkurrenz für die Intelligenz</b> der Menschen dar.	6	6	12	7
<b>Schachcomputer schlagen jeden</b> menschlichen Spielpartner.	4	7	11	7
Die Kapazität von <b>fünf Computern reicht</b> für alle Berechnungen auf der ganzen Welt völlig aus.	1	9	10	8

Weitere Ergebnisse zu Item 3:

**Für mein persönliches Lernen (dies gilt vor allem für andere Fächer) hat sich durch die Informatische Bildung folgendes verändert:**

- Strukturierteres Denken und Arbeiten; neue Interessengebiete.
- Andere Sicht des Faches Englisch; wichtig für die Informatik, da viele Bezeichnungen in der englischen Sprache gegeben sind.
- Mehr Arbeit am Computer [insbesondere für die Facharbeit der 12].
- Mehr Computerkenntnisse; das Internet kann nun benutzt werden, um Informationen zu beschaffen.
- Besseres Englisch; Mitreden unter Freunden.
- Bessere Kenntnisse über die Informatik; hilft wenig im Zusammenhang mit der Mathematik.
- Entdeckung neuer Seiten an dem eigenen PC; dadurch Schaffung von Vorteilen in der heutigen Zeit, aufgrund der Fähigkeit mit dem Computer umzugehen.
- Mehr Computerkenntnisse (braucht man in vielen Bereichen des Lebens).
- Mehr Ahnung von aktuellen Themen und erleichtertes Umgehen mit anderen PC's und dem Internet.
- Das Strukturieren des Lernens für andere Fächer ist besser geworden.
- Eigentlich nicht viel ( bin noch nicht lange im Kurs ), aber ich kann jetzt ein paar Dinge programmieren mehr aber auch nicht.
- Verbesserung der Nutzung eines Computers beim Schreiben eines Referats bzw. einer Facharbeit.

- Ich ärgere mich schneller über nicht geschaffte Aufgaben und schlechte Noten.
- Grundsätzlich hat sich nichts geändert. Die Informatische Bildung lässt sich höchstens im Fach Latein (Lernprogramme) einsetzen.
- Bessere Kenntnisse über die Informatik.
- Hilft wenig im Zusammenhang mit der Mathematik.
- Mehr Arbeit am Computer (insbesondere für die Facharbeit der 12.).
- Mehr Computerkenntnisse; das Internet kann nun benutzt werden, um Informationen zu beschaffen.
- Besseres Englisch; Mitreden unter Freunden.
- Ich sehe das Fach Englisch anders. Es ist wichtig für Informatik, da viele Bezeichnungen in der engl. Sprache gegeben sind.
- Bis jetzt noch nichts Gravierendes.
- Strukturierteres Denken und Arbeiten; Neue Interessengebiete.



## Anhang D

# Gruppeninterview

### D.1 Leitfaden

Um Interviews mit Schülerinnen zu führen, wird hiermit eine Reihe von Fragegegenständen aufgelistet, die Bestandteil einer Gesprächssituation zwischen einer Vertreterin der Didaktik der Informatik der Universität Dortmund und Schülerinnen des 11er Informatikgrundkurses sein sollten. Mit einer schriftlichen Befragung wurden bereits vorher grundlegende Rahmenbedingungen erfragt. Dort konnten keine weitergehenden inhaltsbezogenen Punkte aufgearbeitet werden. Diese sollen durch die Anwendung weiterer Methoden untersucht werden. In einem der schriftlich befragten Informatikkurse wird unter Zuhilfenahme von Gruppenbefragung, Gruppendiskussion und Gruppeninterview (vgl. [Bortz und Döring 1995, S, 221, 222, 293–296]) der Versuch unternommen, die Lerngruppe bezüglich des Fragenkomplexes weiter zu untersuchen.<sup>369</sup> In der Konkretion wurden Hinweise an die Interviewerin gegeben, die hier dokumentiert sind.

---

**Welche Sprachen hast Du im Informatikunterricht kennengelernt?**

**Warum hast Du Informatik gewählt?**

**Bringt das informatische Wissen auch für sprachliche Fächer Vorteile?**

Bei der Beantwortung der Frage zu „Sichten auf die Informatik“, zu dem die Schülerinnen die Aufgabe hatten, drei Sichten in eine zeitlich Reihenfolge zu bringen, wurde deutlich, dass sie – ausgehend von einer eher intuitiven Vorstellung – ein Bild der Informatik entwickelt haben, dieses aber von einer zufälligen Auswahl bezüglich der zeitlichen Anordnung nicht signifikant abweicht.

**Dieses Bild soll in dem Gespräch aus der Sicht der Schülerinnen deutlicher konturiert werden.**

Die folgenden disjunkten Elemente aus den drei Grafiken von GENRICH, PETRI und BRAUER können Gesprächsanlässe darstellen:

**übermächtige Intelligenz (G), Denkverstärker (P), rechtlich selbstständige Person (G), aktiver Assistent für die Arbeit (B), Unterstützung des individuellen und des kollektiven Gedächtnisses (B)**<sup>370</sup>

Stellungnahmen zu Behauptungen – hier können die Thesen von Hermann MAURER eingesetzt werden. Die *Thesen 17, 18 und 19* aus [Maurer 2000, S. 52f] eine gute Grundlage für ein Gespräch liefern. Vor allen die Erläuterungen zu These 18 und These 19 stellen klar heraus, wie die weitere Entwicklung zur Zeit eingeschätzt werden kann. Die Thesen 10 bis 14 können für weitere „Behauptungen“ und damit für

---

<sup>369</sup> mit Mitteln der qualitativen empirischen Sozialforschung

<sup>370</sup> Die in Klammern angegebenen Punkte nehmen Bezug auf die Autoren der jeweiligen Veröffentlichung:

G  $\cong$  GENRICH

P  $\cong$  PETRI

B  $\cong$  BRAUER

Diskussions- und/oder Gesprächsanlässe herangezogen werden. Sehr provokativ und klar: die Thesen 27 bis 31 [Maurer 2000, S. 54], die „in den Raum geworfen werden“ können, wenn die vorherigen Thesen keine Diskussion provozieren.

Weiterer Fragekomplex:

eine einfache Modellierung: Mikrofon – Mischpult – Aufnahmegerät

## D.2 Transkriptausschnitte

00:00:00

1 Interviewerin: Was sagt die Technik. Wir sind so weit. Gut, dann würde ich sagen, fangen wir an, dann  
2 kann jetzt auch aufgenommen werden. Ja, die erste Frage, die an Sie stellen möchte, ist warum Sie das  
3 Fach Informatik gewählt haben und da würde ich ganz kurz einfach von jedem seine persönliche Meinung  
4 zu haben. . . .

5 Schüler 1: Ja, als ich auf die Schule gekommen bin, ich hatte das Fach noch nie; und mein Beratungslehrer  
6 hat mir geraten, das Fach zu wählen, für den späteren Beruf wäre das ganz nützlich, aber für meinen  
7 Beruf, den ich anstrebe, ist das überhaupt nicht nützlich, für Krankenpfleger braucht man eigentlich nicht  
8 irgendwie Computerwissen oder irgendwie so und ich habe es eigentlich nur genommen, weil mir mein  
9 Beratungslehrer dazu geraten hat. . . .

10 Schüler 2: Ja, warum habe ich da Fach Informatik gewählt. Ja, vielleicht, weil Computer Zukunft ist, ein  
11 bisschen mehr darüber Informationen bekommen. . . Nein, also, ich habe es einfach gewählt, weil ich es  
12 interessant finde, ich wusste gar nichts vorher, ich wollte es kennen lernen. . . .

13 Schüler 3: . . . ich hatte mal Praktikum an der Berufsschule als Informationstechnischer Assistent gemacht,  
14 und ich fand das eigentlich sehr interessant da und habe mir gedacht, wenn ich später was in die Richtung  
15 machen will, dann wäre es ganz praktisch, wenn ich schon mal das Fach Informatik hatte, und dann habe  
16 ich das Fach genommen. . . .

17 Schüler 4: Ich hab das eigentlich nur genommen, weil das alle genommen haben und weil mir eigentlich  
18 keine andere Wahl blieb, so wie ich meine anderen Fächer gewählt hatte – dann blieb eigentlich nur noch  
19 Informatik über, was ich hätte wählen können.

20 Schüler 5: Ich aus drei Gründen – erster Grund war . . . Mathe Leistungskurs ist gekoppelt an Informatik  
21 und ich wollte mir die Möglichkeit offen halten, Mathe Leistungskurs zu nehmen – zweiter ist, die sich  
22 aber als völlige Blamage rausgestellt hat, ich wollte lernen, richtig mit Computern umzugehen, aber da wir  
23 überall nur Mac haben und keine Computer, keinen richtigen PC, ist das meiste eigentlich ziemlich sinnlos.  
24 Ja und drittens, das ich vielleicht einmal aus dieser Abhängigkeit von Computern rauskommen kann. . . .

25 Schüler 5: Ja, ich habe mir gedacht, du möchtest gerne etwas über dieses Fach erfahren und so, ich habe  
26 gedacht, das wäre sehr interessant – momentan bin ich nicht der Meinung – und ich hoffe, das ich da  
27 irgendwann mal etwas draus gelernt habe.

28 Schüler 6: Ja, ich dachte auch, das man was draus lernen kann und ich habe das mal gewählt um raus zu  
29 finden, wie das so ist.

30 Schüler 7: Ich hatte vorher zwei Jahre Informatik gehabt, . . . ja . . . das ist ein bisschen anders als vorher,  
31 ja, das war's eigentlich auch schon.

32 Schüler 8: Ja, ich hatte vorher auch kein Informatik und bei den Wahlen hat mein Beratungslehrer mir das  
33 empfohlen, weil angeblich Computer und so, weil das in jedem Beruf ist und so und darum habe ich das  
34 auch gewählt.

35 Schüler: [Gelächter]

36 Interviewerin: Ja, dann erstmal danke schön, für diese ersten Informationen. Es klingen ja jetzt schon  
37 unterschiedlichen Meinungen über das heraus, was man so von Informatik erwartet, welche persönlichen  
38 Vorstellungen man damit für sie verbindet. Wenn sie jetzt mal in sich hinein horchen, was ist denn für

39 sie Informatik. Wie würden sie Informatik sehen, wie würden sie es definieren, wie würden Sie vor allen  
40 Dingen überlegen, wie beispielsweise Ihre Eltern Informatik sehen, was ist für die Informatik. Wenn sie  
41 da einfach versuchen, das einmal so ein bisschen so darzutun, und vielleicht sogar, wie ein Informatiker  
42 möglicherweise Informatik sieht. Gibt es da vielleicht Unterschiede zwischen dem... Vielleicht einfach  
43 mal zu diesem Themenkomplex einfach mal zwei drei Worte

44 Schüler 1: Ja ich weiß nicht, ich hoffe, ich verstehe Sie jetzt richtig, Informatik für meine Eltern, vielleicht  
45 für meine Mutter und meinen Vater bedeutet das überhaupt nichts, mit Computern haben die gar nichts am  
46 Hut, wenn die Inf hören, dann denken die eh nur Computer und rechnen und tippen halt, aber ich weiß  
47 nicht, für meinen Vater, der braucht den Computer für den Beruf und aus meinem ... jetzt aus dem hohlen  
48 Bauch heraus würde ich sagen Informatik ist langweilig. Wenn ich das Fach so höre, ist langweilig, du sitzt  
49 vor dem Computer, und guckst dir das an, du tippst ein bisschen, und ja ...

50 Interviewerin: Und wenn sie mal versuchen würden, Informatik so ein bisschen zu definieren, was würden  
51 sie dazu sagen ...

52 Schüler 2: Also ich kann an der Stelle auch nur Beispiele dazu bringen, also Informatik, wir haben zuhause  
53 einen PC, und wenn ich mal Probleme damit habe, dann frage ich meine Vater: Ja Papa, kannst du mir da  
54 mal helfen. Da sagt er nur zu mir: Wieso, du hast doch Informatik. Ich sag: Ja, da machen wir aber was  
55 ganz anderes, das hilft mir hierbei nicht. So gesehen kann ich da zu Informatik auch nichts sagen.

56 Interviewerin: Ja, aber versuchen Sie doch mal eine Definition für den Begriff Informatik aus Ihrer bishe-  
57 rigen Erfahrung so.

58 Schüler 2: Aus meiner bisherigen Erfahrung. Ja am Anfang haben wir gezeigt, was der Bildschirm und so  
59 etwas ist, und jetzt machen wir sofort objektorientierte ... , ich kann dazu voll nichts sagen, ich habe es ja  
60 auch erst ein halbes Jahr. ...

61 Schüler 3: Ich von klein auf mit Computern aufgewachsen, wir hatten immer mindestens zwei Stück zu-  
62 hause, weil meine Mutter die für die Arbeit immer gebraucht hat, und mit Computern, bei uns kennt sich  
63 eigentlich jeder damit aus, nur ich mittlerweile besser als meine Mutter, weil die ganze Technik und so hat  
64 sie nicht so im Griff. Weiß nicht, eigentlich normal alles.

65 Interviewerin: Gut, dann sind Sie aber doch der Fachmann und können mir eine Definition für den Begriff  
66 Informatik geben.

67 Schüler 3: Ich weiß nicht, also, wie ich das hier so merke, geht das nur um irgendwelche Programme,  
68 Programme erstellen, und irgendwelche Strukturen am PC erschaffen und wie das ganze überhaupt läuft  
69 und so. ...

70 Schüler 4: Ich weiß nicht, auf eine Art könnte man Informatik als eine Erleichterung in beruflicher Hinsicht  
71 sehen, aber was mir dazu einfällt, ich kann auch nur sagen, ich finde, was wir jetzt lernen, da nützt uns später  
72 überhaupt nichts, weil die dann wieder so weit sein werden und dann können wir mit dem, was wir hier  
73 gelernt haben, überhaupt nichts mehr anfangen, so sehe ich das einfach.

74 Interviewerin: Können Sie das ein bisschen erläutern

75 Schüler 4: Ja ich weiß nicht, jeden Tag werden, würde ich so spinnereimäßig sagen, bessere Computer  
76 erfunden, vielleicht muss man irgendwann nur noch, ... was man gerade denkt, der macht das dann sofort,  
77 führt das dann sofort aus, dann braucht man nicht dann erst stundenlang irgendwas eingeben, bevor das  
78 überhaupt alles läuft, und ... weiß ich nicht, ich finde es einfach doof. ...

79 Schüler 5: ... Informatik ist so was, das gehört auch dazu, wenn man ein Problem hat, und man löst das  
80 Problem mit Hilfe von Computern. Also, Informatik hat immer was mit Computern zu tun, immer was mit  
81 ständigem Fortschritt, so dass man gar nicht mehr überblicken kann, und, wie sie schon gesagt hat, das  
82 ist so ein richtiger Fortschritt und man kann sich dem gar nicht mehr richtig anpassen, früher da gab es  
83 irgendwie Schelllack, dann gab es die Schallplatte

84 Ein Schüler: C 64

85 Schüler 5: immer alles schön langsam hintereinander, dann irgendwann Kasette und Cd, aber jetzt kommt  
86 alles irgendwie so direkt hintereinander, da blickt man hinterher gar nicht mehr richtig durch. Vor allem,

87 ich finde, Informatik ist zu schnell, Informatik ist Schnelligkeit, Informatik ist Fortschritt, aber, ... das ist  
88 nicht unbedingt nützlich. ...

89 Schüler 5: Ich würde ich dem ganzen anschließen, weil ich weiß nicht, was ich da noch ergänzen soll. ...

90 Schüler 8: Wenn ich ehrlich bin, ich habe Informatik gewählt, damit ich weiß, wie ich so einen Lebenslauf,  
91 so eine Tabelle, weil ich so alles schreiben kann, elektronische Datenverarbeitung oder so, weil ich wollte  
92 früher Industriekaufmann werden und da war ich so auf einem Informationsnachmittag, da hat der was  
93 von EDV gelabert, und darum habe ich auch Informatik gewählt, damit ich später nicht auf die Nase falle.  
94 Und das wir hier eigentlich gar nicht so beigebracht, weil früher, da hatte ich gar nicht Informatik und auf  
95 einmal soll ich solche Programme erschaffen, wie soll das denn gehen, ja und Definition von Informatik  
96 da fällt mir nichts ein, weil, wie gesagt, das wächst halt immer, Pentium I, II, III und das wird auch immer  
97 ständig wachsen, und das ist viel zu schnell für uns, das ist, sage ich mal, etwas unkontrolliertes ...

98 Interviewerin: ... Nächste Frage wäre, Welche Sprachen haben sie im Informatik-Unterricht kennen gelernt  
99 ... aber denken sie doch mal mehr an Informatik-spezifische Sprachen, die sie da ja wahrscheinlich auch  
100 kennen gelernt haben. Was fällt ihnen da denn noch zu ein

101 verschiedene Schüler: die Dokumentenbeschreibungssprache – HTML, Python, und diese ganzen.

102 Schüler 6: aber so richtig kennen gelernt haben wir die eigentlich nicht.

00:11:12 103 Schüler 5: Ja doch, HTML und Python ...

104 Interviewerin: Ja, da wir jetzt beim Sprachen sind, denken, dass das informatisches Wissen auch was für  
105 die sprachliche Fächer, wie Englisch, Deutsch, Französisch usw. gebracht hat ... Haben Sie das Gefühl,  
106 dass Ihnen das Informatikunterricht überhaupt irgend etwas gebracht hat.

107 Schüler 4: Ja, beachten von Groß- und Kleinschreiben, Wenn diese nicht beachtet, dann gibt es ERROR .

108 Schüler 5: Das einzige was das Schreibeerei gebracht hat, ich kann jetzt schneller schreiben. Aber sonst für  
109 andere Fächer hat der Informatikunterricht nichts gebracht, es ist sinnlos

110 Schüler 5: Also Informatik braucht man gar nicht für andere Fächer

111 Schüler 6: Information aus dem Internet holen, den ich finde das ist sehr wichtig.

112 Schüler 4: Früher hat man das auch ohne Computer gekonnt

113 Schüler 7: Da war das aber viel aufwendiger. Bücherei, Buch ausleihen, lesen, das ist das falsche, geh  
114 nochmal hin, hol mir ein neues

115 Schüler 5: Unmündigkeit, in Philosophie haben wir das doch gemacht mit Unmündigkeit. Wir gehen weiter  
116 in die Unmündigkeit rein.

117 Interviewerin: Wenn Sie jetzt noch mal versuchen, direkt an die sprachlichen Fächer zu denken, also jetzt  
118 an Deutsch und Englisch, würden Sie da irgendwo noch einen persönlichen oder würde irgendwer von  
119 Ihnen noch einen persönlichen Nutzen für diese Fächer also für sich ganz persönlich erkennen

120 Schüler 2: Also sag ich mal so. Unser Lehrer meinte, dass man durch Informatik lernt ordentlicher zu sein.  
121 Aber ich für mich persönlich habe das nicht festgestellt. Das man das dann ordentlicher aufschreibt, weil  
122 in der Informatik, wenn man das nicht genauso schreibt, wie man es schreiben soll, dann funktioniert es  
123 halt nicht und auf die Ordnung bezogen, oder so, man findet alles schneller wieder, aber so, weiß ich nicht.

124 Interviewerin: Wie ist das bei Ihnen

125 Schüler 6: Ich weiß nicht, ich sehe da irgendwie ...

126 Schüler 1: keinen Nutzen

127 Schüler 6: genau

128 Interviewerin: Gut, also ich mein

129 Schüler 6: Ne ist nicht, ich weiß vielleicht wie andere Fächer gut sind für die Informatik, wie Schüler 4  
130 gesagt hat, Englisch, wenn da irgendein Fehler ist, oder so, dann brabbelt mir da der Computer einen auf

- 00:13:43 131 Englisch vor, und da brauch ich vielleicht das Fach Englisch für. Das ist aber andersrum jetzt so, aber sonst,  
132 ... ich weiß nicht wofür Informatik für die anderen Fächer gut sein soll. Keine Ahnung.
- 133 Interviewerin: Wir kommen jetzt auf den von Ihnen schon erwähnten Begriff objektorientierte Modellie-  
134 rung. Ich möchte Sie bitten, einige der Begriffe, die in diesem Zusammenhang aufgetreten sind, zu nennen  
135 und zu versuchen diese zu definieren. Nennen Sie einfach die Begriffe die Ihnen in diesem Zusammenhang  
136 einfallen und was Sie mit diesem Begriffen verbinden – Ich meine nicht Computer. Ich meine konkrete  
137 Begriffe zum objektorientierten Modellieren und – Problemlösen.
- 138 Schüler 2: Erweiterung ...
- 139 Interviewerin: Sammeln wir mal einfach ein paar Begriffe, die Sie gehört haben
- 140 Schüler: Klassen, Buntstift, Struktur, Licht, Blinklicht, Stift, Tisch, Bildschirm ...
- 141 Interviewerin: Sie haben jetzt zwei unterschiedlichen Kategorien von Begriffen genannt. Eine sind kon-  
142 krete Begriffe, konkrete Ausprägungen von Klassen und zum anderen haben Sie schon beispielsweise den  
143 abstrakten Begriff der Klasse genannt. Gibt es noch weitere Begriffe, die Ihnen in dieser Richtung einfallen
- 144 Schüler 5: Ja, das sind die Erweiterungen von Klassen, z. B. die Klasse Stift und die Klasse Buntstift. Der  
145 Stift kann zeichnen allerdings nur schwarz und der Buntstift kann auch andere Farben und so was ...
- 146 Interviewerin: Na gut, als Ihr das Blinklicht besprochen habt, bin ich auch hier bei Euch im Unterricht  
147 gewesen. Und wie haben Sie das Blinklicht dazu gebracht zu blinken - das Blinklicht blinkt ja auch nicht  
148 von alleine, was gab es da denn noch.
- 149 Schüler 1: Schleifen.
- 150 Interviewerin: Ja, so was gab es auch. Das hat jedoch mit dem objektorientiertem Konzept nicht so viel zu  
151 tun. Schleifen gehören zu den Kontrollstrukturen, die Sie auch gelernt haben, das ist richtig. Ja, aber wie  
152 bringt man überhaupt ein Blinklicht dazu zu blinken
- 153 Wir haben jetzt zwei wichtige Begriffe genannt, die Begriffe Klasse und Methode. Versuchen Sie mal, die  
154 beiden Begriffe zu definieren mit Ihren eigenen Worten – Versuchen Sie es einfach mal
- 155 Schüler 2: Ja, Klassen sind fertig modellierte, benutzbare Strukturen, Gegenstände, wie auch immer, also,  
156 Objekte, die man so benutzen kann. Sie sind so erstellt, man benutzt sie, um weitere Klassen zu erstellen.  
157 Und Methoden sind halt Methoden womit man die Klassen zum Laufen bringt, Warte oder so. ...
- 158 Interviewerin: ... nehmen wir die Begriffe Mikrophon, Mischpult und Aufnahmegerät, übrigens heute sehr  
159 passend. Versuchen Sie einmal, die auf ähnliche Art und Weise darzustellen, wie Sie das mit dem Licht  
160 und dem Blinklicht gemacht haben
- 161 Schüler 5: Oberbegriff ist Musik oder Gesang, Töne danach kommt das Mikrophon, wo man das rein  
162 spricht, danach kann das Mischpult kommen und danach das Aufnahmegerät, aber es kann auch umgekehrt  
163 sein, denn man kann es erst mischen dann aufnehmen oder erst aufnehmen und dann mischen. Man kann  
164 das jedenfalls tabellenförmig machen. ...
- 165 Interviewerin: Versuchen Sie jetzt bitte doch einmal mit Ihren eigenen Worten zu beschreiben, wie Sie beim  
166 Licht und Blinklicht vorgegangen sind, als Sie versucht haben, dieses Problem zu lösen ... von Anfang an.  
167 ...
- 168 Schüler 4: Ich weiß nicht, ich habe erst mal geschaut was wir vorher alles gemacht haben, also diese  
169 einzelnen Sachen, was man auch verstanden hat und dann hat man sich das irgend wie zusammen gebastelt.  
170 Zusammengesammelt von den Nachbarn und irgendwann hat es ja funktioniert.
- 171 Interviewerin: Gut, das ist auch eine typische Methode der Informatik – Ja bitte
- 172 Schüler 8: Am Anfang war das ja so, dass wir nur einen Strich zeichnen sollten und das war irgend was  
173 mit Import und so. Benötigten Sachen importieren und dann angeben und das war es und dann sollten wir  
174 von so einen einfachen Strich sofort einen Tisch zeichnen, dann kam noch ein bißchen Mathematik dazu,  
175 wieviel Grad und so, die Länge. Und jetzt auf eine Ampel und das alles wurde kompliziert und ich habe  
176 nichts mehr aufgeschrieben. Das vergißt man doch schnell wieder.

177 Schüler 6: Vielleicht weiß ich das nächste Woche noch, aber die Woche darauf habe ich dann das Ganze  
178 vergessen.

179 Schüler 8: Das Problem ist ja, wenn man einmal gefehlt hat, dass die anderen das Thema schon abgeschlos-  
180 sen haben und man hat fast keine Möglichkeit mehr, das nachzuarbeiten, weil es ... kein Buch gibt.

181 Interviewerin: Ja gut, das ist durchaus schwierig, das sehe ich ein. Ich muß noch mal auf eine Sache  
182 zurückkommen. Versuchen Sie noch mal mit Ihren eigenen Worten zu beschreiben oder zu erklären, warum  
183 Sie beim Beispiel mit Licht und dem Blinklicht die Notation an der Tafel hatten, diese Bilder mit den  
184 Vierecken und mit den Pfeilen dazwischen. Was ist denn da passiert. Was sollte denn dadurch überhaupt  
185 dargestellt werden. ...

186 Schüler 8: Also, was ich dazu noch sagen kann, Blinklicht ist die Erweiterung von der Klasse Licht. Und  
187 um das Blinklicht zu erschaffen, mußten wir das Licht importieren und dann mußten wir noch ein paar  
188 kleinere andere Sachen importieren und dann noch ein paar Sachen und dann war das Blinklicht fertig.

23:24 189 Interviewerin: Gut, ich würde sagen wir lassen es in diesem Themenbereich gut sein.

190 Interviewerin: Kommen wir zum letzten Themenkomplex. Es gibt so ein paar Thesen, so zur Informatik  
191 und da würde ich einfach mal so Ihre persönliche Meinung zu haben wollen und zwar, das erste wäre –  
192 Ohne Computer wird man sich im Jahre 2010 nackter fühlen als ohne Kleidung, sagte jemand 1989. Was  
193 meinen Sie dazu.

194 Schüler 1: Wird man sich nackter fühlen als ohne Kleidung

195 Schüler: [Gelächter]

196 Schüler 6: Ich kann mir das nicht vorstellen.

197 Schüler 3: Ja doch

198 Schüler 5: mittlerweile schon

199 Schüler 4: Es kann doch keiner mehr etwas selbstständig

200 Schüler 5: Mittlerweile ja, wir sind von den Dingen abhängig

201 Schüler 4: Wie er schon gesagt hat, warum in die Bücherei rennen, wenn man da mal eben Knöpfchen  
202 drücken kann.

203 Schüler: [allgemeine Zustimmung]

204 Schüler 1: Man holt sich alles aus dem Internet, wer schlägt noch ein Buch auf, wer geht noch zur Bücherei,  
205 geht zum Regal, zieht das Buch raus, ist noch am Suchen, wenn man einfach nur eintippen braucht, und  
206 der rattert das alles dann runter. gibt sofort den Begriff ein ...

207 Schüler 7: man spart Zeit

208 Schüler 5: Sie haben doch auch den Film Matrix gesehen, bestimmt, nehme ich mal an, als Informatiker,  
209 da war das ja so, die haben da diesen Stopfen reingekriegt und dann konnte der Kung Fu. Irgendwann wird  
210 es vielleicht auch so sein. Das ist schlecht, schlecht, dämlich, dumm, ich finde das nicht gut.

211 Schüler 6: Schlecht ist das nicht

212 Schüler 5: Doch

213 Schüler 8: Also ich würde sagen, auch wenn man im Jahr 4000 oder so ist, das Informatik nicht lebens-  
214 wichtig ist, sondern das Leben nur ein bisschen erleichtert. Nicht ein bisschen, sondern viel erleichtert,  
215 aber es wird nie lebenswichtig sein.

216 Schüler 5: Ist es doch jetzt schon – was wäre ein Atomkraftwerk ohne Computer.

217 Schüler 8: Ja ich meine jetzt auf die Gesellschaft selber, z. B. mit den Büchern, ich kann ja zur Bibliothek  
218 laufen, und nach dem Buch suchen, und das so und so machen, aber ich könnte ja auch zum Computer,  
219 also das Computer macht mir das Leben ja nur leichter. ...

- 220 Schüler 5: Ja, was ist wenn es irgendwann gar keine Büchereien mehr gibt
- 221 Schüler 4: Ja eben . . .
- 222 Schüler 3: Ich denke mir das ganz anders, von wegen, ehm, die nächste Zukunft, also die nächste Generation  
223 der Leute, die werden das ganz anders sehen, wir wissen jetzt noch wie das geht, aber wenn man sich  
224 überlegt, man versucht immer alles so einfach wie möglich zu machen. und es wird uns ja auch immer  
225 beigebracht, der Mensch ist von Natur aus faul. Warum soll man dann nicht den leichten Weg gehen.
- 226 Schüler 5: Papa, was ist ein Buch.
- 227 Schüler 4: Die ganzen Kinder werden sofort in der Grundschule an den Computer gesetzt, so seh ich das  
228 dann, und dann gibt's ja nichts mehr, wirklich so: Papa, was ist ein Buch.
- 229 Schüler 2: Stell dir mal eine Zukunft vor, wo man nicht mehr arbeiten braucht, wo das die ganzen Computer  
230 und Roboter dann schon tun. So in ferner Zukunft
- 231 Schüler 5: Hast du Simsons geguckt, wo Homer zuhause war....
- 232 Interviewerin: OK
- 233 Schüler: [Gelächter]
- 234 Schüler 2: Jaja
- 235 Schüler 5: . . . fernsehen
- 236 Interviewerin: Waren ja schon viele Meinungen von Ihnen. Zweite Sache wäre – Das Tragen von Hardware  
237 wird Mode
- 238 Schüler 7: Das Tragen von was.
- 239 Interviewerin: Das Tragen von Hardware wird Mode
- 240 Schüler 3: Einen Chip mit mir rumtragen..
- 241 Schüler 1: Oder du hast 'nen Computer in der Jacke oder so
- 242 Schüler 5: Das hat man doch jetzt schon, Handys und so was
- 243 Schüler 8: Ja dann hast du alles voller Disketten hängen.
- 244 Schüler 5: Überall ist das doch drin, Communicator oder Handys und so
- 245 Schüler 6: Ja es gibt ja auch schon Uhren, da kann man Geld drauf speichern kann man dann damit bezahlen  
246 oder so, so ähnlich wie so ein Ski Pass oder so etwas . . . und Handys kriegst du oder Laptops, das ist ja im  
247 Grunde nichts anderes, oder diese, wie heißen die, diese kleinen Notiz. . .
- 248 Schüler 5: Notebooks
- 249 Schüler 6: . . . genau Notebooks, oder so, hat doch auch schon fast jeder
- 250 Interviewerin: Aha
- 251 Schüler 8: Ist im Walkman auch so was drin
- 252 Schüler: Gelächter
- 253 Schüler 4: Das ist noch etwas Gutes, Altes
- 254 Schüler 2: Haha
- 255 Schüler 1: Da musste noch kurbeln
- 256 Interviewerin: Kommt auf den Walkman an, würde ich sagen. Der Computer als gesondertes Einzelobjekt  
257 verschwindet. Things that think werden ihn ersetzen
- 258 Schüler 6: Also doch englischer Kram

- 259 Schüler 5: Dinge, die denken
- 260 Schüler 4: Könnte passieren ...
- 261 Schüler 3: Gucken Sie doch mal, es gibt doch jetzt World Wide Web, Netze, kleine Server, das ist ja nicht  
262 mehr ein Computer nur, der geht ja an viele andere. Das Problem ist dann dabei wieder, wenn man auf den  
263 Server zugreifen kann, kann man auch auf die anderen zugreifen. Und dann, kommt drauf an, wer davor  
264 sitzt, dann ist der halt das Ding das denkt ... Irgendwann machen die Computer was für sich alleine, und  
265 bestimmen uns und nicht wir sie
- 266 Schüler 6: Vielleicht ist ja irgendwann so weit, dass künstliche Intelligenz in dem Sinne uns überwuchert  
267 oder so – quasi wie so ein Virus oder so – wenn man das intergalaktisch sieht.
- 268 Schüler 8: Die Computer werde eh irgendwann die Welt beherrschen und uns auslöschen.
- 269 Schüler: [Gelächter und Stimmengewirr]
- 270 Interviewerin: Naja, das wollen wir ja nicht hoffen.
- 271 Schüler 5: Dann machen wir Sie dafür verantwortlich
- 272 Schüler 4: Ja
- 273 Interviewerin: Also mich persönlich bitte nicht, vielleicht meinen Berufsstand
- 274 Schüler 5: Sind Sie ein guter Informatiker oder ein böser
- 275 Interviewerin: Ok. Dann bedanke ich mich recht herzlich ...
- 276 Schüler 1: Wir haben zu danken
- 277 Interviewerin: ... dass Sie so gut mitgearbeitet haben und, ja, so gut Sie das halt jetzt wissen konnten und  
278 so gut Sie jetzt durch Ihren Unterricht darauf vorbereitet waren. Also, ich denke, ich denke, dass das für  
279 uns auf jeden Fall sehr aufschlussreich ist und dass wir das auch entsprechend auswerten werde und es  
280 kann sein, dass wir durchaus noch einmal wiederkommen und noch mal horchen.
- 281 Schüler 5: Wie stellen Sie sich denn die Zukunft mit Computern vor, ich meine Sie arbeiten ja jetzt an  
282 dieser Zukunft mit, quasi
- 283 Interviewerin: Wie stell ich mir die Zukunft mit Computern vor.
- 284 Schüler 4: Geld, Geld, Geld
- 285 Interviewerin: Nein, Geld nicht unbedingt, also ich denke auf jeden Fall nicht, dass es irgendwann mal  
286 dahin kommen wird, dass es so etwas wie eine allmächtige künstliche Intelligenz geben wird. Ganz einfach,  
287 weil es Probleme gibt, die einfach so komplex sind, dass sie ein Computer einfach prinzipiell nicht lösen  
288 kann.
- 289 Schüler: Noch nicht
- 290 Interviewerin: Nicht nur auch noch nicht. Es gibt Probleme, die sind einfach so komplex und so schwierig,  
291 das es also auch, wenn ein Computer noch beliebig schneller wird, das es irgendwann immer Grenzen  
292 geben wird, dass es Probleme geben wird, die ein Computer prinzipiell nicht lösen kann.
- 293 Schüler 3: Aber das kann man doch jetzt noch gar nicht sagen.
- 294 Interviewerin: Doch, dass kann man sagen – das kann man auf Grund von theoretischen Überlegungen  
295 sagen, ich meine es gibt ja einen Komplex der Informatik, das ist die theoretische Informatik, die sich  
296 mit nichts anderem befasst, als mit der prinzipiellen Lösbarkeit von Problemen; ob man die überhaupt  
297 einmal prinzipiell wird mit Rechnern lösen können. Und da gibt es eben Problemklassen, die so schwierig  
298 sind, dass also selbst der schnellste Computer, selbst wenn er die gesamte Dauer des Universums rechnen  
299 würde, er würde diese Probleme nicht lösen können. Und selbst, wenn man davon ausgeht, dass Computer  
300 irgendwann noch viel, viel schneller werden, wird es irgendwann Probleme geben, die prinzipiell nicht  
301 lösbar sein werden; und deswegen wird es zumindest keine allmächtigen Computer geben. Aber sie werden  
302 uns sicherlich sehr stark beherrschen, unser Leben noch mehr prägen und noch mehr beherrschen, als es  
303 jetzt der Fall ist.
- 00:29:47 304 Schüler 8: Also, Sie sagen ja gerade, künstliche Intelligenz, das ist ausgeschlossen.

## Anhang E

# Hospitationsprotokolle zur Evaluation

Datum	Thema
Montag, 19. Februar 2001	Ordnerbäume – objektorientierte Beschreibung
Dienstag, 20. Februar 2001	Textverarbeitung – objektorientiert betrachtet
Mittwoch, 21. Februar 2001	E-Mail – objektorientierte Beschreibung
Donnerstag, 22. Februar 2001	Verweisstruktur – objektorientiert beschreiben und modellieren
Freitag, 23. Februar 2001	E-Mail – Aufbau der Adresse, Transport im Netz

Tabelle E.1: Hospitationen im Bayerischen Schulversuch – Themenübersicht

### E.1 Ordnerbäume – objektorientierte Beschreibung

#### Informatisches Fachkonzept

- Ordnerbäume:
  - Ordner als Dateikollektion/Container/Baum ohne Blätter
  - Operationen/Methoden: Erschaffen, Öffnen, Ausschneiden, Einfügen, Löschen (intuitiv)
  - Rechner werden freigegeben und das Lernmaterial wird durch die Schülerinnen „abgeholt“
  - Kanten im Baum als „Enthalten sein“

Problem: Organisation von Daten/Dateien mit Hilfe von Ordnerstrukturen

- Aspekte: Personenstruktur und Anwendungsfelder

Bewertung der Ergebnisse der Strukturierung unter verschiedenen Aspekten:

- Ordner für mehrere Personen von Bedeutung (Informatik)
- Verwechslungsgefahr: Mathematik und Mathematikaufgabe

Entwicklung von Kriterien: Übersichtlichkeit, Effizienz im Zugriff (Pfadlänge), Rechte der Eigentümer (intuitiv)  
 Kriterium: nicht zu viele Ordner auf einer Ebene vs. nicht zu viele Ebenen (Pfadlänge)  
 Eindeutigkeit der Bezeichnungen (Problem Namensgleichheit)  
 Pfad als Muster bestehend aus Laufwerksbezeichnung, Ordnerstruktur „entlang des Weges“  
 Wiederholung: Methode und Attribut (woran erkenne ich, ob es sich um einen Ordner handelt)  
 Größe als temporäres Attribut, Pfad als bewußt organisiertes Attribut  
 Ordnerbaum organisieren, optimieren / anpassen / ändern  
 Archiv als Beispiel für spezielle Anforderungen: gemeinsam nutzen und Synchronisierungsprobleme bei Aktualisierung / Schreibschutz, Datenintegrität

### Schülerinnen- und Schüleraktivität und -verhalten

Schülerinnen erarbeiteten die Aufgabenstellung „Baumstruktur entwickeln“ in Partner- und Einzelarbeit.  
 Die entworfenen Lösungsvarianten der Schülerinnen waren sehr umfangreich (mehrere Ebenen) Aufgabenblatt 3 a).  
 Diskussion zur Bewertung und Verbesserung der vorgeschlagenen Ordnerstrukturen mit begleitender Demonstration.  
 Aufgabenblatt 3 b) Problemstellung  
 Umsetzung erarbeiteter Lösungsvorschläge in der praktischen Arbeit mit den Informatiksystemen, am Beispiel des Verschiebens zweier Teilbäume innerhalb einer Ordnerstruktur;  
 Knoten „Fächer“ entfernen, um den Baum nach dem Kriterium „Anzahl zu öffnender Ordner minimieren“ (Benutzungsfreundlichkeit)  
 Systematische Zusammenfassung:

- Erarbeitete Theorie- und Praxiselemente in Form von „Merksätzen“ zum Löschen eines Ordners mit „wervollen Inhalten“ zusammenfassend formulieren.

### Fachdidaktische Konzepte

Motivation und Problemstellungsphase:  
 Beispiel Dokumente und Dateien einer Familie,  
 selbstständige Entwurfsphase der Schülerinnen,

fragend-entwickelndes Gespräch zur Bewertung der Ergebnisse  
 parallel vorgeschlagene Ordnerstrukturen mit begleitender Demonstration  
 kognitives Problem: äquivalente Darstellungsformen Dateiverwaltungssystem und Ordnerbaum  
 Erarbeitung der Vorgehensweise zum Löschen eines Ordners unter Erhalt wertvoller Komponenten  
 Erarbeitung eines Ordners mit Gruppenzugriff und Datenkonsistenz (Archiv)

Sicherungsphase: selbstständiges Implementieren der verbesserten Ordnerstruktur,  
 Zusammenfassung zum Löschen eines Ordners  
 Stellung der Hausaufgabe

**Medieneinsatz**

16 Schülerarbeitsplätze Win 98,  
Server WinNT, Verteilen der Ordnerstruktur über das Netz  
MasterEye:

- Anzeigen von Dokumenten/Verzeichnisstrukturen und
- Demonstration (unterrichtsgesprächsbegleitend)

Beamer  
Arbeitsblätter  
Tafelbild  
Betriebssystemfunktionen: Dateiverwaltung

**E.2 Textverarbeitung – objektorientiert betrachtet****Informatisches Fachkonzept**

- Textverarbeitung:
  - Klassen, Dokument, Absatz, Zeichen, Fußzeile, Kopfzeile, Rahmen, Grafik, Tabelle, Zelle
- Klassenbeziehung enthalten sein (Aggregation)
- Zähligkeit: Ein Absatz kann viele Zeichen enthalten (Multiplizität)
- Ordner
- Inspektion von Ordnern und Dateien (ansatzweise) durch ein Dateiverwaltungssystem
- In Vektorgrafik und Textverarbeitung erzeugte Dokumente wurden in Dateien gespeichert. Die Dateien befinden sich in einem Ordner.

Dateien und Ordner:

- Dateien
  - Speichern gleicher Dokumente unter unterschiedlichen Dateinamen
  - Speichern eines geänderten Dokumentes unter einem bereits vorhandenen Dateinamen
  - Möglichkeiten des Dateischutzes:
    - \* Sicherheitskopie, Schreibschutz
  - Attribut einer Datei und Veränderung des Attributwerts: Schreibschutz,
  - Methode Speichern eines Dokumentes

Betriebssystemdateien: wurde implizit und informell und auf Schülererfahrung aufbauend eingeführt  
Datei enthält ein Dokument, d. h. eine Datei ist ein Behälter (Container) für ein Dokument.

neue Klasse: Datei

alle vom Dateiverwaltungssystem angezeigten Attribute der Klasse Datei

Pfad: Laufwerk, Backslash, Festplatte

- Festplatte kann auch in mehrere Laufwerke unterteilt werden
- Verzeichnis versus Ordner

Archiv: wurde bisher nicht eingeführt, Möglichkeiten zur Verdeutlichung der indirekten Attributveränderung (durch Sicherung)

Attribut versteckt

Aktualisieren einer schreibgeschützten Datei

Am Suffix, d. h. an der Dateierweiterung erkennt man eine Klassifizierung der Dateien, die betriebs-systemspezifisch ist. Beispiel zur Wiederaufnahme: Hypertext-Dokumenten (Speicherung mit Suffix HTM, verbreitet: html)

Methode Umbenennen einer Datei

### Schülerinnen- und Schüleraktivität und -verhalten

fragend-entwickelndes Unterrichtsgespräch zur Wiederholung (Klasse, ...)

Erarbeitung des neuen Stoffes mit praktischer Schülertätigkeit (mit Hilfe eines eigenen Dokumentes): gleiches Dokument unter einem anderen Dateinamen speichern

Änderung des Dokumentes und unter einem anderen Dateinamen speichern

Realisierung der Schutzmöglichkeit und Erkunden der Konsequenzen

Unterrichtsgespräch zum Herausarbeiten der neuen Klasse Datei mit Attributen und Methoden

Wechsel zwischen den Phasen Erarbeitung, Zusammenfassung, Präsentation und Sicherung (Schülerinnenaufzeichnung)

### Fachdidaktisches Konzept

Wiederholung des Datei- und Dokumentenbegriffs der Textverarbeitung zur Motivation der tieferen Betrachtung der Klasse Datei mit ihren Attributen und Methoden

Erarbeiten wesentlicher Attribute und Methoden anhand von klar definierten Aufträgen, die den Schülerinnen die Nützlichkeit ihrer früheren Erfahrungen zeigen.

Im Wechsel wurde sowohl experimentell neue Erarbeitung durchgeführt, zusammenfassend im Gespräch verdeutlicht, gesichert (durch Präsentation) und von den Schülerinnen in ihre Aufzeichnungen übernommen

Eine Lehrerdemonstration zur Verdeutlichung der Änderung: geändertes Dokument unter gleichem Dateinamen, Gefahr des Verlustes wichtiger Inhalte, Schutzmöglichkeiten, Diskussion und Sicherung durch Schülereigentätigkeit

Lehrer berät die Schülerinnen und kann individuell sowohl Hilfestellung bei der konkreten Umsetzung geben, wie auch beratend die Arbeit bei Schwierigkeiten unterstützend zur Seite stehen.

Attribute einer Datei: Verzahnung zwischen Arbeitsblatt, Projektion (Beamer) und Erkundungsaufträgen für Attribute am Schülereinzelpplatz

Klassifikation von Dateien über Suffixe

Arbeitsblatt ermöglicht die Weiterbearbeitung in Form einer fakultativen Hausaufgabe.

**Medieneinsatz**

lokales Netz (BNC) mit Windows 95 - Klienten, Lehrerrechner Windows 98  
Beamer unter der Decke (verbunden mit dem Lehrerrechner)  
Whiteboard (wurde nicht genutzt)  
Arbeitsblatt  
von den Schülerinnen bereits in früheren Unterrichtsstunden erstellte Dokumente wurden als Lernmaterialien für die Aufträge genutzt

**E.3 E-Mail – objektorientierte Beschreibung****Informatisches Fachkonzept**

Objektorientierte Beschreibungen

**Wiederholung:**

Klassenkonzept mit Attributen am Beispiel E-Mail: To, Cc, Bcc  
informeller Zugang zum Methodenbegriff: Beantworten (Reply)

**Methodenbegriff:**

Erarbeitung und Formulierung (inkl. Parameter)

Entwicklung und Vertiefung fachsprachlicher Elemente

Anbindung an die Klasse NACHRICHT mit ihren bekannten Attributen

**Methoden der Klassen NACHRICHT**

- weiterleiten (Empfänger)
- antworten () – Erweiterung auf ReplyToAll (mit Erklärung)
- löschen()
- senden (Empfänger)
- ... Hinweis auf weitere Möglichkeiten

Der Weg einer elektronischen Nachricht (Routing)

Intranet der Schule (mit Login), lokaler Mailserver

Netiquette und Authentifizierung (altersgerechte Gestaltung)

Internetanbindung (nächster Mailserver)

Empfängerseite (PC der Lehrerin und zugeordneter Mailserver)

Mailaustausch/-verbindungsrechner/server (nicht thematisiert wurde: SMTP)

Darstellung des Headers einer Mail, um zu zeigen, dass dieMail tatsächlich das Intranet verlassen hat, zudem Angaben über den zurückgelegten Weg umfasst (z. B. Identifizieren von Servernamen)

**Schülerinnen- und Schüleraktivität und -verhalten**

Den Beobachtern wurde deutlich, dass die Schülerinnen mit den Anforderungen produktiv, handelnd und erfolgreich umgehen konnten. Es gelang, die abstrakten Konzepte mit konkreten Erfahrungen zu untersetzen. Sowohl die altersgemäße Sprechweise, wie auch die Verbindung zur Fachsprache wurde erfolgreich in den Unterricht eingebracht. Es wurde deutlich, dass die Schülerinnen und Schüler motiviert sind, sich sowohl mit der inhaltlichen (E-Mail), wie auch mit der fachlichen Seite des Unterrichtsgegenstandes konstruktiv auseinanderzusetzen.

Aus einer früheren Befragung ist bekannt, dass alle Schülerinnen zuhause einen Zugang zu Computer haben. Allerdings haben sie keinen Internetzugang.

Zu den einzelnen unterrichtlichen Arbeitsformen  
experimentelle Arbeiten

- Weiterleitung elektronischer Nachrichten in Gruppen (Impuls: Lehrermail an alle)
  - Adressbuchnutzung
- Erzeugen einer Mail an der Hausarbeitsplatz der Lehrerin

gelenktes Unterrichtsgespräch

- Wiederholung Klassenkonzept und Attribute
- Systematisierung der Methoden
- Formalisierung der Transportmechanismen
  - Vorschlag einer Schülerin: universeller Mailserver für die ganze Welt
  - Schülerfrage: Ist es möglich, einen Verbindungsrechner aufzurufen?

Übertragen der Unterrichtsergebnisse ins Heft

### Fachdidaktische Konzepte

#### **Einordnung in die Reihe**

Objektorientierte Beschreibungen

#### **Einheit zu E-Mail**

0. Handling

1. Attribute

#### **2. Methoden und praktische Erprobung**

experimentelle, differenzierte Gruppenarbeit (durch die Beantwortung einer Mail werden unterschiedliche Stränge für verschiedene Schüler entwickelt - Wortspiel) mit Hilfe der Weiterleitung (Fw)  
Ausweisen der Methoden durch zusammenfassende Darstellung im Unterrichtsgespräch

#### **3. Architekturkonzept E-Mail-Transport**

Ausgangspunkt: E-Mail-Versand und -Empfang im lokalen Netz

Schülerinnen beschreiben, dass es auch E-Mail außerhalb der Schule gibt

Unterscheidung und Verbindung verschiedener Komponenten: PC, Mail-Server, Verbindungsrechner/-server

Unterscheidung zwischen Datenübertragung über Mail-Server (Netzkomponenten) und dem Nutzungsverhalten in der Telekommunikation

Durch Analyse der Lehrerinnenadresse wurde von den Schülerinnen vermutet, dass gewisse Netzstrukturen, -komponenten notwendig sind, um eine Nachricht erfolgreich auszuliefern.

### Medieneinsatz

#### **Ausgangssituation:**

Informatikraum – 16 Schülerarbeitsplätze und 1 Lehrerarbeitsplatz (mit Scanner)

2 Drucker (ein Laserdrucker für die Schülerarbeit, ein Tintenstrahldrucker nur für den Lehrerarbeitsplatz)  
lokales Netz (Novell – Version 4.x) mit Anschluss ans Internet,  
lokaler Server – INES (Version 4.x)  
Beamer unter der Decke - mit Fernbedienung  
Whiteboard, häufig wird über den Beamer projiziert, damit alle Schülerinnen es sehen können  
**Einsatz in der beobachteten Unterrichtssituation:**

- lokales Netz
- Beamerprojektion (die für die Darstellung des externen Mailempfangs unabdingbar ist)
- Internetverbindung
- Tafelbild

## E.4 Verweisstruktur – objektorientiert beschreiben und modellieren

### Informatisches Fachkonzept

Übergang von traditionellen Verweisstrukturen zur informatischen Verweisstruktur in Hypertexten  
Abbilden einer erkannten Verweisstruktur, die objektorientiert beschrieben und modelliert wurde

- Bezug zur Textverarbeitung und zur Dateiorganisation (Baumstruktur, Datei-Ordnerbaumstruktur) aktive Änderung von Attributwerten
- Pfadbegriff (relativ vs. absolut)

Hypertextdokumente und damit verbundenes Dateiattribut

Nachrichtenkonzept, Kaskade von Nachrichten

Revidierung der vom Verweis-Objekt ausgelösten Attributwertänderung des Zeichens

- Objektorientierte Beschreibung des Prozesses Datei öffnen, ändern, speichern

aktive Verweise – u. a. durch Verweisaktivierung werden Dateien geschlossen und andere geöffnet

Änderung auf der Benutzungsoberfläche

ausgelöste Aktion

#### **Browsermodus vs. Entwurfsmodus**

Unterschiede zwischen zwei Sichten auf ein Dokument

Dokumentenstruktur – ist verknüpft mit, m : n Beziehung (Multiplizität) – neuer Grad der Abstraktion

Objektbeziehungen – verweist auf, Multiplizität (Analogieschluss bezogen auf Textmodellierung)

Schüler als Navigierende vs. Schüler als Gestaltende

### Schülerinnen- und Schüleraktivität und -verhalten

Vergleich traditioneller Verweise mit informatischen Verweisstrukturen am konkreten Beispiel (Lexikon), Aufbau einer angemessenen Verweisstruktur, Abbildung in objektorientierte Beschreibungen (5 Dokumente mit 5 Verweisen) – siehe AB 1

Stillarbeit mit anschließendem Vergleich der Ergebnisse

aktive Arbeit mit den Schulinformaticsystemen:

- OS
- StarOffice
  - Öffnen von existierenden Dokumenten,
  - Einfügen von Verweisen und Abspeichern
    - \* Einige Schüler allerdings gingen den Weg der Dokumentenerstellung und Einfügen von Verweisen und Abspeichern
- IE
  - Öffnen eines Dokuments mit Verweisen
  - Aktivieren eines Verweise (und Feststellen der zugehörigen Aktionen)

Vertiefung zum Modellierungsprozess  
fragend-entwickelndes Unterrichtsgespräch  
Arbeitsblatt

### Fachdidaktische Konzepte

Lehrerin: Motivation durch traditionelles Lexikon

Unterrichtsgespräch mit Wiederholung:

- a) Informatiksysteme (Ordner, Textverarbeitung . . .)
- b) objektorientiertes Modellieren

Arbeit an den neuen Inhalten:

Verknüpfungsstruktur der Dokumente mit Verweisen (Haftbilder an der Tafel, AB 1)

Demonstration durch die Lehrerin, Gestaltung eines Verweises, Neugestaltung brachte einige Schülerinnen im nachfolgenden Arbeitsprozess trotz anderer Arbeitsanweisung auf die Idee, ebenfalls ein neues Dokument zu erstellen

praktische Arbeit der Schülerinnen

Alle Schülerinnen hatten ausreichend Zeit, sich mit den gestellten vielschichtigen Arbeitsaufträgen angemessen auseinanderzusetzen.

In diesem Prozess wurden sie von der Lehrerin beraten, unterstützt - nach und nach auch durch die Mitschülerinnen, die die Aufgaben bereits erfolgreich bearbeitet hatten.

Die erfolgreiche, zielgerichtete Arbeit der Schülerinnen ist bemerkenswert (Unterrichtszeit beachten, Komplexität und Abstraktionsanforderungen).

Fragend-entwickelndes Unterrichtsgespräch zur Vertiefung:

- a) es zeigt sich der Erfolg des durchgehend objektorientierten Ansatzes
- b) Erkennung Objektbeziehungen und der darüberliegenden Abstraktionsebene -

- Prozessverunschärfung (Unscharfes Bild nach Abnehmen der Brille durch die Lehrerin)
- Verdecken der enthaltenen Objekte (Umdrehen der Haftbilder)

c) Entwurfsmodus

Nachrichtenaustausch und -weiterleitung (neue)

Rückbindung an die zu Beginn der Einführung (Demonstration) erkannte automatische Attributänderung

Nachrichtenaustausch (auslösende Aktion Benutzerin) aktivieren, Datei öffnen und Datei schließen wurde von den Schülerinnen als verbales Erklärungsmuster deutlich gemacht

### Medieneinsatz

hardwaremäßige Videovernetzung (Übertragung des Lehrerbildschirms auf die Schülerbildschirme)  
schulisches Intranet (Fileserver/Printserver - Novell Version 4.x) und Kommunikationsserver (Linux)  
pro Arbeitsplatz steht ein dediziertes Verzeichnis zur Verfügung  
(Schule verfügt über zwei fahrbare Beamer-/Rechnereinheiten)  
Applikationen: StarOffice 5.2 zur Dokumentenerstellung, IE zur Anzeige  
Whiteboard: Haftobjekte (1. Arbeitsblatt)  
Kinderlexikon (mehrbändig, Brockhaus)  
OHP:

- Auszüge aus dem Lexikon
- Benutzerhefteintrag
- Arbeitsblatt zur Vertiefung

Arbeitsblätter (Objektdiagramm Lexikon)  
Klasse Verweis

## E.5 E-Mail – Aufbau der Adresse, Transport im Netz

### Informatisches Fachkonzept

Wiederholung/Voraussetzungen:  
elektronische Nachricht (Kopf, Header) ist analysiert:  
To, Cc, Subject, Body  
An, ... [deutschsprachige Analogie]

Erster E-Mail-Versand bereits durchgeführt

Neu:

Aufbau einer E-Mail-Adresse:

account@domain

benutzername@Domäne

Domäne: Mail-Server/Server/Namensauflösung (DNS)

Weg einer E-Mail durchs Internet

Verbindungsrechner

verschiedene Mailserver (Sender-Empfänger)

verschiedene Varianten des Umgangs mit E-Mail-Abruf (IMAP, POP)

Vorteil/Nachteil

Analogieschluß: Gelbe Post – Elektronische Post

(und umgekehrt)

Praktisches Umsetzen: E-Mail-Erstellen und Versand

### Schülerinnen- und Schüleraktivität und -verhalten

Gruppenarbeiten in vorher zufällig zusammengestellten Gruppen  
 verschiedene Gruppenaufträge (arbeitsteiliges Vorgehen) über Arbeitsblatt (für alle)  
 Vorstellen der Gruppenlösungen durch Schülervorträge - es wurde ruhig und ausführlich erklärt  
 Ergebnissicherung  
 Raumwechsel (→ Informatiklabor)  
 Schülerinnen müssen sich anmelden, um ihre gruppen- und arbeitsplatzbezogene Accounts zu nutzen:  
 Arbeitsauftrag 1: E-Mail abholen  
 diese enthält weitere Arbeitsaufträge ...  
 Bearbeiten der Arbeitsaufträge (inkl. Senden der E-Mail mit Cc an die Lehrerin)  
 Systematisierung und Zusammenfassung:  
 GA-Ergebnisse wurden wieder aufgenommen und in dem Zusammenhang mit der praktischen Tätigkeit reflektierender Bewertung und zusammenfassender Darstellung unterzogen

### Fachdidaktische Konzepte

Motivation für die Gruppenarbeit (GA)  
 GA  
 Vorstellung der Gruppenergebnisse durch die Schülerinnen  
 Sicherung  
 Raumwechsel  
 praktische Arbeit mit E-Mail im Intranet

1. Befragung bzgl. der Evaluation des Informatikunterrichts
2. ...

Unterrichtsgespräch: Systematisierung  
 Projektion (Beamer) Schülermithilfe mit Rückbezug zur Eingangsproblemstellung

### Medieneinsatz

16 vernetzte Schülerarbeitsplätze (Win 98), Lehrerarbeitsplatz Win 98, mehrere Linux-Server (Intranet-Mail)  
 Beamer (fahrbar)  
 OHP  
 Arbeitsblatt  
 Folien (Gruppenarbeit)  
 E-Mail der Lehrerin an alle Schülerinnen (Arbeitsaufträge)  
 Raum 1 war für die Gruppenarbeit vorbereitet: Raumwechsel

## Anhang F

# Exemplarische Unterrichtseinheiten

Bei den im folgenden dokumentierten Unterrichtseinheiten handelt es sich um erprobte Modelle zur Einführung der Informatik in Grundkursen in der gymnasialen Oberstufe. Im Laufe der Zeit wurden vom Autor verschiedene Einstiegsvarianten für den Informatikunterricht in der Sekundarstufe II entwickelt und schulisch umgesetzt. Die Basis für die Beschreibungen sind die eigenen unterrichtlichen Erfahrungen, Erfahrungen in der Lehrerbildung (erste und zweite Phase) und in Lehrerfortbildungen.

### F.1 Informatik in der Sekundarstufe II: Automatentheorie<sup>371</sup>

Die Vorbereitung des Informatikunterrichts für die Sekundarstufe II führte zu einem Stoffverteilungsplan (siehe Tabelle F.1, Seite 198), in dem deutlich wird, wie Elemente der theoretischen Informatik bereits auf Anfängerniveau für den Unterricht in der Sekundarstufe II zur Planungsgrundlage gemacht werden können.

Für Schülerinnen sind Automaten integraler Bestandteil des Alltags (Getränkeautomaten, Fahrkartenautomaten, ...). Alle Automaten lassen das EVA-Prinzip deutlich werden: Die Eingabeobjekte sind als Aufzählungstyp in Modula-2 oder Pascal darstellbar (in Oberon muss hier auf die Ordnungszahl abgehoben werden, da es dort keine selbstdefinierten Typen gibt). Das gleiche gilt für die möglichen Ausgaben (einzige Hürde: leere, d. h. keine Ausgabe). Dann wird der Blick auf die „innere Struktur“ gelenkt: Automaten sind von Schülerinnen abstrakt analysierbar. Sie erkennen, dass auch ohne Kenntnis der dem Automaten „innewohnenden Technik“ das „Verhalten“ des Automaten i. d. R. deterministisch ist. Sie sind in der Lage, dieses „Verhalten“ graphisch darzustellen: die Übergangs„funktion“<sup>372</sup> wird als gerichteter Graph entwickelt. Dieser Graph lässt sich in zwei Prozeduren umsetzen. Eine Prozedur ermittelt ausgehend von einem aktuellen (inneren) Zustand des Automaten und einer Eingabe (mittels geschachtelter, doppelter Fallunterscheidung) den neuen (inneren) Zustand des Automaten. Entsprechend lässt sich die zu einem Zustand und einer Eingabe „gehörende“ Ausgabe ermitteln. Da die Ein- und Ausgabe selbstdefinierter Typen den Rahmen des Anfängerunterrichts unnötig belasten würde, ist dieser Teil in einen eigenen Modul ausgelagert worden (für Modula-2 und Oberon); bei Pascal würde eine Unit diesen Zweck erfüllen. Die Schülerinnen benutzen also für die komplette Ein- und Ausgabe (bzw. IO) eine minimal definierte Schnittstelle Automatio.

<sup>371</sup> Das Skript der Vorlesung von Giegerich (vgl. [Giegerich 1989, S. 3f]) lieferte den Impuls zur didaktischen Gestaltung für den Informatikanfangsunterricht in der Sekundarstufe II.

<sup>372</sup> in "" eingeschlossen, da der Begriff in der verwendeten Programmiersprache für „vorbelastete Schülerinnen“ die Verwendung einer Funktionsprozedur nahelegen würde, besser aber - weil der Zustand in dieser Prozedur geändert werden muss - eine Prozedur mit einem Wertparameter (Eingabe) und einem Referenzparameter (nämlich Zustand) verwendet werden sollte. Dies sei im Unterschied zu [Bruhn u. a. 1988, Seite 88ff] angemerkt; die Darstellung bei [Modrow 1998, S. 12ff] führt m. E. zu nichtssagenden Bezeichnungen - in beiden Büchern lassen sich allerdings für Lehrerinnen weitere Beispiele für einfache Automaten, deren Realisierung im Unterrichtszusammenhang von Interesse sein könnte, finden.

## **0 Vorbemerkungen**

Begriffsklärung: Informatik

## **1 Endliche Automaten**

### **1.1 Simulation von Automaten mit Ausgabe (Transduktoren)**

- Der Müsliriegelautomat
  - das EVA-Prinzip
  - Festlegung der Aufzählungstypen EINGABE und AUSGABE
  - Graphische Veranschaulichung der Übergänge des Automaten
    - \* Automatengraph
  - Beschreibung der Übergänge mit Hilfe einer Automatentafel
    - \* Automatentabelle
  - Umsetzung der Automatentafel oder des Automatengraphen in zwei Struktogramme AusgabeFunktion und Übergang
  - Darstellung der beiden Struktogramme als Prozeduren
    - \* PROCEDURE AusgabeFunktion und PROCEDURE Uebergang
  - Entwicklung des Moduls MuesliriegelAutomat
    - \* die Ein-/Ausgabeschnittstelle AutomataIO
  - Eingabe und Austesten des Moduls  
Voraussetzung dazu: kurze Einführung in die grundlegenden Arbeitsweisen mit dem Modula-2 System
- Der Getränkeautomat (gleiche Vorgehensweise wie vor)
- Entwicklung und Codierung eigener Automaten (mit mindestens drei verschiedenen Eingabemöglichkeiten und mindestens drei Zuständen)

### **1.2 Es wird einfacher: erkennende Automaten ohne Ausgabe (Akzeptoren)**

Erkennen eines Schlüssels

### **1.3 Klassifizieren zulässiger Eingaben**

## **2 Zusammenhänge zwischen syntaktischen Strukturen und Automaten**

### **2.1 Was können Akzeptoren nicht leisten?**

### **2.2 Akzeptoren und zugehörige syntaktische Strukturen (links- bzw. rechtsreguläre Grammatiken)**

Umwandlung von ... in ... und vice versa

### **2.3 Kellerautomaten - Erkennen von Klammerstrukturen**

- deterministische, kontextfreie Grammatiken
- der allgemeine PDA

**Algorithm 1** Schnittstelle AutomatIO

DEFINITION MODULE AutomatIO;

PROCEDURE Initialisierung(EingabeTitel, EingabeStr, AusgabeStr, ZustandsStr: ARRAY OF CHAR);

(\* EingabeTitel: &gt;&gt;Titel&lt;&lt; für das Eingabemenü;

EingabeStr: enthält alle möglichen Eingabestrings, jeweils durch | getrennt;

AusgabeStr: enthält alle möglichen Ausgabestrings, jeweils durch | getrennt;

ZustandsStr: enthält alle möglichen Zustandsstrings, jeweils durch | getrennt;

\*)

PROCEDURE MenuEingabe(VAR EingabeNR: CARDINAL);

(\* Nicht die Eingabe aus EINGABE, sondern die Nummer wird hier zurückgegeben. \*)

PROCEDURE ZustandAnzeigen(ZustandsNR: CARDINAL);

(\* Nicht der Zustand aus ZUSTAND, sondern die Nummer ist hier anzugeben! \*)

PROCEDURE AusgabeAnzeigen(AusgabeNR: CARDINAL);

(\* Nicht die Ausgabe aus AUSGABE, sondern die Nummer ist hier anzugeben! \*)

END AutomatIO.

**Die Unterrichtssequenz: „Von Null zur Realisierung einfacher Automaten“**

Begriffsklärung: Informatik ist ein Kunstwort, das aus *Information* + *Automatik* gebildet werden kann.

Ausgehend von dem vorläufigen Begriff wird das Augenmerk auf Automaten gelegt. Den Schülerinnen soll über Automaten aus ihrem Umfeld die Modellierung mit Hilfe der Vorstellung, dass einem Automaten eine unzugängliche „Mimik“ innewohnt, es aber trotzdem möglich ist, Automaten zu bedienen. Dazu gehört offenbar ein „Bild von der Funktionsweise“ eines Automaten, damit die Benutzerinnen voraussehen können, was durch die Bedienung ausgelöst wird, um dem Ziel (z. B. eine Fahrkarte zu erhalten) näherzukommen.

Bei den folgenden Beschreibungen beziehe ich mich auf die Implementierung in Modula-2; das Folgende ist jedoch auch in andere dem imperativen Paradigma zuzuordnenden Sprachen, wie Oberon oder die verschiedenen Ausprägungen von Pascal umsetzbar. Auch die Umsetzung für das funktionale Paradigma sollte ohne größere Probleme möglich sein.

**Zur Notation:** In Modula-2 und in Oberon werden im Gegensatz zu Pascal Groß- und Kleinbuchstaben unterschieden, d.h. EINGABE und Eingabe sind zwei verschiedene Bezeichner. Im Folgenden gilt (für selbstdefinierte Objekte): Typen werden komplett großgeschrieben; Elemente eines Typs beginnen mit einem Kleinbuchstaben; Variablen, Prozeduren und Modulbezeichner beginnen mit einem Großbuchstaben.<sup>373</sup>

**Experimente mit Automaten**

Um nach und nach einen „richtigen“ Automaten zu entwickeln, wird als erstes den Schülerinnen die folgende Problemstellung vorgelegt:

**Problemstellung:**

Es soll die Funktionsweise eines Müsliriegelautomaten einfachster Bauart nachgebaut werden. Ein Riegel soll 1,50 DM kosten. Als Eingabe werden nur 50 Pfg.-Stücke zugelassen. Es gibt keine Tasten – weder für die Ware noch für die Rückgabe.

Bei der Bearbeitung dieser ersten Problemstellung müssen die wesentlichen Elemente für die Umsetzung der Problemstellung durch die Lehrerin angegeben werden.

**Hinweis:**

Bei der Erweiterung der Problemstellung zum Getränkeautomaten sollte die Erarbeitung wesentlich von den Schülerinnen getragen werden. Alle nötigen Sprachelemente werden als Formalismen im Problemlösungszusammenhang eingeführt.

Begriffe, die (richtlinienkonform) an dieser Stelle eingeführt werden:

<sup>373</sup> Alle in Typewriter smallest gesetzten Zeichenfolgen sind (mögliche) Konstrukte in der Programmiersprache Modula-2

Eingabe	Input	E
Innere Zustände des Automaten und die Übergänge von einem Zustand zu einem anderen		V
Ausgabe	Output	A

Tabelle F.2: Das EVA-Prinzip im unterrichtlichen Kontext der Realisierung von Automaten

In diesem Zusammenhang werden gemäß der üblichen (unterstrichfreien) Syntax für Bezeichner (identifier) die zur Ein- und Ausgabe gehörenden Aufzählungstypen zusammengestellt (keine Mengen):

```
TYPE EINGABE = (pf50);
```

```
AUSGABE = (ware, nichts);
```

Zugleich wird darauf hingewiesen, dass zu jedem Element eines selbstdefinierten Typs eine Ordnungsnummer gehört

```
0..MAX(EINGABE) und 0..MAX(AUSGABE).
```

Die Standardfunktionen `ORD` (um die Ordnungsnummer zu erhalten) und

`VAL` (um den zu der Ordnungsnummer gehörenden Wert zu gewinnen) werden vorgestellt und zur Übung an weiteren Beispielen eingeübt.

Als nächstes wird der Zusammenhang zwischen Eingabe und Ausgabe mit Hilfe eines aus einfachsten graphischen Elementen bestehenden gerichteten Graphen erarbeitet: *Ovale* bezeichnen *interne Zustände des Automaten*; *Pfeile* mit jeweiliger Eingabe bezeichnen den *Übergang von einem zum nächsten internen Zustand des Automaten*; **E**: das Folgende wird eingegeben; **A**: das Folgende wird ausgegeben.

Der Graph stellt abstrakt die interne Funktionsweise des Automaten dar. Dabei wird nicht auf die technische Realisierung im tatsächlichen Automaten abgehoben - soll doch auch „nur“ seine Funktion beschrieben werden. Bei der Entwicklung des gerichteten Graphen werden parallel die einzelnen Elemente des Typs `ZUSTAND` benannt:

```
ZUSTAND = (nichtsEingeworfen, eingeworfen50, eingeworfen100);
```

Die graphische Darstellung kann leicht überprüft werden. Diese graphische Darstellung ist dann in eine Automatentafel umzuwandeln. Die so entstandene Tabelle F.3, Seite 200 hat den Vorteil, dass die Vollstän-

Zustand: <b>ZUSTAND</b>	Eingabe: <b>EINGABE</b>	Ausgabe: <b>AUSGABE</b>	Zustand: <b>ZUSTAND</b>
nichtsEingeworfen	pf50	nichts	eingeworfen50
eingeworfen50	pf50	nichts	eingeworfen100
eingeworfen100	pf50	ware	nichtsEingeworfen

Tabelle F.3: Automatentafel für MuesliAutomat

digkeit aller  $(\text{MAX}(\text{ZUSTAND})+1) * (\text{MAX}(\text{EINGABE})+1)$  Möglichkeiten leichter überschaut werden kann (vor allem, wenn es sowohl eingabeseitig, wie auch bei der Anzahl der Zustände „viele“ Möglichkeiten gibt). Zum anderen ist es methodisch sinnvoll, mit den Schülerinnen verschiedene Darstellungsformen bereits frühzeitig zu entwickeln. Anschließend wird die Tabelle in die beiden Prozeduren `AusgabeFunktion` und `Uebergang` umgesetzt.

Diese Umsetzung kann sehr anschaulich schrittweise vorgenommen werden.

1. Transformation des Graphen (bzw. der Tabelle) in Struktogramme ( [Norm DIN 66 261 ] nach dem Vorschlag [Nassi und Shneiderman 1973]).

**AusgabeFunktion (Eingabe: EINGABE; Zustand: ZUSTAND): AUSGABE**  
— Prozedur/Funktion mit Rückgabe vom Typ AUSGABE

CASE Zustand OF		
nichtsEingeworfen	eingeworfen50	eingeworfen100
RETURN nichts	RETURN nichts	RETURN Ware

Abbildung F.1: MuesliAutomat – Ausgabefunktion als Struktogramm

**Uebergang (Eingabe: EINGABE; VAR Zustand: ZUSTAND)**  
— Prozedur mit Referenzparameter vom Typ ZUSTAND

CASE Zustand OF		
nichtsEingeworfen	eingeworfen50	eingeworfen100
Zustand:= eingeworfen50	Zustand:= eingeworfen100	Zustand:= nichtsEingeworfen

Abbildung F.2: MuesliAutomat – Zustandsübergänge als Struktogramm

2. Umsetzung der Struktogramme in die entsprechenden Prozeduren. Dabei ist die Umsetzung mit Hilfe einer geschachtelten Fallunterscheidung möglich. Die nötigen Sprachelemente, also die Prozedurdeklaration und die Fallunterscheidung, werden wie oben bereits angemerkt, als Formalismus eingeführt. Realisierung der Prozeduren `AusgabeFunktion` und `Uebergang` siehe Algorithmus 2, Seite 202. Diese beiden Prozeduren (in Pascal: Funktion bzw. Prozedur) stellen das „Herz“ des Automaten dar<sup>374</sup>.

Was fehlt noch? Bisher wurde nur angemerkt, dass die Ein- und Ausgabe in einen eigenen Modul ausgelagert ist. Damit dieser Modul für alle möglichen Automaten arbeiten kann und die zu einem konkreten Automaten gehörenden Zeichenketten vorhanden sind, muss die Routine `AutomatIO.Initialisierung` aufgerufen werden. Damit ist das, was auf den Eingabetasten des Automaten steht, eine Beschreibung dessen, was ausgabeseitig geschieht sowie eine Beschreibung, die auf den aktuellen internen Zustand des Automaten schließen lässt, an den Modul `AutomatIO` übergeben worden. Über die Ordnungsnummer können diese Zeichenketten anschließend für die Ein- und Ausgabe benutzt werden. Für das Schreiben des Hauptprogramms müssen einige Überlegungen vorgeschaltet werden, die in der *verbalen Beschreibung* der Struktur des Hauptprogramms ihren Ausdruck finden (siehe Algorithmus 3, Seite 202).

<sup>374</sup> Die Arbeitsweise des Automaten wird damit in zwei Phasen zerlegt: Ermittlung der Ausgabe; Ermittlung des neuen Zustands

**Algorithm 2** Prozeduren AusgabeFunktion und Uebergang

---

```

PROCEDURE AusgabeFunktion (Eingabe: EINGABE; Zustand: ZUSTAND): AUSGABE;
BEGIN
  CASE Zustand OF
    nichtsEingeworfen: RETURN nichts
    | eingeworfen50: RETURN nichts
    | eingeworfen100: RETURN ware
  END (* CASE *)
END AusgabeFunktion;

PROCEDURE Uebergang(Eingabe: EINGABE; VAR Zustand: ZUSTAND);
BEGIN
  CASE Zustand OF
    nichtsEingeworfen: Zustand:= eingeworfen50
    | eingeworfen50: Zustand:= eingeworfen100
    | eingeworfen100: Zustand:= nichtsEingeworfen
  END (* CASE *)
END Uebergang;

```

---

**Algorithm 3** Verbale Beschreibung: MuesliriegelAutomat**Anfang** MuesliriegelAutomat

Initialisierung der Zeichenketten für die Eingabemaske, für die Eingabezeichenketten, für die Ausgabezeichenketten und für die Zustandsbeschreibungen;

Festlegung des Urzustands des Automaten;

**Wiederholung**

Eingabe anfordern; Ausgabe ermitteln und anzeigen;

Zustandsübergang ausführen und anzeigen

**bis falsch**

Ende MuesliriegelAutomat.

---

Die für den Automaten nötigen Typen `EINGABE`, `AUSGABE` und `ZUSTAND` sind bereits oben formal exakt deklariert worden. Die Konkretisierung der oben verbal formulierten Initialisierung führt zu der folgenden Anweisung:

```

Initialisierung (
  "Müsliriegelautomat Eingabe:",
  "50 Pfg",
  "Müsliriegel | leer",
  "nichts eingeworfen | 50 Pfg eingeworfen | 1,- DM eingeworfen");

```

Spätestens an dieser Stelle muss deutlich herausgearbeitet werden, dass es einen aktuellen Zustand des Automaten gibt, und ein Objekt, das diesen festhält; außerdem muss die Ordnungszahl dieses Zustands festgehalten werden.

Hier ist es nötig, den Anweisungsteil, der gerade erarbeitet wird, und den bereits in Teilen erstellten Vereinbarungsteil gleichzeitig in den Blick zu nehmen und parallel an beiden weiterzuarbeiten.

**Vereinbarungsteil:**

```

VAR
  (*Hilfsgrößen für die interne Funktionsweise des Automaten:*)
  AktuellerZustand: ZUSTAND; AktuellerZustandNR: CARDINAL;

```

**Weiter im Anweisungsteil:**

```
AktuellerZustand := nichtsEingeworfen;
```

Weiterhin muss herausgearbeitet werden, dass der Automat grundsätzlich niemals „aufhört“, und immer wieder eine neue Eingabe verlangen muss; die Eingabe, die mit `AutomatIO.MenuEingabe` „geholt“ wird, ist eine Zahl, die abzulegen ist: damit muss eine Variable vom Typ `CARDINAL` zur Verfügung gestellt werden:

**Vereinbarungsteil:** `EingabeNR: CARDINAL;`

**Weiter im Anweisungsteil:** `MenuEingabe(EingabeNR);`

Um die zugehörige Eingabe vom Typ `EINGABE` zu ermitteln, kann die Standardprozedur `VAL` benutzt werden:

```
Eingabe:= VAL(EINGABE, EingabeNR);
```

Da die Ermittlung der Ausgabe nicht zu einer Veränderung von Objekten führt, kann das erzielte Ergebnis in die zugehörige Ordnungszahl umgewandelt und der Prozedur `AutomatIO.AusgabeAnzeigen` zur Ausgabe übergeben werden:

```
AusgabeAnzeigen(ORD(AusgabeFunktion(Eingabe, AktuellerZustand)));
```

Als letztes ist noch zu gewährleisten, dass in Abhängigkeit von Eingabe und AktuellerZustand der nächste aktuelle Zustand ermittelt wird:

```
Uebergang(Eingabe, AktuellerZustand);
```

Abschließend sind noch die beiden Schritte: Umwandlung von AktuellerZustand in die zugehörige Ordnungszahl und Ausgabe des zu dieser Zahl gehörigen Strings zu implementieren:

```
AktuellerZustandNR:= ORD(AktuellerZustand);
ZustandAnzeigen(AktuellerZustandNR)
```

Diese Schritte sind „endlos“ zu wiederholen. Zusammengefasst ergibt sich der Modul `MuesliriegelAutomat` (siehe Algorithmus 4, Seite 203).

---

#### Algorithm 4 Modul MuesliriegelAutomat

---

```
MODULE MuesliriegelAutomat;
FROM AutomatIO IMPORT (* Prozeduren: *) Initialisierung, MenuEingabe, ZustandAnzeigen, AusgabeAnzeigen;
TYPE
  EINGABE = (pf50);
  AUSGABE = (ware, nichts);
  ZUSTAND = (nichtsEingeworfen, eingeworfen50, eingeworfen100);
VAR
  Eingabe: EINGABE;
  AktuellerZustand: ZUSTAND;
  Ausgabe: AUSGABE;
  EingabeNR, AktuellerZustandNR: CARDINAL;
PROCEDURE Uebergang(Eingabe: EINGABE; VAR Zustand: ZUSTAND);
(* siehe Algorithmus 2, Seite202 *)
PROCEDURE AusgabeFunktion (Eingabe: EINGABE; Zustand: ZUSTAND): AUSGABE;
(* siehe Algorithmus 2, Seite202 *)
BEGIN (* M u e s l i r i e g e l A u t o m a t *)
  Initialisierung(
    "Müsliriegelautomat Eingabe:",
    "50 Pfg.",
    "Müsliriegel|leer",
    "nichts eingeworfen|50 Pfg. eingeworfen|1,- DM eingeworfen"
  );
  AktuellerZustand:= nichtsEingeworfen;
  REPEAT
    MenuEingabe(EingabeNR);
    (* Ermittlung von Eingabe aus EingabeNR: *)
    Eingabe:= VAL(EINGABE, EingabeNR);
    AusgabeAnzeigen(ORD ( AusgabeFunktion (Eingabe, AktuellerZustand) ));
    Uebergang(Eingabe, AktuellerZustand);
    (* Ermittlung von AktuellerZustandNR aus AktuellerZustand: *)
    AktuellerZustandNR:=ORD(AktuellerZustand);
    ZustandAnzeigen(AktuellerZustandNR)
  UNTIL FALSE (* hört nie auf zu arbeiten; muss mit CTRL+Break abgebrochen werden. *)
END MuesliriegelAutomat.
```

---

An dieser Stelle ist anzumerken, dass für die folgenden Automaten die erarbeitete Struktur im Wesentlichen erhalten bleibt. Das Gerüst (Hauptprogramm) kann weiter benutzt werden. Daraus resultiert, dass anschließend mit verschiedenen Automaten experimentiert wird.

Als zweites Beispiel wird den Schülern die Aufgabe gegeben, einen Getränkeautomaten zu simulieren, in den sowohl 50 Pfg-Stücke als auch 1 DM-Stücke eingeworfen werden können. Das Getränk soll zum Preis von 1,50 DM erworben werden können. Die Typen `EINGABE`, `AUSGABE` und `ZUSTAND` könnten (in Anlehnung an das erste Beispiel) folgendermaßen deklariert sein:

```
TYPE EINGABE=(pfg50, dml);
AUSGABE=(ware, nichts);
ZUSTAND=(nichtsEingeworfen, eingeworfen50, eingeworfen100);
```

Die Umsetzung wird, wie am Beispiel des MuesliriegelAutomat bereits dokumentiert, für diesen einfachsten Getränkeautomaten von den Schülerinnen vollständig und möglichst selbstständig erarbeitet und soll hier nicht dokumentiert werden. Allerdings beginnen an dieser Stelle einige Schülerinnen erste Kritik an den bisher entwickelten Automaten zu üben: Was passiert, wenn in dem Zustand `eingeworfen100` noch `dml` eingegeben wird?

Im nächsten Schritt wird die Funktionalität des Getränkeautomaten erweitert:

Der Getränkeautomat soll zusätzlich einen *Warenhebel* und eine *Korrekturtaste* erhalten. Damit ändern sich die Eingabe- und Ausgabemöglichkeiten des Automats. Außerdem wird die Einführung eines weiteren Zustands notwendig: `eingeworfen150`. Damit ergibt sich bereits ein wesentlich unübersichtlicherer Automatengraph.

Die Tabelle stellt an dieser Stelle bereits einen Zwischenschritt dar, so dass der Graph auf Vollständigkeit geprüft werden kann.

Die Struktogramme mit ihren geschachtelten Mehrfachauswahlen werden an dieser Stelle nicht dokumentiert, sind aber ein Hilfsmittel, um an dieser Stelle etwaige Unklarheiten bzgl. des Ablaufs der Aktionen innerhalb der Prozeduren an konkreten Beispielen zu verdeutlichen.

Für die Umsetzung in den Modul `GetraenkeAutomat2` in Algorithmus 5, Seite 205, stellt sie bereits für die meisten Schülerinnen ein absolut notwendiges Hilfsmittel dar.

Zu Beginn der Unterrichtsreihe erarbeiten die Schülerinnen verschiedene selbst gewählte Transduktoren und setzen sie in funktionierende Programme um (in der Programmiersprache Modula-2). Dabei greifen sie auf die zur Verfügung gestellte Schnittstelle (`AutomatIO`) zurück, mit der sie die komplette Ein- und Ausgabe realisieren. Dies führt relativ schnell zu Erfolgserlebnissen und wurde von Schülerinnen in verschiedenen Kursen, die nach diesem Ansatz unterrichtet wurden, sehr positiv aufgenommen.

Nachdem diese Beispielautomaten „gebaut“ und ausprobiert wurden, sollten zur Übung verschiedene Erweiterungen vorgenommen werden, z. B. verschiedene Getränke zu unterschiedlichen Preisen. Damit können die Modelle bzgl. der Komplexität den in der Realität vorgefundenen Automaten zunehmend besser entsprechen.

Es können „ganz andere“ Automaten gebaut werden: das Element Zufall kann benutzt werden, um zufalls-gesteuerte Ausgaben (und damit verbunden Zustände) zu ermöglichen. Für diese werden dann Übergangsgraphen und Automatentabellen entwickelt.

Im Anschluss an die Erarbeitung von **Transduktoren** werden unter dem Stichwort

### „Es wird einfacher!“

**Akzeptoren** besprochen und realisiert. Automaten ohne Ausgabe, die über Endzustände anzeigen, zu welchem Ergebnis sie kommen (Akzeptoren).

Weitere Hinweise zur konkreten Durchführung der Unterrichtssequenz. Beispiele für die Einführung von Akzeptoren, die erfolgreich im Unterricht von den Schülerinnen bearbeitet wurden:

- Die Titanic geht unter, aber auf einem Schiff in der unmittelbaren Nähe schläft der Funker: Es ist ein Automat zu entwickeln, der den gesamten Funkverkehr „abhört“ und bei Auftreten von "SOS" in einen Endzustand (z. B. alarm) übergeht, aus dem er nicht mehr herauskann.
  1. Wir simulieren einen Safe: gutwillig, d.h. es kann probiert werden, solange es sinnvoll erscheint, d.h. es gibt nur einen Endzustand (z. B. offen).
  2. Der böswillige Safe: nur die richtige Eingabe der Ziffern öffnet (z. B. offen) alle anderen Eingaben führen in einen Endzustand (z. B. fuerImmerGeschlossen). Die Aufstellung des Automatengraphen und der -tabelle konnte (durch die Vorbereitungen bei Transduktoren) von den Schülerinnen bereits mit einer gewissen Routine selbstständig geleistet werden.

**Algorithm 5** Modul GetraenkeAutomat2

```

MODULE GetraenkeAutomat2;
FROM AutomatiO IMPORT (* Prozeduren: *)      Initialisierung, MenuEingabe, ZustandAnzeigen, AusgabeAnzeigen;
TYPE
  EINGABE = (pf50, dml, korrekturtaste, warenhebel);
  AUSGABE = (rueckgabe50, rueckgabe100, ware, nichts);
  ZUSTAND = (nichtsEingeworfen, eingeworfen50, eingeworfen100, eingeworfen150);
VAR
  Eingabe: EINGABE; AktuellerZustand: ZUSTAND; Ausgabe: AUSGABE;
  EingabeNR, AktuellerZustandNR: CARDINAL;
PROCEDURE AusgabeFunktion (Eingabe: EINGABE; Zustand: ZUSTAND): AUSGABE;
BEGIN
CASE Zustand OF
  nichtsEingeworfen:
    CASE Eingabe OF
      pf50, dml, korrekturtaste, warenhebel: RETURN nichts
    END (* CASE *)
  |eingeworfen50:
    CASE Eingabe OF
      pf50, dml: RETURN nichts
      |korrekturtaste: RETURN rueckgabe50
      |warenhebel: RETURN nichts
    END (* CASE *)
  |eingeworfen100:
    CASE Eingabe OF
      pf50: RETURN nichts
      |dml, korrekturtaste: RETURN rueckgabe100
      |warenhebel: RETURN nichts
    END (* CASE *)
  |eingeworfen150:
    CASE Eingabe OF
      pf50: RETURN rueckgabe50
      |dml: RETURN rueckgabe100
      |korrekturtaste: RETURN rueckgabe150
      |warenhebel: RETURN ware
    END (* CASE *)
END (* CASE *)
END AusgabeFunktion; PROCEDURE Uebergang(Eingabe: EINGABE; VAR Zustand: ZUSTAND);
BEGIN
CASE Zustand OF
  nichtsEingeworfen:
    CASE Eingabe OF
      pf50: Zustand:= eingeworfen50
      |dml: Zustand:= eingeworfen100
      ELSE (* nichts passiert *)
    END (* CASE *)
  |eingeworfen50:
    CASE Eingabe OF
      pf50: Zustand:= eingeworfen100
      |dml: Zustand:= eingeworfen150
      |korrekturtaste: Zustand:= nichtsEingeworfen
      ELSE (* nichts passiert *)
    END (* CASE *)
  |eingeworfen100:
    CASE Eingabe OF
      pf50: Zustand:= eingeworfen150
      |korrekturtaste: Zustand:= nichtsEingeworfen
      ELSE (* nichts passiert *)
    END (* CASE *)
  |eingeworfen150:
    CASE Eingabe OF
      korrekturtaste, warenhebel: Zustand:= nichtsEingeworfen
      ELSE (* nichts passiert *)
    END (* CASE *)
END (* CASE *)
END Uebergang;
BEGIN (* G e t r a e n k e A u t o m a t 2 *)
Initialisierung(
  "Getraenkeautomat2 Eingabe:",
  "50 Pfg.|1,- DM|Korrekturtaste|Coca (1,50 DM)",
  "Rueckgabe 50 Pfg.|Rueckgabe 1,- DM|Rueckgabe 1,50 DM|Coca Cola|leer",
  "nichts eingeworfen|50 Pfg. eingeworfen|1,-DM eingeworfen|1,50 DM eingeworfen");
AktuellerZustand:= nichtsEingeworfen;
REPEAT
  MenuEingabe(EingabeNR);
  (* Ermittlung von Eingabe aus EingabeNR: *)
  Eingabe:= VAL(EINGABE, EingabeNR);
  AusgabeAnzeigen(ORD ( AusgabeFunktion (Eingabe, AktuellerZustand) ));
  Uebergang(Eingabe, AktuellerZustand);
  (* Ermittlung von AktuellerZustandNR aus AktuellerZustand: *)
  AktuellerZustandNR:=ORD(AktuellerZustand);
  ZustandAnzeigen(AktuellerZustandNR)
UNTIL FALSE (* hört nie auf zu arbeiten; muss mit CTRL+Break abgebrochen werden. *)
END GetraenkeAutomat2.

```

**Eine gute Idee:** „Könnte man ein Programm nicht auch wie einen Safe schützen, so dass die Benutzer das Passwort kennen müssen?“ Diese Frage einer Schülerin wurde mit „Ja, das sollte sich realisieren lassen“ beantwortet. Nach einer Ideensammlung wurde daraufhin ein „böswilliger“ Safe-Automat entwickelt.

Dieser wurde von den Schülerinnen modelliert und getestet. Ein ablauffähiges Modul ist in Algorithmus 6, Seite 206) dokumentiert.

---

### Algorithm 6 Böswilliger Safe

---

```
MODULE BoeswilligerSafe;
FROM AutomatIO IMPORT Initialisierung, ...
TYPE
EINGABE= (...);
ZUSTAND= (...);
PROCEDURE Uebergang(Eingabe: EINGABE; VAR Zustand: ZUSTAND);
BEGIN
    CASE Zustand OF ...
        END (* CASE Zustand *)
    END Uebergang;

BEGIN (* BoeswilligerSafe *)
    Initialisierung( ... ); ...
END BoeswilligerSafe.
```

---

Die Erweiterung besteht im nächsten unabhängigen Schritt darin, dass ein „Nutz“-programm (Klient) durch die Benutzung des bereits erstellten Moduls (BoeswilligerSafe) einen Passwortschutz erhält. Es ist also ein Modul Passwort zu entwickeln, das den Schutz transparent zur Verfügung stellt. Damit kann die Funktionalität aus diesem Modul importiert werden (zur konkreten Umsetzung siehe Algorithmus 7, Seite 206).

---

### Algorithm 7 Passwortschnittstelle

---

```
DEFINITION MODULE Passwort;
(* (C)copyright L. Humbert;
Beginn: 16. November (11. Unterrichtswochen) letzte Änderung: 16. November *)
PROCEDURE Schutz(): BOOLEAN;
END Passwort.
```

---

Die Implementierung erfolgt, in dem der Anweisungsteil des MODULE BoeswilligerSafe genau das realisiert, was die Prozedur Schutz() leisten soll, also: vor das BEGIN wird die Zeile PROCEDURE Schutz():BOOLEAN; gesetzt, an das Ende (vor END BoeswilligerSafe.) muss dem „Rufenden“ mit RETURN AktuellerZustand=open mitgeteilt werden, ob die Benutzerin das Passwort korrekt eingegeben hat oder nicht.

An dieser Stelle ist es möglich, syntaktische Strukturen (in diesem Fall reguläre Grammatiken; Chomsky Typ 3) aus dem Graphen des Automaten abzuleiten, um so den **Zusammenhang zwischen Automaten und formalen Sprachen** herzustellen.

Als Schnittstelle für Akzeptoren wurde KlasAutomatIO entwickelt (siehe Algorithmus 8, Seite 206).

---

### Algorithm 8 Schnittstelle KlasAutomatIO

---

```
DEFINITION MODULE KlasAutomatIO;
(* (C)copyright 1992 by L. Humbert; GE-Dortmund-Scharnhorst;
Beginn : 8. Juli 1992 Letzte Änderung : 8. Juli 1992 *)
PROCEDURE InitialisierungAufzaehlungen (HauptMenu, UnterMenu, ZustandsStr: ARRAY OF CHAR);
(* HauptMenu: enthält alle möglichen Eingabeklassen, jeweils durch | getrennt;
UnterMenu: enthält alle möglichen Eingabestrings, jeweils innerhalb der Klasse durch | getrennt,
die Klassen werden durch "&" getrennt;
ZustandsStr: enthält alle möglichen Zustandsstrings, jeweils durch | getrennt; *)
PROCEDURE Eingabe(VAR KlassenNR: CARDINAL; VAR Wert: ARRAY OF CHAR);
(* Die KlassenNR und der tatsächlich eingegebene Wert werden zurückgegeben. Es wird Protokoll geführt. *)
PROCEDURE ZustandAnzeigen(ZustandsNR: CARDINAL);
(* Nicht der Zustand aus ZUSTAND, sondern die Nummer ist hier anzugeben! *)
END KlasAutomatIO.
```

---

Damit ist es möglich, „Klassen“ von Eingabemöglichkeiten zusammenzufassen und die Eingabe über ein Pull-downmenü vorzunehmen. Als Beispiel dient das Modul ReelleZahlenAkzept (Algorithmus 9, Seite 207).

**Algorithm 9** Modul ReelleZahlenAkzept

```

MODULE ReelleZahlenAkzept;
FROM KlasAutomatIO IMPORT (* Prozeduren: *)
  InitialisierungAufzaehlungen, Eingabe, ZustandAnzeigen;
TYPE
  EINGABE = (vorzeichen, ziffer, punkt, exponent, sonst);
  ZUSTAND = (leer, vorz, ganz, gebrochen, ohneExp, exp, expMitVorz, reelleZahl, falsch);
VAR
  EingabeKlasse: EINGABE; AktuellerZustand: ZUSTAND;
  EingabeNR, AktuellerZustandNR: CARDINAL;
  Wert: ARRAY [0..10] OF CHAR;
PROCEDURE Uebergang(Eingabe: EINGABE; VAR Zustand: ZUSTAND);
BEGIN
CASE Zustand OF
  leer:
    CASE Eingabe OF
      vorzeichen: Zustand:= vorz
      | ziffer: Zustand:= ganz
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | vorz:
    CASE Eingabe OF
      ziffer: Zustand:= ganz
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | ganz:
    CASE Eingabe OF
      ziffer: Zustand:= ganz
      | punkt: Zustand:= gebrochen
      | exponent: Zustand:= exp
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | gebrochen:
    CASE Eingabe OF
      ziffer: Zustand:= ohneExp
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | ohneExp:
    CASE Eingabe OF
      ziffer: Zustand:= ohneExp
      | exponent: Zustand:= exp
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | exp:
    CASE Eingabe OF
      vorzeichen: Zustand:= expMitVorz
      | ziffer: Zustand:= reelleZahl
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | expMitVorz:
    CASE Eingabe OF
      ziffer: Zustand:= reelleZahl
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | reelleZahl:
    CASE Eingabe OF
      ziffer: Zustand:= reelleZahl
      ELSE Zustand:= falsch
    END (* CASE Eingabe *)
  | falsch:
    ELSE Zustand:= falsch
  END (* CASE Zustand *)
END Uebergang;

BEGIN
InitialisierungAufzaehlungen(
  "Vorzeichen|Ziffer|Punkt|Exponent|Sonstiges",
  "+|-&0|1|2|3|4|5|6|7|8|9&.&E|D&A|B|C|F|G|a|b|c|d|e",
  "Leer|Vorzeichen|Ganz|Gebr.|ohne Exp.|Exp.|Exp. mit Vorzeichen|Reelle Zahl|Fehler");
AktuellerZustand:= leer;
REPEAT
  Eingabe(EingabeNR, Wert);
  (* Ermittlung von Eingabe aus EingabeNR: *)
  EingabeKlasse:= VAL(EINGABE, EingabeNR);
  Uebergang(EingabeKlasse, AktuellerZustand);
  (* Ermittlung von AktuellerZustandNR aus AktuellerZustand: *)
  AktuellerZustandNR:=ORD(AktuellerZustand);
  ZustandAnzeigen(AktuellerZustandNR)
UNTIL AktuellerZustand=falsch
END ReelleZahlenAkzept.

```

Weitere Anwendungsmöglichkeiten: Erkennung von Wörtern in Texten; Bezeichner in Programmiersprachen.

### „Es wird schwieriger!“

Automaten mit „Gedächtnis“: Kellerautomaten (Push Down Automata - PDA). Bei der Realisierung bietet sich eine Schnittstelle an, die Keller verwaltet: siehe 10, Seite 208).

---

#### Algorithm 10 Schnittstelle Stapel

---

```

DEFINITION MODULE Stapel;
FROM SYSTEM IMPORT BYTE;

TYPE STAPEL;

PROCEDURE Make(VAR Keller: STAPEL);
(* Ein neuer Keller wird angelegt; muss vor der ersten Benutzung aufgerufen werden! *)

PROCEDURE Push(Element: ARRAY OF BYTE; VAR Keller: STAPEL);
(* Fügt Element auf Keller. *)

PROCEDURE Pop (VAR Element: ARRAY OF BYTE; VAR Keller: STAPEL);
(* Holt Element aus Keller. *)

PROCEDURE Top (VAR Element: ARRAY OF BYTE; Keller: STAPEL);
(* Zeigt oberstes Element von Keller. *)

PROCEDURE IsEmpty (Keller: STAPEL): BOOLEAN;
(* Ist der Keller leer? *)

END Stapel.

```

---

### Anwendungsmöglichkeiten für PDAs: Erkennung korrekt geklammerter Ausdrücke

Hier ist es angebracht, die tatsächliche qualitative Veränderung der Strukturen anzusprechen, die mit solchen Automaten „geknackt“ werden können: Klammerstrukturen (deterministische, kontextfreie Grammatiken; echte Teilmenge der Chomsky-Grammatik Typ 2). Entwicklung eines Scanners für bestimmte, ausgewählte syntaktische Strukturen (deterministische, kontextfreie Grammatiken); korrekte Klammerung von Ausdrücken u. v. a. m.

### Zur gesellschaftlichen Bedeutung von Automaten

Ausblick: Bisher ist noch nicht auf die gesellschaftliche Bedeutung von Automaten abgehoben worden. Nun, hier gilt es folgende Facetten zu berücksichtigen: Automaten wird von vielen Menschen (nicht zuletzt auch von Jugendlichen) Objektivität, absolute Richtigkeit der Ergebnisse und last-not-least Intelligenz zugesprochen. Wenn das Abbild der Automaten hinsichtlich ihrer Leistungsfähigkeit untersucht wird (formale Sprachen, Sprachklassen) ist es möglich, Grenzen der programmierbaren Automaten zu erkennen: Church'sche These, Entscheidungs- und Halteproblem, Berechenbarkeit<sup>375</sup>. Der Turing-Test [Turing 1967] führt anschließend zum Einstieg in die Problematik der Künstlichen Intelligenz. (vgl. z. B. [Penrose 1991], [Dreyfus und Dreyfus 1987], [Dreyfus 1989], [Haller 1990]).

### Erkennung von Isomorphismen zwischen Automaten und zugehörigen syntaktischen Strukturen

Realisierung mit der vorgestellten aber inzwischen erweiterten Schnittstelle `KlassAutomatIO` - dabei ist der Wert des eingegebenen Zeichens für den Automaten nicht von Belang. Übergang auf „Klammerstrukturen“ sowie ihre Umsetzung durch PDA.

Ein erster Scanner: Vorbemerkungen: die Schülerinnen haben bisher keine selbstdeklarierten, linearen, statischen Listen kennengelernt.

---

<sup>375</sup> siehe u. a. [Hopcroft und Ullmann 1988], [Penrose 1991], [Winograd und Flores 1986]

Für einen Scanner ist die Kenntnis der Interna der Datenstruktur Liste nicht von Belang. Den Schülerinnen wird über eine definierte Schnittstelle eine Schlange zur Verfügung gestellt, die den Abstrakten Datentyp (ADT) Schlange mit allen Operationen realisiert wird.

Ausblick auf das zweite Halbjahr: Wird der Ansatz im zweiten Halbjahr weitergeführt, so steht mit dem vermittelten Grundgerüst die Bearbeitung von Problemen aus der Automatentheorie, dem Compilerbau und der Linguistik auf einem guten Fundament. Für das zweite Halbjahr bietet sich nach diesem Einstieg der Wechsel des Programmierparadigmas an, damit die Problemlösekompetenz, die durch diesen imperativ orientierten Ansatz entwickelt wurde, im Zusammenhang an einem anderen Paradigma weiter gefestigt werden kann.

## F.2 Gestaltung eines Grundkurses Informatik

### 11. Jahrgang:

- Vorbereitung:
  - Sammlung von Vorstellungen der Schülerinnen über die Wissenschaft Informatik
  - Vorstrukturierung der genannten Begriffe
- Rechnernetze und verteilte Systeme (RvS)
  - Einführung in die produktive Arbeit mit dem schulischen Intranet
  - Da Unterrichtszeit des 11. Jahrgangs auf die Erarbeitung der Voraussetzungen für die Arbeit mit den schulischen Informatiksystemen verwendet werden muss, ist die Thematisierung des Themenbereichs Rechnernetze und verteilte Systeme (RvS) mit dem Ziel des Erwerbs von Handlungskompetenzen zur effektiven Nutzung der schulischen Intranetinfrastruktur unabdingbar.
    1. personenbezogenes Accounting im schulischen Intranet und seine Konsequenz: Vertrauen ist gut, Kontrolle ist besser?
      - Überwachungsmöglichkeiten
    2. Erläuterungen zur Nutzung der serverbasierten Profilaufbewahrung im schulischen Intranet
    3. praktische Übungen:
      - (a) Einlogvorgang,
      - (b) Schreibtischmetapher für die Nutzung von graphischen Benutzungsoberflächen (engl.: Graphical User Interface – GUI) von Informatiksystemen
        - i. Organisationsmöglichkeiten – Anlegen von Ordnerstrukturen
        - ii. Hinweise auf explorative Arbeitsweisen (Nutzung der Hilfefunktion, Kurzbefehle, besondere Zeichen/Tasten)
      - (c) Starten und Beenden von Programmen
    4. Netiquette und ihre informatischen Hintergründe
    5. Einführung in die Arbeit mit dem konkreten Mailprogramm
      - allgemeine Funktionen
        - (a) Empfang von E-Mail
        - (b) Senden von E-Mail
        - (c) Organisationsmöglichkeiten für Ordnerstrukturen am Beispiel von Mailordnern
          - Hierarchie: flache vs. tiefe Schachtelung – Pro und Kontra
- elektronisches Publizieren (für Übungen wird ein einfacher Texteditor eingesetzt – die Ergebnisse werden in einem Standardbrowser betrachtet)
  1. Dokumentenbeschreibungssprachen: Zweck und Ziel
  2. von SGML nach HTML: zur geschichtlichen Entwicklung
  3. Klammerstrukturen als Beispiel für syntaktische Strukturen
  4. Struktur von Texten und ihre Repräsentation in HTML
    - Tags: html, head, body, header ( $h_i$  mit  $i \in \{1 \dots 6\}$ ), paragraph
  5. Einbindung von Bildern in Texte
    - Tag: image (img) – Attribute: source (src), width, height, border

## 6. Strukturieren von Dokumenten durch Verweise – Vernetzen

- Tag: Anchor (a) – Attribut: href="<url>"  
hier wird <url> := http://<rechnername>/<pfad>/<dokument> betrachtet und benutzt
- (a) Zur Strukturierung von Dokumenten mit Hilfe von Verweisen
  - \* tiefe vs. flache Hierarchie
- (b) HTML-Dokumente und Netiquette
  - \* Tag: anchor – Attribut: href="mailto:<mailadresse>" nach [Berners-Lee 1994]
- Überlegungen zu minimalen Anforderungen, die an HTML-Seiten zu stellen sind, führen zu einem von den Schülerinnen erarbeiteten Gestaltungsvorschlag für einen grundsätzlichen, minimalen Rahmen

## 7. Strukturierung innerhalb von Dokumenten: Listen und Tabellen

- 8. die weitere Entwicklung von Dokumentenbeschreibungssprachen:  
Normen: Vereinbarungen/Veränderungen/Erweiterungen z. B. XML

- Computer Supported Cooperative Learning

- erweiterte Schreibtischmetapher als Ausgangspunkt für Überlegungen zur kollaborativen Arbeit in vernetzten Strukturen  
Unterrichtsplanung:  
F.3.3 Einführung in die Arbeit mit einem webbasierten CSCW zur Kursplanung, S. 215
- Strukturüberlegungen: Anpassung der Bereiche an die Nutzungsanforderungen, die durch die Analyse der Struktur der Aufbau- und Ablauforganisation eines konkreten Arbeitsprozesses gefunden werden

- informatisches Modellieren am Beispiel der Objektorientierung

Grundlegende informatische Konzepte werden an Hand von nicht trivialen offenen Problemstellungen erarbeitet  
integriert in die objektorientierte Modellierung wird die Erstellung von Algorithmen unterrichtlich umgesetzt

### 1. Unterschiedliche Arten der Modellierung:

- (a) Strukturierung von Texten als Modellierung,
- (b) Verbinden und Anordnen von Dokumenten in Strukturen als Modellierung,
- (c) allgemeiner Modellbegriff und Ausprägungen in der Informatik

### 2. Einführung in die objektorientierte Modellierung (OOM)

- (a) das konkrete Informatiksystem als Grundlage für die Modellierung  
Unterrichtsplanung:  
F.3.1 Objektorientierte Modellierung am Beispiel des Schülercomputerarbeitsplatzes, S. 213
- (b) Modellierung graphisch orientierter Sachverhalte durch die Nutzung vorgegebener Klassen,
- (c) Ereignisse als Kennzeichen graphischer Benutzungsoberflächen und ihre Nutzung in der Modellierung interaktiver Informatiksysteme,
- (d) Modellierung einer offenen Problemsituation (Ampelanlage)  
Unterrichtsplanung:  
F.3.2 Modellierung grundlegender Elemente für ein Blinklicht, S. 214
- (e) Modellierung einer analogen Uhr – das Problem der Nebenläufigkeit

**12. Jahrgang:**

- Fachsystematik
  - effiziente Organisation von Daten mit Hilfe von Datenstrukturen  
Unterrichtsplanung:  
F.3.4 Das Fachkonzept Keller - Realisierung als Klasse, S. 217
  - Server-/Klienten-Modellierung  
Unterrichtsplanung:  
F.3.5 Modellierung von Server-/Clientstrukturen, S. 220
  - Anwendungsbereich
  - Auswahl und Modellierung konkreter Problemstellungen
  - Präsentationsunterstützung durch strukturierte Dokumentation
  - Computer Supported Cooperative Learning
    - \* Awareness (als Funktion)
    - \* Berichtswesen
    - \* Erarbeitung der notwendigen Grundlagen, um ein elektronisches Portfolio anzulegen
    - \* Anlegen eines Portfolios

## F.3 Ausgewählte Unterrichtsplanung und -durchführung

### F.3.1 Objektorientierte Modellierung am Beispiel des Schülercomputerarbeitsplatzes

#### Einordnung: Objektorientierte Modellierung – Klassenstruktur

##### Stundenlernziel

Die Schülerinnen erhalten einen Einblick in die objektorientierte Beschreibung des Computerarbeitsplatzes.

##### kognitive Zieldimension:

Die Schülerinnen

- beschreiben ihren Schülercomputerarbeitsplatz mit Hilfe einer einfachen Klassenstruktur, bestehend aus *Bildschirm*, *Tastatur*, *Maus*.
- entdecken, dass eine Klasse (*Stift*) nötig ist, um den Bildschirminhalt bearbeiten zu können.
- beschreiben vermutete Eigenschaften (Attribute) und Funktionen (Methoden), die die Klassen *Bildschirm*, *Tastatur*, *Maus* und *Stift* für die Arbeit haben können/sollen.
- füllen eine grafische Klassenbeschreibung mit den Attributen und Methoden.

##### affektive Zieldimension:

Die Schülerinnen

- entdecken, dass der Computerarbeitsplatz mit objektorientierten Begriffen beschrieben werden kann.

Phase	Inhalte	Form/Medien
Einstieg	Präsentation: Abbildung eines Computerarbeitsplatzes	Abbildung/Beamer
Problem	Impuls: Modell bilden - Beschreiben - Klassifizieren	
Erarbeitung	Bildung von Funktionseinheiten Zuordnung der Elemente der Abbildung zur Struktur Die Ergebnisse werden in HTML-Tabellen dokumentiert	Schülerpartnerarbeit HTML-Seite <sup>376</sup>
Zwischensicherung	Vorstellung der Ergebnisse	Schüler/Beamer
Problem	Attribute und Funktionen der Klassen	
Erarbeitung	Ergänzung der Klassentabellen um mögliche Attribute und Funktionen - je Klasse ein Attribut und eine Funktion	Schülerpartnerarbeit
Zusammenfassung	Präsentation ausgewählter Schülerlösungen	Schüler/Beamer
Ausblick/Hausaufg.	Was erwarten die Schülerinnen?	

Tabelle F.4: Unterrichtsverlauf – objektorientierte Beschreibung des Computerarbeitsplatzes

<sup>376</sup> Arbeitsanweisung (per Beamer) Die HTML-Seite erhält den Namen: <account>.stiftUndCo.html .

## methodisch-didaktische Entscheidungen

Es zeigte sich in Diskussionen mit den Schülerinnen, dass die Motivation im Fach Informatik sehr unterschiedlich ist: einige Schülerinnen forderten eine starke Orientierung des Unterrichts zur *Bedienung* (Keyboarding) von üblichen GUIs; einige verlangten nach einer stärkeren theoretischen Durchdringung; einige wünschen die Einbeziehung der Besprechung von Hardware (und ihrer Installation) in den Unterricht.

### F.3.2 Modellierung grundlegender Elemente für ein „Blinklicht“

**Einordnung:** Objektorientierte Modellierung – Kontrollstrukturen; Nutzung der Klassen: *BuntStift*, *Farbe*, *Hilfe*

**Reihe:** Objektorientierte Modellierung

- Verwenden gegebener Klassen
- Programmgesteuertes Zeichnen
- Strukturierung mit eigener Klasse
- Malen mit der Maus:
- Kontrollstrukturen (Schleife mit Eingangsbedingung, einseitige Verzweigung)
- Ampel - hier zuerst Blinklicht, dann Ampel
- Steuern einer Ampel durch die Tastatur

#### Stundenlernziel

Die Schülerinnen erweitern ihre Fähigkeiten zum Explorieren und Nutzen der Klassenbibliothek zur Bearbeitung eines konkreten Problems.

#### kognitive Zieldimension:

Die Schülerinnen

- *kennen die Funktion des Python-Inspektionskommandos **dir(Klasse)** als Möglichkeit, Attribute und Methoden von **Klasse** herauszufinden.*
- *dokumentieren Ergebnisse einer eigenständigen Inspektion, um sie den anderen Schülerinnen vorstellen zu können.*
- *strukturieren die für die Lösung des (kleinen) Gesamtproblems notwendigen Teilschritte.*
- *nutzen die neuen Methoden **setzeFuellMuster(1)** und **setzeFarbe(meineFarben.SCHWARZ)** der Klasse **BuntStift**, um gefüllte Flächen zu erzeugen.*
- *nutzen die Methode **warte(Millisekunden)** aus der Klasse **Hilfe**, um eine Zeitverzögerung zu realisieren.*
- *setzen eine gefundene Lösungsstruktur in ein Programm um, mit dem ein **Blinklicht** realisiert wird.*

#### affektive Zieldimension:

Die Schülerinnen

- *empfinden Freude an der Realisierung einer selbst entwickelten Problemlösung und der Realisierung einer korrekt funktionierenden Implementierung.*

Phase	Inhalte	Form/Medien
Einstieg	Präsentation eines <b>Blinklichts</b>	Beamer
Problem	<b>Impuls:</b> Was wird benötigt, um das Blinklicht realisieren zu können? erwartete Antworten: <i>Farbe, Füllmuster, Warten</i>	fragend- entwickelnd
Erarbeitung	<b>Arbeitsauftrag:</b> Inspektion der Klassen <i>BuntStift, Farbe</i> und <i>Hilfe</i> mit dem Ziel, die <b>notwendigen Elemente</b> für die Realisierung eines Blinklichts herauszuarbeiten und zu dokumentieren.	Arbeitsauftrag Beamer
	<b>Hinweis:</b> Nach Einlogvorgang, Aktivieren von Jython kann mit <i>from stiftUndCo import Klassenname; dir(Klassenname)</i> der Funktionsumfang einer Klasse gefunden werden Die Ergebnisse werden mit dem Editor in einer persönlichen Datei: <i>Klassenname.Dokumentation.txt</i> zusammengefasst	Partnerarbeit  Computer/Jython
Zwischensicherung	Vorstellung der Ergebnisse	Schülerin
Problem	<b>Arbeitsauftrag:</b> Realisierung eines Blinklichts durch Beschreibung (ggf. mit Hilfe) und formales Aufschreiben als Programm Abspeichern unter dem Dateinamen: <i>Blinklicht.py</i>	Arbeitsauftrag  Beamer
Erarbeitung	Unter Zuhilfenahme bereits erarbeiteter Problemlösungen wird das Programm zur Erzeugung des Blinklichts sukzessive entwickelt und interaktiv überprüft.	Partnerarbeit Computer Jython
Zusammenfassung	Präsentation ausgewählter Schülerlösungen	Schülerinnen
Ausblick/Hausaufg.	Erstellung einer Ampellichtleiste	

Tabelle F.5: Unterrichtsverlauf – objektorientierte Modellierung - Kontrollstrukturen

### F.3.3 Einführung in die Arbeit mit einem webbasierten CSCW zur Kursplanung

#### Einordnung:

#### Stundenlernziel

Die Schülerinnen verstehen den CSCW-Ansatz und nutzen BSCW, um eine Diskussion über die Arbeit im Informatikkurs zu führen, Kursinhalte und -methoden abzuwägen und Entscheidungen über den weiteren Verlauf des Kurses vorzubereiten.<sup>377</sup>

#### kognitive und affektive Zieldimension:

Die Schülerinnen

- helfen sich bei der Bearbeitung ihrer Aufgaben – gerade im Zusammenhang mit dem Handling auftretende Fragen können so oftmals ohne „Eingreifen“ des Lehrers geklärt werden.

#### kognitive Zieldimension:

Die Schülerinnen

- erkennen die grundlegende Struktur von CSCW mit Hilfe der Schreibtischmetapher.
- entdecken die Informations- und damit Kontrollmöglichkeiten, die CSCW mit sich bringen, indem sie die Informationen über Objekte im BSCW lesen.
- ordnen die „**Awareness**“-Funktion von CSCW den entsprechenden Symbolen des BSCW zu und nutzen diese Funktion, um sich über Änderungen im Arbeitsbereich (Ordner) zu informieren.

<sup>377</sup> oft wird von CSCW-Systemen gesprochen/geschrieben. Hier wird die „Kurzschreibweise“ ohne das Suffix „System“ verwendet.

- wissen, dass mit Hilfe der Elemente in der Kopfleiste des BSCW dem Ordner neue Objekte zugeordnet werden.
- aktivieren den Verweis zu einer Notiz, lesen sie und antworten, indem sie eigene Notizen verfassen.
- erstellen Notizen, um eine „Schreibdiskussion“ über ihre Wünsche zur Ausgestaltung der Arbeit im 12. Jahrgang im Fach Informatik zu führen.

#### affektive Zieldimension:

Die Schülerinnen

- empfinden die „Schreibdiskussion“ als neue und überraschende Möglichkeit, sich über inhaltliche Fragen auseinander zu setzen.
- spüren eine Befriedigung ihres Mitteilungsbedürfnisses durch die Nutzung des BSCW.

#### psychomotorische Zieldimension:

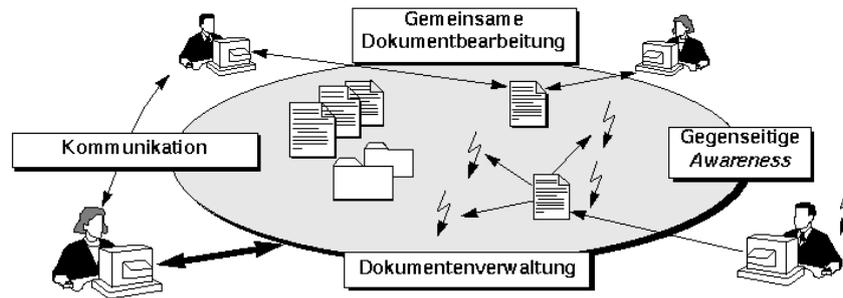
Die Schülerinnen

- üben ihre Fertigkeiten im Umgang mit Maus und Tastatur im Zusammenhang mit einer Applikation, die über Webseiten bedient wird.

Phase	Inhalte	Form/Medien
Einstieg	Abfragen und Kontrolle der Hausaufgaben: Einloggen in die Client-Server-Struktur; Einloggen in den BSCW-Server	Computer/ Einzel- und Partnerarbeit
Problem	Präsentation der Folie zu CSCW Impuls: „Schreibtisch“ „gemeinsam arbeiten“	Folie (Grafik des Arbeitsblatts ohne Text)
Erarbeitung	Schreibtischmetapher CSCW: Symbole und ihre Bedeutung Zuordnung der Elemente der Folie zur Struktur	fragend-entwickelnd Arbeitsblatt
Zwischensicherung	Vorstellung der Ergebnisse	Folie wird ausgefüllt
Problem	BSCW - die Realisierung eines webbasierten CSCW Hinweis auf „i“; Awareness; Notiz Die Kursplanung für 12.1 soll „diskutiert“ werden	Lehrerinformation ggf. Beamer (Symbole) Notiz im BSCW
2. Erarbeitung	Jede Schülerin kommentiert die vorliegende Notiz. Nach Aufforderung werden eigene Notizen verfaßt	interaktive Diskussion <sup>378</sup>
Zusammenfassung	im direkten Austausch werden die Erfahrungen und die Methoden einer Beurteilung durch die Schülerinnen unterzogen	Kreisgespräch
Ausblick	Die Schülerinnen fassen ihre Anmerkungen mit Hilfe weiterer Notizen im BSCW zusammen wird in der folgenden Unterrichtsstunde fortgesetzt	Notizen im BSCW Einzel-/Partnerarbeit

Tabelle F.6: Unterrichtsverlauf – Arbeit mit einem webbasierten CSCW zur Kursplanung

<sup>378</sup> Wird die *interaktive Diskussion* „gut angenommen“, ist das Kreisgespräch nicht notwendig.  
Dann wird die Zusammenfassung mit den Mitteln des BSCW durchgeführt.

Abbildung F.3: Folie – Arbeit mit einem webbasierten CSCW zur Kursplanung<sup>379</sup>

### F.3.4 Das Fachkonzept Keller - Realisierung als Klasse

#### Einordnung:

- **Reihe: Automaten – Transduktoren, Akzeptoren**
- **Einheit:** Graphische Darstellungen in der Informatik
- **Reihe:** Automatische Erstellung strukturierter Dokumente - informatische Fachkonzepte
  - Doppelstunde: Vorüberlegungen zur Klammerstruktur in XML Dokumenten am Beispiel von HTML
  - Einzelstunde: die Klasse Keller (Modellierungsüberlegungen)
  - Einzelstunde: Das Fachkonzept Keller - Vorüberlegungen zur Umsetzung mit Hilfe der Datenstruktur Liste
  - **Doppelstunde:** Das Fachkonzept Keller - Realisierung als Klasse

#### Stundenlernziel

Die Schülerinnen implementieren die Klasse **Keller** durch Benutzen der Datenstruktur Liste.

#### kognitive Zieldimension:

Die Schülerinnen

- erkennen, dass die Methode `__init__` (Konstruktor) eine leere Liste erzeugen muss.
- erkennen, dass die Methode `push(element)` durch die Listenmethode `append` realisiert werden kann
- erkennen, dass die Methode `pop()` durch zwei Aktionen auf der Liste realisiert wird: Zurückliefern des letzten Listenelements und Löschen des letzten Elements.
- erkennen, dass die Methode `empty()` die Anzahl der Elemente prüfen muss.
- erstellen die Klasse Keller in dem sie die Methoden `__init__`, `push`, `pop` und `empty` füllen.
- entwickeln eine Testmöglichkeit für die erstellte Klasse Keller und fügen diese dem Quellcode hinzu. (fakultativ)

<sup>379</sup> [Wasserschaff 1995, Abbildung 2-3]

**affektive Zieldimension:**

Die Schülerinnen

- stellen erfreut fest, dass die Klasse **Keller** kompakt implementiert werden kann und damit eine einfache Möglichkeit bereitsteht, die geforderte Funktionalität zur Verfügung zu stellen.

Phase	Inhalte	Form/Medien
Einstieg	Zusammenstellung der Anforderungen für die Klasse Keller bisherige Erkenntnisse zur Umsetzung Zusammenhänge zwischen Liste und Keller	Lehrerimpuls Schülerantworten Tafel
Problem	konkrete Realisierung der Klasse Keller	
Erarbeitung	formaler Rahmen: class ... def ...	fragend-entwickelnd/Heft
Zwischensicherung	Vorstellung ausgewählter Ergebnisse	Schülerinnen: Heft/Tafel
Erarbeitung	Eingabe der Klasse und Prüfung der syntaktischen Korrektheit	Schüler/Computer Editor und Jython
Problem	Testroutine für alle Methoden der Klasse	Lehrerimpuls
Erarbeitung	Entwicklung und Realisierung einer Testroutine Was muss in welcher Reihenfolge getestet werden? Entscheidung und Umsetzung Instanziierung der Klasse Keller, Prüfung mit empty pop bei leerem Keller? mehrmaliges push und dann pop	fragend-entwickelnd Schüler/Computer Editor und Jython  Vorführung durch Schülerinnen/Beamer
Ausblick	Erweiterung: Einrücken	Zusammenfassung und Diskussion

Tabelle F.7: Unterrichtsverlauf – Fachkonzept Keller – Realisierung als Klasse

Zum Zusammenhang zwischen Keller und Liste																	
Wie kann die Klasse Keller mit einer Liste realisiert werden?																	
<table border="1"> <thead> <tr> <th colspan="2">Klasse Keller</th> </tr> </thead> <tbody> <tr> <td colspan="2">Attribut</td> </tr> <tr> <td>lifo</td> <td>Liste</td> </tr> <tr> <td colspan="2">Methoden</td> </tr> <tr> <td>__init__(self)</td> <td>(Konstruktor)</td> </tr> <tr> <td>push(self, element)</td> <td></td> </tr> <tr> <td>pop(self)</td> <td>element</td> </tr> <tr> <td>empty(self)</td> <td>true oder false</td> </tr> </tbody> </table>		Klasse Keller		Attribut		lifo	Liste	Methoden		__init__(self)	(Konstruktor)	push(self, element)		pop(self)	element	empty(self)	true oder false
Klasse Keller																	
Attribut																	
lifo	Liste																
Methoden																	
__init__(self)	(Konstruktor)																
push(self, element)																	
pop(self)	element																
empty(self)	true oder false																
<table border="1"> <thead> <tr> <th>Datenstruktur Liste</th> </tr> </thead> <tbody> <tr> <td>self.lifo=[ ]</td> </tr> <tr> <td>self.lifo.append(element)</td> </tr> <tr> <td>element=self.lifo[-1]</td> </tr> <tr> <td>del (self.lifo[-1])</td> </tr> <tr> <td>return element</td> </tr> <tr> <td>return len(self.lifo)==0</td> </tr> </tbody> </table>		Datenstruktur Liste	self.lifo=[ ]	self.lifo.append(element)	element=self.lifo[-1]	del (self.lifo[-1])	return element	return len(self.lifo)==0									
Datenstruktur Liste																	
self.lifo=[ ]																	
self.lifo.append(element)																	
element=self.lifo[-1]																	
del (self.lifo[-1])																	
return element																	
return len(self.lifo)==0																	

Abbildung F.4: Tafel – Fachkonzept Keller – Realisierung als Klasse

---

**Algorithm 11** Klasse Keller (implementiert als Liste in Python)

---

```
class Keller:
    'Implementierung eines Kellers als Liste'

    def __init__(self):
        '''vorher: -,
        nachher: eine leere Liste, die den Stapel aufnehmen soll, ist angelegt
        '''
        self.lifo=[]

    def push(self,element):
        '''vorher: es gibt einen Stapel,
        nachher: element liegt oben auf dem Stapel'''
        self.lifo.append(element)

    def pop(self):
        '''liefert das oberste Element eines Stapels zurück;
        vorher: Stapel,
        nachher: element ist vom Stapel entfernt'''
        if not self.empty():
            element=self.lifo[-1]
            del self.lifo[-1]
        else:
            print 'der Stapel/Keller ist leer; pop ist nicht möglich'
            element=None
        return element

    def empty(self):
        '''liefert zurück, ob der Stapel leer ist
        vorher Stapel, nachher Stapel (der Stapel wird nicht verändert)'''
        return len(self.lifo)==0

    def top(self):
        '''das obere Element des 'Stapels' wird zurückgegeben (ansehen)'''
        return self.lifo[-1]

if __name__ == '__main__':
    ersterKeller = Keller()
    zweiterKeller = Keller()
    if ersterKeller.empty(): print 'ersterKeller ist leer'
    ersterKeller.push('</html>')
    # ...
    ersterKeller.push('</table>')
    print 'letztes Element auf ersterKeller:', ersterKeller.pop()
    zweiterKeller.push(ersterKeller.pop())
    print 'letztes Element auf zweiterKeller:', zweiterKeller.pop()
    if zweiterKeller.empty(): print 'zweiterKeller leer'
    while not ersterKeller.empty():
        print ersterKeller.pop()
    if ersterKeller.empty(): print 'ersterKeller leer'
    ersterKeller.pop()
```

---

### F.3.5 Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient

**Einordnung:** Rechnernetze und verteilte Systeme (RvS)

**Reihe:** Intranet/Internet aus Sicht von Informatiksystemen

- Schichtenmodell
- Vorstellung der Klasse UDFsocket (UDFsocket.py)
- Realisierung von Klienten
- Realisierung von Servern
- Nebenläufigkeit

#### Stundenlernziel

Die Schülerinnen beschreiben Anforderungen an die Struktur der parallelen Nutzung der Funktionalität von Server- und Klientenaktivitäten zum Nachrichtenaustausch im schulischen Intranet und realisieren prototypische Lösungen durch die Einrichtung eines nebenläufigen Prozesses.

#### kognitive Zieldimension:

Die Schülerinnen

- *kennen und nutzen die Begriffe: Klient, Server, Prozess.*
- *erkennen, dass das Problem, gleichzeitig Server (zur Entgegennahme von Daten) und Klient (der aktiv Daten aussenden soll) durch ein Programm realisieren zu wollen, sich nur dadurch lösen läßt, dass ein (neuer) Server-Prozess erzeugt wird, der im Hintergrund arbeitet.*
- *beschreiben die grundlegende Idee der Nebenläufigkeit mit eigenen Worten.*
- *wissen, dass **Nebenläufigkeit** programmtechnisch durch die Einrichtung von sog. **Threads** umgesetzt werden kann.*
- *realisieren die Pseudoparallelität von Server- und Klientenprozess durch die Installation eines nebenläufigen Prozesses, der die Serverfunktion realisiert.*

#### methodisch-didaktische Entscheidungen

Nebenläufigkeit ist als eines der Themen anzusehen, denen ausgewiesenermaßen eine hohe Bedeutung in der Arbeit mit verteilten Systemen zukommt. Vor allem in der Erstellung von Skripten, die, wie in dem Fall des Servers „permanent warten“ stellt sich die Frage, wie eine Umsetzung so erfolgen kann, dass mit dem System während dieser Wartezeit auch noch andere Arbeiten erledigt werden können.

In diesem Fall bietet sich Nebenläufigkeit als Lösungsidee an.

Zur Umsetzung kann in der hier zur Verwendung gebrachten Programmiersprache Jython auf ein mächtiges Sprachkonzept zurückgegriffen werden, das es ermöglicht, eine Funktion durch die Aktivierung einer Funktion ( `thread.start_new_thread` ) zur Nebenläufigkeit zu veranlassen.

Phase	Inhalte	Form/Medien
Einstieg	aktuelle Lösungen - Server und Klient getrennte Prozesse	Schülerinnen Intranet, Beamer
Problem	gleichzeitig Server und Klient in einem Skript realisieren	
Erarbeitung	Ideensammlung Parallelität rote <b>Fäden</b> (Threads)	Schülerinnen Lehrerinformation <sup>380</sup>
Zwischensicherung	Definitionen der Fachbegriffe im Kontext	Folie/Beamer
Problem	Umsetzung der Nebenläufigkeit Python-Spezifika zur Parametrisierung von <i>start_new_thread</i>	Lehrererklärung - Beamer <i>from thread import start_new_thread</i>
2. Erarbeitung	Realisierung - Umsetzung der Lösungsideen für den speziellen Fall	Schülerinnen, Informatiksysteme Intranet
Zusammenfassung	Vorstellung von Lösungsansätzen und Thematisieren von Problemen bei der Umsetzung	Schülerinnen
Ausblick	Planung Anforderungen an erweiterte Problemstellungen / -lösungen	

Tabelle F.8: Unterrichtsverlauf – Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient

Marco Thomas - Nebenläufigkeit im Informatikunterricht - Universität Potsdam - Didaktik der Informatik - Oberhausen, 18.09.1998 2

## Nebenläufigkeit

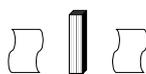
### Def. 1

- Zwei Ereignisse/Vorgänge heißen **nebenläufig**, wenn sie
- zeitweilig voneinander unabhängig auftreten bzw. ablaufen können und
  - abhängigkeits erzeugende Wirkungszusammenhänge zwischen ihnen bestehen können.



### Def. 2

- Nebenläufige Ereignisse oder Vorgänge heißen **parallel** oder gleichzeitig, wenn zwischen ihnen keine Wirkungszusammenhänge bestehen, die die Unabhängigkeit beeinflussen.



### Beispiele

- Während eines Vortrags können die Zuhörer parallel irgendwelchen anderen Tätigkeiten nachgehen.
- Ein zum Vortrag nebenläufiges Gespräch zwischen zwei Zuhörern zu einer aufgelegten Vortragsfolie kann gestört werden, wenn eine andere Vortragsfolie auf den Projektor aufgelegt wird.
- Zwei Prozesse auf einem Rechner mit einer CPU können nebenläufig, aber nicht parallel ablaufen.
- Zwei nicht-vernetzte Rechner arbeiten vollständig parallel (gleichzeitig).

Abbildung F.5: Folie/Tafel – Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient

<sup>380</sup> THOMAS, Marco Nebenläufigkeit im Schulfach Informatik – Legitimation und Vermittlung. <http://didaktik.cs.uni-potsdam.de/HyFISCH/Informieren/Nebenlaeufigkeit/Nebenlaeufigkeit.pdf>. September 1998, Folie 2

**Algorithm 12** UDFsocket.py – didaktisch gestaltete Schnittstelle

---

```
import socket

# Standardeinstellungen

class UDFsocket:
    def __init__(self, art="Klient", server="localhost", port=8081):
        self.rechnername=socket.gethostname()
        self.art= art
        self.server= server
        self.port= port
        self.absenderAdresse=server
        self.sockel = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        if art=="Server":
            self.sockel.bind((self.server, self.port))
        else: self.sockel.bind(("", 0))

    def datenempfang(self):
        daten, self.absenderAdresse = self.sockel.recvfrom(1024)
        return daten

    def getAbsender(self):
        return self.absenderAdresse

    def datenaussenden(self, daten):
        self.sockel.sendto(daten, (self.server, self.port))

    def dieserRechner(self):
        return self.rechnername
```

---

---

**Algorithm 13** UDFserver\_klient.py – „gleichzeitig“ Server und Klient mit einem Skript

---

```
from UDFsocket import UDFsocket
from thread import start_new_thread

class Server:
    def __init__(self, art="Server", meinName="ada", meinPort=8081):
        self.port=meinPort
        self.serverSteckdose= UDFsocket(art, meinName, self.port)

    def warteAufDaten(self):
        def warteSchleife(self):
            i=1
            while 1:
                eingangsDaten= self.serverSteckdose.datenempfang()
                print "\nEMPfang:", i, eingangsDaten, "von: Rechner", \
                    self.serverSteckdose.getAbsender()[0], \
                    "Port:", self.serverSteckdose.getAbsender()[1]
                i=i+1
            print "Server wartet auf Verbindung am Port:", self.port
            start_new_thread(warteSchleife, (self, ))

class Klient:
    def __init__(self, art="Klient", ServerName="ada", meinPort=8081):
        self.sender=UDFsocket(art, ServerName, 8081)

    def sende(self, text="Holy Guido! It's working."):
        self.sender.datenaussenden(text)

    def kontinuierlichTastatureingabensenden(self):
        while 1:
            naechsterText=raw_input("Was soll gesendet werden? ")
            self.sender.datenaussenden(naechsterText)
            print "... wird gesendet ..."

#-----
if __name__=="__main__":
    meinServer=Server(meinName="babagge")
    # der Server soll auf ankommende Daten warten - ...
    meinServer.warteAufDaten()

    meinKlient=Klient(ServerName="babagge")
    meinKlient.sende("mein Klient sendet mit freundlichen Grüßen ...")

    meinKlient.kontinuierlichTastatureingabensenden()
```

---

**Algorithm 14** UDFserver\_klient\_plus\_GUI.py – Server und Klient mit einer Benutzungsoberfläche

---

```

from UDFsocket import UDFsocket
from thread import start_new_thread
from java import awt
from pawt import swing

class Server:
    def __init__(self, art="Server", meinName="", meinPort=8081):
        self.port=meinPort
        self.serverSteckdose= UDFsocket(art, meinName, self.port)
        self.empfangeneDaten=""

    def warteAufDaten(self):
        def warteSchleife(self):
            i=1
            while 1:
                eingangsDaten= self.serverSteckdose.datenempfang()
                self.empfangeneDaten=str(i)+ " " +str(eingangsDaten) + \
                " von: Rechner " + str(self.serverSteckdose.getAbsender()[0]) + \
                " Port: " + str(self.serverSteckdose.getAbsender()[1])
                self.datenanzeige(self.empfangeneDaten)
                i=i+1
            self.empfangeneDaten="Server wartet auf Verbindung am Port:"+str(self.port)
            start_new_thread(warteSchleife, (self, ))

        def datenanzeige(self, text):
            pass # die Methode muss überschrieben werden

class Klient:
    def __init__(self, art="Klient", ServerName="", meinPort=8081):
        self.sender=UDFsocket(art, ServerName, 8081)

    def sende(self, text="Holy Guido! It's working."):
        self.sender.datenaussenden(text)
        print "Klient:",text

class Modell:
    def __init__(self,rechnername="haspe", servername="haspe"):
        self.meinServer=Server(meinName=rechnername) # hat ein Objekt der Klasse Server
        self.meinServer.warteAufDaten()
        self.meinKlient=Klient(ServerName=servername) # hat ein Objekt der Klasse Klient
        self.meinKlient.sende(" Na ... es klappt ...")

class ViewControl:
    def __init__(self, modell):
        self.modell=modell # kennt das Modell
        self.aufschrift = 'eingegebene Daten absenden'
        self.knoepfe = swing.JPanel(awt.GridLayout(1, 1)) # Anordnung
        self.serverDatenempfang = swing.JTextField()
        self.modell.meinServer.datenanzeige = self.serverDatenempfang.setText # Methode füllen
        self.klientenDaten = swing.JTextField()
        self.knopf = swing.JButton(self.aufschrift)
        self.knopf.actionPerformed = self.aktionDruecke
        oberflaeche = swing.JPanel(awt.BorderLayout())
        oberflaeche.add("North", self.serverDatenempfang)
        oberflaeche.add("Center", self.klientenDaten)
        oberflaeche.add("South", self.knopf)
        swing.test(oberflaeche)

    def aktionDruecke(self,ereignis): # Verbindung von Ereignis und Aktion
        self.modell.meinKlient.sende(self.klientenDaten.text)
        self.klientenDaten.setText("")

if __name__=="__main__":
    modell= Modell("stec", "stec")
    meinMVC= ViewControl(modell)

```

---

# Anhang G

## Schemata zum Problemlösen

### G.1 Imperativ – strukturiertes Problemlösen

#### Problemstellung erkannt?

1. Ist die Problemstellung verstanden bzw. wie lautet das eigentliche Problem?
  - Wenn die Problemstellung nicht verstanden ist, dann weitere Angaben zur Problemstellung einholen.
2. Hat das Problem bereits einen Namen?
  - Wenn nicht, dann dem Problem einen kurzen und treffenden Namen geben.
3. Was ist bekannt, bzw. was ist gegeben?
  - Zusammenstellen aller bekannten Eingabegrößen:  
jeder Eingabegröße einen Namen geben und seine Art bzw. seinen Typ beschreiben.
4. Was ist unbekannt, bzw. was ist gesucht?
  - Zusammenstellen aller bekannten Ausgabegrößen:  
jeder Ausgabegröße einen Namen geben und seine Art bzw. seinen Typ beschreiben.
5. Welche Bedingungen für Ein- und Ausgabegrößen müssen erfüllt sein oder werden gefordert?
  - Zusammenstellen dieser Bedingungen (z. B. formelmäßige Zusammenhänge).

#### Entwurf eines Lösungsplans

6. Ist dasselbe Problem oder ein ähnliches bzw. vergleichbares Problem bekannt?
  - Wenn ja, dann Kenntnisse über die Lösung dieses Problems beschaffen.
7. Ist ein allgemeineres Problem bekannt?
  - Wenn ja, dann Kenntnisse über die Lösung dieses Problems beschaffen. Es ist zu prüfen, ob das gegebene Problem als Sonderfall des allgemeinen Problems behandelt werden kann.
    - Wenn ja, dann kann zur Lösung die allgemeine Lösung angewendet werden.
    - Wenn sich das Problem verallgemeinern läßt, ohne das die Lösung erheblich schwieriger wird, dann wird das allgemeinere Problem gelöst.

#### Lösung des Problems

8. Entwickle eine Problemlösung und verfeinere sie nach und nach.

**Hinweis:** Jeder der angegebenen Schritte ist schriftlich durchzuführen!

## G.2 Objektorientierte Modellierung

Problemstellung erkannt?

1. Ist die Problemstellung verstanden bzw. wie lautet das eigentliche Problem?
  - Wenn die Problemstellung nicht verstanden ist, dann weitere Angaben zur Problemstellung einholen.
  - Formuliere die Problemstellung mit eigenen Worten (schriftlich).
2. Kandidaten für Klassen
  - Aus der originalen und der eigenen Problemformulierung sind alle Substantive herauszufiltern (schriftlich) und vermutete Kandidaten für Klassen zu kennzeichnen.

CRC-Karten

3. Beziehungen zwischen vermuteten Klassen
  - Es für jede vermutete Klasse wird eine Karteikarte angelegt.
    - Auf jeder Karteikarte ist der Name der Klasse, ihre Aufgabe(n) und mögliche Partner(klassen) zu vermerken.

Klassenstruktur und -beziehungen

4. Ordnung der beschrifteten Karteikarten
  - (a) Hierarchische Strukturierung:  
die Karteikarte einer abgeleiteten Klasse wird auf die Karteikarte der konkreten Klasse gelegt.
  - (b) Beziehungen zwischen Klassen (**hat**, **kennt**) werden durch Anordnen der Karteikarten auf dem Tisch sichtbar und können durch Probieren leicht verändert werden.

Ausgestaltung der Klassen

5. Attribute von Klassen
  - Isolierung der in der Problemstellung auftretende **Adjektive**. Sie deuten auf Attribute von Klassen.
6. Finden von Kandidaten für Methoden
  - Die **Verben** aus den Problemformulierungen geben Hinweise auf mögliche Methoden.
7. Ordne die vermuteten Methoden den Klassen zu.
  - Prüfe, ob übriggebliebene Substantive zu Klassen gehören.
    - Wenn ja, dann handelt es sich ggf. um Attribute von Klassen.

UML-Klassendiagramm

8. Zusammenstellen der Analyseergebnisse
  - Die Ergebnisse werden in einem Klassendiagramm zusammenfassend dargestellt.
  - Die Grundstruktur des zugehörigen Programmcodes ergibt sich durch die entsprechende Übersetzungsschablone.
9. Methoden der Klassen schrittweise ausgestalten.
  - Werden in dieser Phase Änderungen an Ergebnissen vorheriger Phasen vorgenommen, so ist die Analyse an dem zugehörigen Punkt wieder aufzunehmen.

# **Verzeichnisse, Listen, Index**



# Literaturverzeichnis

- [Abbott 1983] ABBOTT, Russell J.: Program Design by Informal English Descriptions. In: *CACM* 26 (1983), November, Nr. 11, S. 882–894
- [ACM 1993] ACM: *Model High School Computer Science Curriculum*. New York : ACM (Association for Computing Machinery), 1993. – <http://www.acm.org/education/hscur/index.html>
- [ACM 1997] ACM: *Model High School Computer Science Curriculum*. November 1997. – <http://www.acm.org/education/hscur/index.html>
- [Adam 1971] ADAM, Adolf: *Informatik. Probleme der Mit- und Umwelt*. Opladen : Westdeutscher Verlag, 1971
- [Aebli 1998] AEBLI, Hans: *Zwölf Grundformen des Lehrens. Eine allgemeine Didaktik auf psychologischer Grundlage. Medien und Inhalte didaktischer Kommunikation. Der Lernzyklus*. Stuttgart : Klett-Cotta, 1998. – frühere Auflage: 1983
- [Alex u. a. 2002] ALEX, Markus ; AZEM, Ahmad ; FRICKE, Tobias ; ISCAN, Hülya ; ROHR, Oliver ; ROSSKOPF, Michael ; SÖCHTIG, Gunnar ; STACHNIK, Adam ; VUKUSIC, Ivana ; WERNER, Thomas ; WOJCIECHOWSKI, Alexander ; BRINDA, Torsten (Hrsg.) ; HUMBERT, Ludger (Hrsg.) ; SIROCIC, Birgit (Hrsg.): *Endbericht der Projektgruppe Nr. 403 des Fachbereichs Informatik der Universität Dortmund – Lernumgebung für objektorientiertes Modellieren im Informatikunterricht (LEO)*. Dortmund : Fachbereich Informatik, August 2002. – [http://www.ham.nw.schule.de/pub/nj\\_bscw.cgi/d29298/Uni\\_Do\\_Abschlussbericht\\_PG\\_LEO.pdf](http://www.ham.nw.schule.de/pub/nj_bscw.cgi/d29298/Uni_Do_Abschlussbericht_PG_LEO.pdf) – geprüft: 5. Dezember 2002
- [Alex 2002] ALEX, Wulf: *Einführung in das Internet*. Universität Karlsruhe. Mai 2002. – <http://www.ciw.uni-karlsruhe.de/skriptum/skriptumI14.ps.gz> – geprüft: 15. Juli 2002
- [Altermann-Köster u. a. 1990] ALTERMANN-KÖSTER, Marita ; HOLTAPPELS, Heinz G. ; KANDERS, Michael ; PFEIFFER, Hermann ; WITT, Claudia de: *Bildung über Computer?* Weinheim : Juventa Verlag, 1990
- [Andelfinger und Voigt 1986] ANDELFINGER, Bernhard ; VOIGT, Jörg: Vorführstunden und alltäglicher Mathematikunterricht. Zur Ausbildung von Referendaren im Fach Mathematik (SI/SII). In: *Zentralblatt für Didaktik der Mathematik* 18 (1986), S. 2–9
- [Anderes u. a. 1999] ANDERES, Michael ; GOORHUIS, Henk ; NIEDERER, Ruedi ; HUGELSHOFER, René (Hrsg.): *Informatik. Anwendungen – Algorithmen – Computer – Gesellschaft*. 5. Aufl. Aarau, Frankfurt a. M., Salzburg : Sauerländer, 1999 (Bildung). – 1. Aufl. 1988 – Materialien zu dem Lehrbuch: <http://www.bildung-sauerlaender.ch/download/hugelshofer/index.html> – geprüft: 13. Oktober 2002
- [Anders 1983] ANDERS, Günter: *Die Antiquiertheit des Menschen*. 6. Aufl. Beck, 1983. – Originalauflage: 1956

- [Anderson 2001] ANDERSON, John R.: *Kognitive Psychologie*. 3. Aufl. Heidelberg, Berlin : Spektrum der Wissenschaften, Januar 2001. – Originaltitel: *Cognitive Psychology and its Implications*, W. H. Freeman and Company, New York, 2002, 5th Edition, übersetzt und herausgegeben von Ralf Graf und Joachim Grabowski
- [Appelt und Busbach 1996] APPELT, Wolfgang ; BUSBACH, Uwe: The BSCW System: A WWW based Application to Support Cooperation of Distributed Groups. In: *Proc. of WET ICE 96: Collaborating on the Internet: The World-Wide Web and Beyond*. Los Alamitos : IEEE Computer Society Press, June 1996, S. 304–310
- [Arctaedius 1996] ARCTAEDIUS, Micke. *ObjectPlant*. <http://www.arctaedius.com/ObjectPlant/> – geprüft: 3. Dezember 2002. 1996
- [Arlt 1982] ARLT, Wolfgang: Einführung in die Schulinformatik. In: HAAS, Hans W. (Hrsg.) ; WILDENBERG, Detlef (Hrsg.): *Informatik für Lehrer – Studentexte und Handreichungen für den Unterricht* Bd. 1 : Einführung in die Schulinformatik. München : Oldenbourg Verlag, 1982, S. 1–61
- [Arlt u. a. 1982] ARLT, Wolfgang ; DIRNBERGER, Josef ; KLÖCKNER, Konrad ; SCHAUER, Helmut ; SCHÜTZ, Ute ; TAVOLATO, Paul ; HAAS, Hans W. (Hrsg.) ; WILDENBERG, Detlef (Hrsg.): *Informatik für Lehrer – Studentexte und Handreichungen für den Unterricht*. Bd. 1 : Einführung in die Schulinformatik. München : Oldenbourg Verlag, 1982
- [Arlt und Haefner 1984] ARLT, Wolfgang (Hrsg.) ; HAEFNER, Klaus (Hrsg.): *Informatik als Herausforderung an Schule und Ausbildung*. Berlin, Heidelberg : Springer, Oktober 1984 (Informatik-Fachberichte 90)
- [Backus 1959] BACKUS, John W.: *The syntax and Semantics of the Proposed International Algebraic Language of the Zuerich ACM-GRAMM conference*. ICIP Paris. Paris : UNESCO, June 1959
- [Backus u. a. 1963] BACKUS, John W. ; BAUER, Friedrich L. ; GREEN, Julien ; KATZ, C. ; MCCARTHY, John ; NAUR, P. ; PERLIS, Alan J. ; RUTISHAUSER, Heinz ; SAMUELSON, Klaus ; VAUQUOIS, Bernhard ; WEGSTEIN, Joseph H. ; WIJNGAARDEN, Adriaan van ; WOODGER, Michael ; NAUR, Peter (Hrsg.): *Revised Report on the Algorithmic Language Algol 60*. diverse Zeitschriften, 1963. – CACM, Vol. 6, pp 1–17; The Computer Journal, Vol. 9, p. 349; Numerische Mathematik, Vol. 4, p. 420. <http://www.masswerk.at/algol60/report.htm> – geprüft: 14. Juli 2002
- [Balzert 1999] BALZERT, Heide: *Lehrbuch der Objektmodellierung: Analyse und Entwurf*. Heidelberg : Spektrum Akademischer Verlag, 1999 (Lehrbücher der Informatik)
- [Balzert 1976] BALZERT, Helmut: *Informatik: 1. Vom Problem zum Programm – Hauptband*. 1. Aufl. München : Hueber-Holzmann Verlag, 1976
- [Balzert 1977a] BALZERT, Helmut: Einige Gedanken zu Informatiklerninhalten und zur Methodik in verschiedenen Ausbildungsstufen und -bereichen. In: **[Bauersfeld u. a. 1977]**, S. 49–65. – (Band I)
- [Balzert 1977b] BALZERT, Helmut: *Informatik: 1. Vom Problem zum Programm – Lösungsband mit methodisch–didaktischer Einführung*. 1. Aufl. München : Hueber-Holzmann Verlag, 1977b
- [Balzert 1977c] BALZERT, Helmut: Vom Problem zum Programm – ein methodischer Ansatz zum systematischen und strukturierten Programmieren. In: GENSCHE, Gunter (Hrsg.): *IDOC [Informatik – Dokumentation]*. Paderborn : Forschung- und Entwicklungszentrum für objektivierte Lehr- und Lernverfahren (FEoLL) - Institut für Bildungsinformatik - Projekt IDOC, April 1977c (Service 4), S. 1–29
- [Balzert 1979] BALZERT, Helmut: *Informatik: 1. Vom Problem zum Programm – Lösungsband mit methodisch–didaktischer Einführung*. 2. Aufl. München : Hueber-Holzmann Verlag, 1979. – 1. Aufl. 1977

- [Balzert 1983] BALZERT, Helmut: *Informatik: 1. Vom Problem zum Programm – Hauptband*. 2. Aufl. München : Hueber-Holzmann Verlag, 1983. – 1. Aufl. 1976
- [Balzert 1996] BALZERT, Helmut: *Lehrbuch der Software-Technik: Software-Entwicklung*. Heidelberg : Spektrum Akademischer Verlag, 1996 (Lehrbücher der Informatik)
- [Bamberg 1998] BAMBERG, Horst: Technische Universität Berlin. In: [Görke 1998], S. 28–29
- [Bauer 1974] BAUER, Friedrich L.: Was heißt und was ist Informatik? Merkmale zur Orientierung über eine neue wissenschaftlichen Disziplin. In: *IBM Nachrichten* 24 (1974), Nr. 223, S. 333–337
- [Bauer 1998] BAUER, Friedrich L.: *Historische Notizen* Wer erfand den von-Neumann-Rechner? In: *Informatik Spektrum* 21 (1998), April, Nr. 2, S. 84–88
- [Bauer und Wössner 1972] BAUER, Friedrich L. ; WÖSSNER, H.: The “Plankalkül“ of Konrad Zuse: A Forerunner of Today’s Programming Languages. In: *CACM* 15 (1972), July, Nr. 7, S. 678–685. – <http://www.zib.de/zuse/Inhalt/Kommentare/Pdf/0679.pdf> – geprüft: 13. Juli 2002
- [Bauersfeld u. a. 1977] BAUERSFELD, H. (Hrsg.) ; OTTE, M. (Hrsg.) ; STEINER, Hans G. (Hrsg.): *Informatik im Unterricht der Sekundarstufe II: Grundfragen, Probleme und Tendenzen mit Bezug auf allgemeinbildende und berufsqualifizierende Ausbildungsgänge*. Bielefeld : Universität Bielefeld, 1977 (Schriftenreihe des IDM (Institut für Didaktik der Mathematik) Nr 15 (Band I) und 16 (Band II)). – Arbeitstagung: Bielefeld 12.-14. September 1977
- [Baumann 1991] BAUMANN, Rüdiger: *Prolog. Einführungskurs*. Stuttgart : Klett-Schulbuchverlag, 1991
- [Baumann 1993] BAUMANN, Rüdiger: Ziele und Inhalte des Informatikunterrichts. In: *Zentralblatt für Didaktik der Mathematik* 25 (1993), Nr. 1, S. 9–19. – <http://www.ham.nw.schule.de/pub/bscw.cgi/d29641/baumann.pdf> – geprüft: 16. Dezember 2002
- [Baumann 1998] BAUMANN, Rüdiger: Fundamentale Ideen der Informatik – gibt es das? In: [Koerber und Peters 1998], S. 89–107. – <http://log-in.fachdid.fu-berlin.de/IBiD/index.html> – geprüft 17. Mai 2002
- [Baumert 2002] BAUMERT, Jürgen: Deutschland im internationalen Bildungsvergleich. In: KILLIUS, Nelson (Hrsg.) ; KLUGE, Jürgen (Hrsg.) ; REISCH, Linda (Hrsg.): *Die Zukunft der Bildung*. Frankfurt a. M. : Suhrkamp, Juni 2002, S. 100–150
- [Baumert u. a. 2002] BAUMERT, Jürgen (Hrsg.) ; ARTELT, Cordula (Hrsg.) ; KLIEME, Eckhard (Hrsg.) ; NEUBRAND, Michael (Hrsg.) ; PRENZEL, Manfred (Hrsg.) ; SCHIEFELE, Ulrich (Hrsg.) ; SCHNEIDER, Wolfgang (Hrsg.) ; TILLMANN, Klaus-Jürgen (Hrsg.) ; WEISS, Manfred (Hrsg.): *PISA 2000 – Die Länder der Bundesrepublik Deutschland im Vergleich*. Opladen : Leske + Budrich, Juni 2002
- [Baumert u. a. 1998] BAUMERT, Jürgen ; BOS, Wilfried ; WATERMANN, Rainer. *TIMSS/III Schülerleistungen in Mathematik und den Naturwissenschaften am Ende der Sekundarstufe II im internationalen Vergleich – Zusammenfassung deskriptiver Ergebnisse*. [http://www.mpib-berlin.mpg.de/TIMSS\\_III/Zusammenfassung.htm](http://www.mpib-berlin.mpg.de/TIMSS_III/Zusammenfassung.htm). Mai 1998
- [Baumert u. a. 2001] BAUMERT, Jürgen (Hrsg.) ; KLIEME, Eckhard (Hrsg.) ; NEUBRAND, Michael (Hrsg.) ; PRENZEL, Manfred (Hrsg.) ; SCHIEFELE, Ulrich (Hrsg.) ; SCHNEIDER, Wolfgang (Hrsg.) ; STANAT, Petra (Hrsg.) ; TILLMANN, Klaus-Jürgen (Hrsg.) ; WEISS, Manfred (Hrsg.): *PISA 2000: Basiskompetenzen von Schülerinnen und Schülern im internationalen Vergleich*. Opladen : Leske + Budrich, 2001
- [Baumert u. a. 1999] BAUMERT, Jürgen ; KLIEME, Eckhard ; NEUBRAND, Michael ; PRENZEL, Manfred ; SCHIEFELE, Ulrich ; SCHNEIDER, Wolfgang ; TILLMANN, Klaus-Jürgen ; WEISS, Manfred: *Erfassung fächerübergreifender Problemlösekompetenzen in PISA*. Berlin : Max-Planck-Institut für Bildungsforschung, November 1999. – <http://www.mpib-berlin.mpg.de/pisa/Problemloesen.pdf> – geprüft: 25. Oktober 2002

- [Baumert u. a. 1998] BAUMERT, Jürgen ; LEHMANN, Rainer ; LEHRKE, Manfred ; SCHMITZ, Bernd ; CLAUSEN, Marten ; HOSENFELD, Ingmar ; KÖLLER, Olaf ; NEUBRAND, Johanna: *TIMSS – Mathematisch-Naturwissenschaftlicher Unterricht im internationalen Vergleich – Deskriptive Befunde*. Berlin, Kiel : Max-Planck-Institut für Bildungsforschung, 1998
- [Baumgartner 1997] BAUMGARTNER, Peter: Didaktische Anforderungen an (multimediale) Lernsoftware. In: ISSING, Ludwig J. (Hrsg.) ; KLIMSA, Paul (Hrsg.): *Information und Lernen mit Multimedia*. Weinheim : Psychologie-Verlags-Union, 1997. – 2. Aufl. [http://ioll.uibk.ac.at/php/documents/pdf/did\\_anforderungen.pdf](http://ioll.uibk.ac.at/php/documents/pdf/did_anforderungen.pdf) – geprüft: 18. September 2002, S. 241–252
- [Bayerisches Kultusministerium 2000] BAYERISCHES KULTUSMINISTERIUM (Hrsg.): *Lehrplanvorschlag für Informatik am Europäischen Gymnasium, Typ III*. München : Staatsministerium für Unterricht und Kultus, April 2000. – <http://www.isb.bayern.de/gym/informat/schulver/lp-inf-egy3.pdf> – geprüft: 1. Dezember 2002
- [Beckmann 1998] BECKMANN, Renate: *Speicher-Synthese für allgemeine Multiprozessor-Systeme mit Constraint-Logikprogrammierung*. Düsseldorf : VDI-Verlag, 1998 (Fortschritt-Berichte VDI Reihe 20: Rechnerunterstützte Verfahren Nr. 285)
- [Berg 2002] BERG, Gary A.: The Big Questions. In: *International Journal on E-Learning* 1 (2002), April-June, Nr. 1/2, S. 5–6
- [Berger 1997] BERGER, Peter: Das 'Computer-Weltbild' von Lehrern. In: [Hoppe und Luther 1997], S. 27–39
- [Berger 2001] BERGER, Peter: *Computer und Weltbild – Habitualisierte Konzeptionen von der Welt der Computer*. 1. Aufl. Wiesbaden : Westdeutscher Verlag, Juni 2001. – Inhalt, Einleitung: <http://home.ph-freiburg.de/bergerpe/pap/cuw.pdf> – geprüft: 4. Oktober 2002
- [Berners-Lee 1994] BERNERS-LEE, Tim. *RFC 1630 – Uniform Resource Identifiers in WWW*. <http://www.faqs.org/rfcs/rfc1630.html>. June 1994
- [Bernfeld 1981] BERNFELD, Siegfried ; BAECKER, Dirk (Hrsg.): *Sisyphos oder die Grenzen der Erziehung*. 4. Aufl. Frankfurt a. M. : Suhrkamp Verlag, 1981 (suhrkamp taschenbuch wissenschaft 37). – Erstmals 1925 erschienen im Internationalen Psychoanalytischen Verlag
- [Bittner 2001] BITTNER, Peter: Informatisches Handeln und Kritische Theorie. Elemente einer Kritischen Theorie der Informatik. In: [Nake u. a. 2001], S. 21–26
- [Bittner 2002] BITTNER, Peter. *Zum Selbstverständnis einer Disziplin. Informatik & Gesellschaft – 2. Vorlesung*. [http://waste.informatik.hu-berlin.de/peter/lehre/i+g\\_ss2002/i+g\\_ss2002\\_nr2\\_020425\\_sec.pdf](http://waste.informatik.hu-berlin.de/peter/lehre/i+g_ss2002/i+g_ss2002_nr2_020425_sec.pdf) – geprüft: 13. Juli 2002. April 2002
- [Blankertz 1980a] ; BLANKERTZ, Stefan (Hrsg.): *Die Schule von Jasnaja Poljana*. Münster : Büchse der Pandora, 1980a (Bibliothek der Schulkritiker 1)
- [Blankertz 1980b] BLANKERTZ, Stefan: Tolstojs Beitrag zur Theorie und Praxis anarchistischer Pädagogik. In: *Die Schule von Jasnaja Poljana* [Blankertz 1980a], S. 5–18
- [Bloch 1964] BLOCH, Ernst: *Tübinger Einleitung in die Philosophie*. Bd. 1. 2. Aufl. Frankfurt a. M. : Suhrkamp, 1964
- [Bloom 1956] BLOOM, Benjamin S. (Hrsg.): *Taxonomy of Educational Objectives. The Classification of Educational Goals, Handbook I: Cognitive Domain*. New York : Longmans, Green, 1956
- [BMBF 2000] BMBF (Hrsg.): *Analyse und Evaluation der Softwareentwicklung in Deutschland*. Bundesministerium für Bildung und Forschung (BMBF), Dezember 2000. – Studie für das BMBF durchgeführt von GfK Marktforschung GmbH, Fraunhofer-Institut für Experimentelles Software Engineering (IESE), Fraunhofer-Institut für Systemtechnik und Innovationsforschung (ISI)

- [Boehm 1984] BOEHM, Barry: Software Engineering Economics. In: **[Broy und Denert 2002]**, S. 641–686. – zuerst veröffentlicht in: IEEE Transactions on Software Engineering, Vol. SE-10(1), pp. 4-21, 1984
- [Boehm 2002] BOEHM, Barry: Early Experiences in Software Economics. In: **[Broy und Denert 2002]**, S. 632–640. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Bönsch 1998] BÖNSCH, Manfred: Kriterien für guten Unterricht: Aufklärend – sinnstiftend – motivierend. In: *Neue Deutsche Schule* (1998), Dezember, Nr. 12, S. 27–29
- [Bortz und Döring 1995] BORTZ, Jürgen ; DÖRING, Nicola: *Forschungsmethoden und Evaluation*. 2. Aufl. Berlin : Springer, 1995
- [Bortz und Döring 2002] BORTZ, Jürgen ; DÖRING, Nicola: *Forschungsmethoden und Evaluation*. 3. Aufl. Berlin : Springer, 2002
- [Bosler 1992] BOSLER, Ulrich u. a. (Hrsg.): *Schulcomputerjahrbuch '93/94*. Hannover, Stuttgart : Metzler Schulbuch und B.G. Teubner, 1992
- [Böszörményi 1998] BÖSZÖRMÉNYI, László: Why Java is not my favorite first-course language. In: *Software – Concepts & Tools* 19 (1998), S. 141–145
- [Böszörményi 2001] BÖSZÖRMÉNYI, László: JAVA für Anfänger? Warum JAVA nicht meine Lieblingssprache für einen Anfängerkurs ist. In: *LOG IN* 21 (2001), Nr. 1, S. 14–19. – übersetzt von Andreas Schwill
- [Bothe 1998] BOTHE, Ingeborg: Mitarbeiterorientiertes Prozeßcontrolling in der betrieblichen Projektarbeit. In: PAUL, Hansjürgen (Hrsg.) ; MAUCHER, Irene (Hrsg.): *Integration von Mensch, Organisation und Technik: eine partielle Bilanz*. Gelsenkirchen : Graue Reihe, 1998, S. 43–52
- [Brauer 2001] BRAUER, Wilfried: Geschichtsbezogene Bemerkungen zur Frage: „Wie kommt es zur Informatik?“. In: **[Nake u. a. 2001]**, S. 17–20
- [Brauer und Brauer 1992] BRAUER, Wilfried ; BRAUER, Ute: Wissenschaftliche Herausforderungen für die Informatik: Änderung von Forschungszielen und Denkgewohnheiten. In: LANGENHEDER, Werner (Hrsg.) ; MÜLLER, G. (Hrsg.) ; SCHINZEL, Britta (Hrsg.): *Informatik cui bono?* Berlin : Springer, 1992 (Informatik aktuell Bd. 15), S. 11–19
- [Brauer und Brauer 1995] BRAUER, Wilfried ; BRAUER, Ute: Informatik – das neue Paradigma (Änderung von Forschungszielen und Denkgewohnheiten der Informatik). In: *LOG IN* 15 (1995), Nr. 4, S. 25–29
- [Brauer u. a. 1976] BRAUER, Wilfried ; CLAUS, Volker ; DEUSSEN, Peter ; JÜRGEN EICKEL (FEDERFÜHREND) ; HAACKE, Wolfhart ; HOSSEUS, Winfried ; KOSTER, Cornelis H. A. ; OLLESKY, Dieter ; WEINHART, Karl ; GESELLSCHAFT FÜR INFORMATIK E. V.: Zielsetzungen und Inhalte des Informatikunterrichts. In: *Zentralblatt für Didaktik der Mathematik, Stuttgart* 8 (1976), Nr. 1, S. 35–43
- [Brauer u. a. 1973] BRAUER, Wilfried ; HAACKE, Wolfhart ; MÜNCH, Siegfried ; GESELLSCHAFT FÜR MATHEMATIK UND DATENVERARBEITUNG MBH (Hrsg.) ; DAAD - DEUTSCHER AKADEMISCHER AUSTAUSCHDIENST (Hrsg.): *Studien- und Forschungsführer Informatik*. 1. Aufl. Sankt Augustin/Bonn : GMD, 1973
- [Brauer und Münch 1996] BRAUER, Wilfried ; MÜNCH, Siegfried: *Studien- und Forschungsführer Informatik*. 3. völlig neu bearbeitete Aufl. Berlin, Heidelberg : Springer, 1996

- [Braun und Fuchs 2001] BRAUN ; FUCHS: *Schreiben für elektronische Medien – „Als die Rechner laufen lernten“ – Die Anfänge der Informatik in Karlsruhe – ein Sprechtext von zwei Minuten Länge aus dem vorliegenden Interviewmaterial – Aufgaben für den 10.12.2001*. Universität Karlsruhe (TH) – Fakultät für Geistes- und Sozialwissenschaften – Lehrmaterialien online – Journalismus und Technik der elektronischen Medien. Dezember 2001. – <http://www.uni-karlsruhe.de/~ea18/journalismus/Interviewmaterial.doc> [http://www.uni-karlsruhe.de/~ea18/journalismus/fuchs\\_aufg10-12-01.doc](http://www.uni-karlsruhe.de/~ea18/journalismus/fuchs_aufg10-12-01.doc) – geprüft: 15. Juli 2002
- [Breiter und Kubicek 1998] BREITER, Andreas ; KUBICEK, Herbert: Schule am Netz – und dann? Informationstechnik-Management als kritischer Erfolgsfaktor für den Multimediaeinsatz in Schulen. In: KUBICEK, Herbert (Hrsg.) ; BRACZYK, Hans-Joachim (Hrsg.) ; KLUMPP, Dieter (Hrsg.) ; MÜLLER, Günter (Hrsg.) ; NEU, Werner (Hrsg.) ; RAUBOLD, Eckart (Hrsg.) ; ROSSNAGEL, Alexander (Hrsg.): *Lernort Multimedia: Jahrbuch Telekommunikation und Gesellschaft 1998* Bd. 6. Heidelberg : R. v. Decker, 1998. – <http://infosoc.informatik.uni-bremen.de/internet/schule/schuleamnetz/home.html> – geprüft: 27. August 2001, S. 120–129
- [Brinda 2001] BRINDA, Torsten: Einfluss fachwissenschaftlicher Erkenntnisse zum objektorientierten Modellieren auf die Gestaltung von Konzepten in der Didaktik der Informatik. In: [Keil-Slawik und Magenheim 2001],
- [Brinda u. a. 2002] BRINDA, Torsten ; HUMBERT, Ludger ; ALEX, Markus ; AZEM, Ahmad ; FRICKE, Tobias ; ISCAN, Hülya ; ROHR, Oliver ; ROSSKOPF, Michael ; SÖCHTIG, Gunnar ; STACHNIK, Adam ; VUKUSIC, Ivana ; WERNER, Thomas ; WOJCIECHOWSKI, Alexander: Entwicklung einer didaktischen Landkarte zum objektorientierten Modellieren. In: *LOG IN* 22 (2002). – im Druck
- [Brinda und Schubert 2001] BRINDA, Torsten ; SCHUBERT, Sigrid: Didaktisches System für objektorientiertes Modellieren / Fachbereich Informatik an der Universität Dortmund. 2001 ( 752). – Forschungsbericht. [http://didaktik-der-informatik.de/ddi\\_bib/forschung/berichte/](http://didaktik-der-informatik.de/ddi_bib/forschung/berichte/)
- [Brödner u. a. 1981] BRÖDNER, Peter ; KRÜGER, Detlef ; SENF, Bernd: *Der programmierte Kopf – Eine Sozialgeschichte der Datenverarbeitung*. Berlin : Verlag Klaus Wagenbach, 1981
- [Brooks 1986] BROOKS, Frederick P.: No Silver Bullet – Essence and Accidents of Software Engineering. In: [Kugler 1986], . – IEEE Computer
- [Brooks 1987] BROOKS, Frederick P.: No Silver Bullet – Essence and Accidents of Software Engineering. In: *IEEE Computer* 20 (1987), Nr. 1, S. 10–19
- [Brooks 2002] BROOKS, Frederick P.: The IBM Operating System /360. In: [Broy und Denert 2002], S. 170–178. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Brooks u. a. 1966] BROOKS, Frederick P. ; MEALY, George H. ; WITT, Bernie I. ; CLARK, William A.: The Functional Structure of OS/360. In: [Broy und Denert 2002], S. 179–229. – zuerst veröffentlicht in: *IBM Systems Journal* Vol. 5 (1), pp. 2-51, 1966
- [Broy und Denert 2002] BROY, Manfred (Hrsg.) ; DENERT, Ernst (Hrsg.): *Software Pioneers Contributions to Software Engineering*. Berlin : Springer, 2002 . – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Broy und Siedersleben 2002] BROY, Manfred ; SIEDERSLEBEN, Johannes: Objektorientierte Programmierung und Softwareentwicklung. In: *Informatik-Spektrum* 25 (2002), Februar, Nr. 1, S. 3–11

- [Bruhn u. a. 1988] BRUHN, Jörn ; BURKERT, Jürgen ; HOLLAND, Gerhard ; MAGENHEIM, Johann ; THIEMANN, Ullrich ; ZIMMERMANN, Rolf ; BURKERT, Jürgen (Hrsg.) ; GRIESEL, Heinz (Hrsg.) ; POSTEL, Helmut (Hrsg.): *Informatik heute*. Bd. 2: *Algorithmen und Datenstrukturen*. Hannover : Schroedel Schulbuchverlag, 1988
- [Bruner 1960] BRUNER, Jerome S.: *The process of education*. Cambridge, Mass. : Harvard University Press, 1960. – deutsche Übersetzung: *Der Prozeß der Erziehung*, Berlin, 1970
- [Bruner 1974] BRUNER, Jerome S.: *Entwurf einer Unterrichtstheorie*. Düsseldorf : Pädagogischer Verlag Schwann, 1974
- [Bruner 1975] BRUNER, Jerome S.: Der Akt der Entdeckung. In: NEBER, Horst (Hrsg.): *Entdeckendes Lernen*. Weinheim, 1975. – 2. Aufl.
- [Bruner 1980] BRUNER, Jerome S.: *Der Prozeß der Erziehung*. 5. Aufl. Düsseldorf : Pädagogischer Verlag Schwann, 1980. – 1. dt. Aufl. 1970
- [Brunnstein u. a. 1974] BRUNNSTEIN, Klaus (Hrsg.) ; HAEFNER, Klaus (Hrsg.) ; HÄNDLER, Wolfgang (Hrsg.) ; ACU – Arbeitskreis Computer-Unterstützter Unterricht (Veranst.): *Rechner-Gestützter Unterricht RGU '74 – Fachtagung, Hamburg, 12.-14. August 1974*. Berlin : Springer, 1974 (Lecture Notes in Computer Science 17)
- [Brunnstein u. a. 1976] BRUNNSTEIN, Klaus ; RUDOPH, Angelika ; UHLENHAUT, Giesela. *PASCAL-E – Eine Einführung in einen elementaren schulorientierten PASCAL-Subset*. Forschungs- und Entwicklungszentrum für objektivierte Lehr- und Lernverfahren (FEoLL) GmbH, Paderborn im Auftrag des Bundesministers für Forschung und Technologie. August 1976
- [Büttemeyer 1995] BÜTTEMEYER, Wilhelm: *Wissenschaftstheorie für Informatiker*. Heidelberg : Spektrum Hochschultaschenbuch, 1995
- [Cannon 2002] CANNON, Brett: *fleinput as a generator*. Python Cookbook. September 2002. – <http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/112506> – geprüft: 16. Dezember 2002
- [Capurro 1990] CAPURRO, Rafael: Ethik und Informatik. In: *Informatik Spektrum* 13 (1990), Dezember, Nr. 6, S. 311–320. – <http://www.capurro.de/antritt.htm> – geprüft: 13. April 2002
- [Cassens 2001] CASSENS, Jörg: Zum Verhältnis der Informatik zu anderen Fachdisziplinen. In: **[Nake u. a. 2001]**, S. 36–38
- [Chomsky 1959] CHOMSKY, Noam: A Review of B. F. Skinner's Verbal Behavior. In: *Language* 35 (1959), Nr. 1, S. 26–58. – <http://cogprints.soton.ac.uk/documents/disk0/00/00/11/48/cog00001148-00/chomsky.htm> – geprüft: 14. Juli 2002
- [Chomsky 1967] CHOMSKY, Noam: A Review of Skinner's Verbal Behavior. In: JAKOBOVITS, Leon A. (Hrsg.) ; MIRON, Murray S. (Hrsg.): *Readings in the Psychology of Language*. New Jersey : Prentice-Hall, 1967. – <http://cogprints.soton.ac.uk/documents/disk0/00/00/11/48/cog00001148-00/chomsky.htm> – geprüft: 14. Juli 2002, S. 142–143
- [Claus 1975] CLAUS, Volker: *Einführung in die Informatik*. Stuttgart : Teubner, 1975
- [Claus 1977] CLAUS, Volker: Informatik an der Schule: Begründungen und allgemeinbildender Kern. In: **[Bauersfeld u. a. 1977]**, S. 19–33. – Band I
- [Claus und Schwill 1984] CLAUS, Volker ; SCHWILL, Andreas: Die Wechselwirkungen zwischen Problemstellung, Programmiersprache und verwendeten Informatikmethoden am Beispiel der beiden Bundeswettbewerbe in Informatik. In: **[Arlt und Haefner 1984]**, S. 87–92

- [Claus und Schwill 1986] CLAUS, Volker ; SCHWILL, Andreas: Informatikkenntnisse von Jugendlichen, untersucht am Beispiel der drei Bundeswettbewerbe Informatik. In: *Informatik Spektrum* 9 (1986), Oktober, Nr. 5, S. 270–279
- [Claus und Schwill 2001] CLAUS, Volker ; SCHWILL, Andreas ; MEYERS LEXIKONREDATION (Hrsg.): *Duden „Informatik“: ein Fachlexikon für Studium und Praxis*. 3. Aufl. Mannheim, Leipzig, Wien, Zürich : Bibliographisches Institut, 2001
- [Combe und Buchen 1996] COMBE, Arno ; BUCHEN, Sylvia: *Belastung von Lehrerinnen und Lehrern: Fallstudien zur Bedeutung alltäglicher Handlungsabläufe an unterschiedlichen Schulformen*. 1. Aufl. Weinheim, München : Juventa Verlag, 1996 (Veröffentlichungen der Max-Traeger-Stiftung Bd. 25)
- [Coy 1992] COY, Wolfgang: Einleitung: Informatik – Eine Disziplin im Umbruch? In: [Coy u. a. 1992], S. 1–9
- [Coy 1997] COY, Wolfgang: Defining Discipline. In: [Freksa u. a. 1997], S. 21–35. – [http://waste.informatik.hu-berlin.de/Coy/Coy\\_Defining\\_Discipline.html](http://waste.informatik.hu-berlin.de/Coy/Coy_Defining_Discipline.html) – geprüft 11. Juli 2002
- [Coy 2001] COY, Wolfgang: Was ist Informatik? Zur Entstehung des Faches an den deutschen Universitäten. In: [Desel 2001], S. 1–22
- [Coy u. a. 1992] COY, Wolfgang (Hrsg.) ; NAKE, Frieder (Hrsg.) ; PFLÜGER, Jörg-Martin (Hrsg.) ; ROLF, Arno (Hrsg.) ; SEETZEN, Jürgen (Hrsg.) ; SIEFKES, Dirk (Hrsg.) ; STRANSFELD, Reinhard (Hrsg.): *Sichtweisen der Informatik*. Braunschweig : Vieweg Verlag, 1992 (Theorie der Informatik)
- [Cunningham und Beck 1986] CUNNINGHAM, Ward ; BECK, Kent: A Diagram for Object-Oriented Programs. In: *Proceedings OOPSLA '86, ACM SIGPLAN Notices* Bd. Vol. 21, 1986. – <http://c2.com/doc/oopsla89/paper.html> – geprüft: 15. Juli 2002, S. 361–367
- [Czischke u. a. 1999] CZISCHKE, Jürgen ; DICK, Georg ; HILDEBRECHT, Horst ; HUMBERT, Ludger ; UEDING, Werner ; WALLOS, Klaus ; LANDEsinstitut für Schule und Weiterbildung (Hrsg.): *Von Stiften und Mäusen*. 1. Aufl. Bönen : DruckVerlag Kettler GmbH, 1999
- [Dahl 2002] DAHL, Ole-Johan: The Roots of Object Orientation: The Simula Language. In: [Broy und Denert 2002], S. 78–90. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Daxner 1996] DAXNER, Michael: Evaluation als übergreifende Aufgabe im Hochschulbereich. In: SCHULZ, Reinhard (Hrsg.): *Verbesserung von Studium und Lehre*. Oldenburg : Bibliotheks- und Informationssystem der Universität, 1996. – <http://www.bis.uni-oldenburg.de/bisverlag/schver96/kap1.pdf> geprüft am 15. April 2001, S. 9–17
- [Decker 1998] DECKER, Hans: Universität Dortmund. In: [Görke 1998], S. 31–37
- [Deckers 1996] DECKERS, Joachim: *Das Internet: Medium und Inhalt für den Informatikunterricht*. Mai 1996. – 2. Staatsarbeit für das Lehramt für die Sekundarstufe II, Informatik <http://www.deckers-online.de/examen/> – geprüft: 16. Dezember 2002
- [DeMarco 1978] DEMARCO, Tom: Structured Analysis and System Specification. In: [Broy und Denert 2002], S. 529–560. – zuerst veröffentlicht in: Tom DeMarco, Structured Analysis and System Specification, Yourdon, New York, pp. 1-7 and 37-44, 1978
- [DeMarco 2002] DEMARCO, Tom: Structured Analysis: Beginning of a New Discipline. In: [Broy und Denert 2002], S. 520–527. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002

- [Denning 1989] DENNING, Peter J. (Hrsg.): *A debate on teaching computing science*. ACM, December 1989. – CACM 32 (1989), Nr. 12
- [Denning 1999a] DENNING, Peter J.: Editor's introductory essay: Computers and Human Aspiration. In: [Denning 1999b], S. xi–xviii. – <http://cne.gmu.edu/pjd/PUBS/thenow99.pdf> – geprüft: 14. Juli 2002
- [Denning 1999b] DENNING, Peter J. (Hrsg.): *Talking back to the Machine: Computers and Human Aspiration*. New York : Copernicus Books (Springer), May 1999b
- [Denning u. a. 1989] DENNING, Peter J. ; COMER, Douglas E. ; GRIES, David ; MULDER, Michael C. ; TUCKER, Allen ; TURNER, A. J. ; YOUNG, Paul R.: Computing as a discipline. In: CACM 32 (1989), January, Nr. 1, S. 9–23
- [Denning und Metcalfe 1997] DENNING, Peter J. (Hrsg.) ; METCALFE, Robert M. (Hrsg.) ; ACM (Veranst.): *Beyond Calculation: The Next 50 Years of Computing*. New York : Copernicus Books (Springer), May 1997. – First softcover printing 1998
- [Der Kultusminister des Landes Nordrhein-Westfalen 1985] DER KULTUSMINISTER DES LANDES NORDRHEIN-WESTFALEN (Hrsg.): *Neue Informations- und Kommunikationstechnologien in der Schule – Rahmenkonzept*. Greven Verlag, 1985 (Strukturförderung im Bildungswesen des Landes Nordrhein-Westfalen)
- [Desel 2001] DESEL, Jörg (Hrsg.): *Das ist Informatik*. Berlin, Heidelberg, New York, Barcelona, Hongkong, London, Mailand, Paris, Singapur, Tokio : Springer-Verlag, Januar 2001
- [Deutsch und Goldberg 1991] DEUTSCH, L. P. ; GOLDBERG, Adele: Smalltalk: Yesterday, Today, and Tomorrow. A look back and a look ahead at this innovative programming language – first featured 10 years ago in BYTE. In: *Byte* 16 (1991), August, Nr. 8. – <http://www.byte.com/art/9608/sec4/art1.htm> – geprüft: 11. August 2002
- [Dijkstra 1969] DIJKSTRA, Edsger W.: *Structured programming*. August 1969. – circulated privately – <http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD268.PDF> – geprüft: 12. August 2002
- [Dijkstra 1970] DIJKSTRA, Edsger W.: *Notes on Structured Programming*. April 1970. – EWD249 – circulated privately – <http://www.cs.utexas.edu/users/EWD/ewd02xx/EWD249.PDF> – geprüft: 12. August 2002
- [Dijkstra 1989] DIJKSTRA, Edsger W.: Dijkstra's Reply To Comments. In: [Denning 1989], S. 1414. – CACM 32 (1989), Nr. 12
- [Dijkstra 2001] DIJKSTRA, Edsger W.: The End of Computing Science? In: CACM 44 (2001), March, Nr. 3, S. 92
- [Dodds 2001] DODDS, Philip (Hrsg.): *Advanced Distributed Learning Initiative – Sharable Content Object Reference Model. The SCORM Overview*. US : Advanced Distributed Learning (ADL), October 2001. – Version 1.2. [http://www.adlnet.org/ADLDOCS/Documents/SCORM\\_1.2\\_Overview.pdf](http://www.adlnet.org/ADLDOCS/Documents/SCORM_1.2_Overview.pdf) – geprüft: 6. Oktober 2002
- [Dostal 2000] DOSTAL, Werner: Die Informatisierung der Arbeitswelt – Ein erster Blick auf die Ergebnisse der BIBB/IAB-Erhebung. In: DOSTAL, Werner (Hrsg.) ; JANSEN, Rolf (Hrsg.) ; PARMENTIER, Klaus (Hrsg.): *Wandel der Erwerbsarbeit: Arbeitssituation, Informatisierung, berufliche Mobilität und Weiterbildung*. Nürnberg : Bundesanstalt für Arbeit, 2000, S. 151–167
- [Dostal 2001] DOSTAL, Werner: Turbulenzen im IT-Arbeitsmarkt. In: *Informatik-Spektrum* 24 (2001), August, Nr. 4, S. 207–217
- [Dostal 2002] DOSTAL, Werner: IT-Arbeitsmarkt – Katastrophe oder Normalisierung? In: *Informatik Spektrum* 25 (2002), Oktober, Nr. 5, S. 341–348

- [Dreyfus 1989] DREYFUS, Hubert L.: *Was Computer nicht können: die Grenzen künstlicher Intelligenz*. Taschenbuchausgabe. Frankfurt a. M. : Athenäum, Februar 1989 (athenäum's taschenbücher 123). – Aus dem Amerikanischen von Robin Cackett, Originaltitel: *What computers can't do - The Limits of Artificial Intelligence*, 1972, 1979, Harper & Row Publishers, New York
- [Dreyfus und Dreyfus 1987] DREYFUS, Hubert L. ; DREYFUS, Stuart E. ; MOOS, Ludwig (Hrsg.) ; WAFENDER, Manfred (Hrsg.): *Künstliche Intelligenz - Von den Grenzen der Denkmachine und dem Wert der Intuition*. Taschenbuchausgabe. Reinbek : Rowohlt, April 1987 (rororo computer 8144). – Aus dem Amerikanischen von Michael Mutz, Originaltitel: *Mind over Machine*, 1986, The Free Press, New York
- [Dubs 1995] DUBS, Rolf: *Lehrerverhalten. Ein Beitrag zur Interaktion von Lehrenden und Lernenden im Unterricht*. Zürich : Verlag des Schweizerischen Kaufmännischen Verbandes – SKV, 1995
- [Eberle 1996] EBERLE, Franz ; WETTSTEIN, Emil (Hrsg.) ; WEIBEL, Walter (Hrsg.) ; GONON, Philipp (Hrsg.): *Didaktik der Informatik bzw. einer informations- und kommunikationstechnologischen Bildung auf der Sekundarstufe II – Ziele und Inhalte, Bezug zu anderen Fächern sowie unterrichtspraktische Handlungsempfehlungen*. 1. Aufl. Aarau : Verlag Sauerländer, 1996 (Pädagogik bei Sauerländer: Dokumentation und Materialien 24)
- [Engbring 1995] ENGBRING, Dieter: Kultur- und technikgeschichtlich begründete Bildungswerte der Informatik. In: [Schubert 1995], S. 68–77
- [Eulenhöfer u. a. 1997a] EULENHÖFER, Peter ; SIEFKES, Dirk ; STACH, Heike: Informatics as Cultural Development – Case Studies from the Social History of Information Technology / Technical University Berlin – Department of Computer Science. 1997 ( No. 97-2). – Research Report <http://tal.cs.tu-berlin.de/RoteReihe/RR97-02.rtf> – geprüft: 14. April 2002
- [Eulenhöfer u. a. 1997b] EULENHÖFER, Peter ; SIEFKES, Dirk ; STACH, Heike: Informatics as Cultural Development. The Creation of Programming Languages. In: [Eulenhöfer u. a. 1997a], S. 16–26
- [Faulstich-Wieland und Nyssen 1998] FAULSTICH-WIELAND, Hannelore ; NYSSSEN, Elke: Geschlechterverhältnisse im Bildungssystem – Eine Zwischenbilanz. In: ROLFF, Hans-Günter (Hrsg.) ; BAUER, K.-O. (Hrsg.) ; KLEMM, Klaus (Hrsg.) ; PFEIFFER, Hermann (Hrsg.): *Jahrbuch der Schulentwicklung*. Weinheim : Juventa, 1998. – <http://www.erzwiss.uni-hamburg.de:80/Personal/Liesner/Lehre/SoSe02/FaulstichWieland-Nyssen.doc> – geprüft: 2. Dezember 2002, S. 163–199
- [Fend 1974] FEND, Helmut: *Gesellschaftliche Bedingungen schulischer Sozialisation*. Weinheim : Beltz, 1974
- [Feuck 2001] FEUCK, Jörg: Big Point des Kleinen in der Kultusminister-Liga. Das Saarland stellt im Sommer flächendeckend auf das achtjährige Gymnasium um. In: *Frankfurter Rundschau* – 26. April (2001)
- [Fischer u. a. 2000] FISCHER, Arthur ; FRITZSCHE, Yvonne ; FUCHS-HEINRITZ, Werner ; MÜNCHMEIER, Richard ; SHELL, Deutsche (Hrsg.): *Jugend 2000 – 13. Shell Jugendstudie*. Opladen : Leske + Budrich, 2000. – 2 Bände
- [Fischer 1996] FISCHER, Hans R.: Vorwort. In: *Wege des Wissens. Konstruktivistische Erkundungen durch unser Denken* [von Glaserfeld 1997b], S. 7–9. – Übersetzungen aus dem Englischen von Wolfram Köck
- [Fischer 2001] FISCHER, Martin: Informatiker erzeugen Welten. In: [Nake u. a. 2001], S. 39–40
- [Fischer 1984] FISCHER, Roland: Unterricht als Prozeß von der Befreiung vom Gegenstand – Visionen eines neuen Mathematikunterrichts. In: *Journal für Mathematik-Didaktik (JMD)* 1 (1984), S. 51–85

- [Floyd 1992] FLOYD, Christiane: Human Questions in Computer Science. In: FLOYD, Christiane (Hrsg.) ; ZÜLLIGHOVEN, Heinz (Hrsg.) ; BUDDE, Reinhard (Hrsg.) ; KEIL-SLAWIK, Reinhard (Hrsg.): *Software Development and Reality Construction*. Berlin : Springer, 1992, S. 15–27
- [Floyd 1994] FLOYD, Christiane: Software-Engineering – und dann? In: *Informatik Spektrum* 17 (1994), Februar, Nr. 1, S. 29–37
- [Floyd 1997] FLOYD, Christiane: Autooperationale Form und situiertes Handeln. In: HUBIG, C. (Hrsg.): *Cognitio Humana - XVII. Deutscher Kongreß für Philosophie (Sept. 1996)*, Akademie Verlag, 1997, S. 237–252
- [Floyd 2001] FLOYD, Christiane: *Informatik – Mensch – Gesellschaft 1. Prüfungsunterlagen*. Universität Hamburg – Fachbereich Informatik, Oktober 2001. – zugl. Informatik – eine Standortbestimmung – Hamburg, September 1998 von C. Floyd und R. Klischewski
- [Floyd und Klischewski 1998] FLOYD, Christiane ; KLISCHEWSKI, Ralf: Modellierung – ein Handgriff zur Wirklichkeit. Zur sozialen Konstruktion und Wirksamkeit von Informatik-Modellen. In: POHL, Klaus (Hrsg.) ; SCHÜRR, Andy (Hrsg.) ; VOSSEN, Gottfried (Hrsg.): *Modellierung 98 – Proceedings*. Universität Münster : Institut für angewandte Mathematik und Informatik, März 1998 (Bericht 6/98-I). – <http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-9/> – geprüft: 15. Juli 2002, S. 21–26
- [Floyd u. a. 1987] FLOYD, Christiane ; MEHL, Wolf-Michael ; REISIN, Fanny-Michaela ; SCHMIDT, Gerhard ; WOLF, Gregor ; MAGS (Hrsg.): *SCANORAMA – Methoden, Konzepte, Realisierungsbedingungen und Ergebnisse von Initiativen alternativer Softwareentwicklung und -gestaltung in Skandinavien*. MAGS – Ministerium für Arbeit, Gesundheit und Soziales der Landes Nordrhein-Westfalen, 1987 (Werkstattberichte 30). – Landesprogramm „Mensch + Technik – Sozialverträgliche Technikgestaltung“
- [Frank 1969] FRANK, Helmar: *Kybernetische Grundlagen der Pädagogik*. 2. völlig überarbeitete und wesentlich erweiterte Aufl. Stuttgart, Berlin, Köln, Mainz : Kohlhammer, 1969. – 2 Bde. (erste Auflage 1962)
- [Freire 1977] FREIRE, Paulo: *Erziehung als Praxis der Freiheit*. Reinbek : Rowohlt, März 1977
- [Freksa u. a. 1997] FREKSA, Christian (Hrsg.) ; JANTZEN, Matthias (Hrsg.) ; VALK, Rüdiger (Hrsg.): *Foundations of Computer Science: Potential – Theory – Cognition, to Wilfried Brauer on the occasion of his sixtieth birthday*. Heidelberg, Berlin, New York : Springer, 1997 (Lecture Notes in Computer Science 1337)
- [Frey u. a. 2001] FREY, Elke ; HUBWIESER, Peter ; HUMBERT, Ludger ; SCHUBERT, Sigrid ; VOSS, Siglinde: Erste Ergebnisse aus dem Informatik-Anfangsunterricht in den bayerischen Schulversuchen. In: *LOG IN* 21 (2001), Nr. 1, S. 25–37. – [http://www.ham.nw.schule.de/pub/nj\\_bscw.cgi/d29166/Log\\_In\\_Informatik-Anfangsunterricht.pdf](http://www.ham.nw.schule.de/pub/nj_bscw.cgi/d29166/Log_In_Informatik-Anfangsunterricht.pdf) – geprüft: 29. November 2002
- [Frey 1998] FREY, Karl: *Die Projektmethode. Der Weg zum bildenden Tun*. 8. überarbeitete Aufl. Weinheim, Basel : Beltz Verlag, 1998 (Beltz Pädagogik). – Unter Mitarbeit von Ulrich Schäfer, Michael Knoll, Angela Frey-Eiling, Ulrich Heimlich und Klaus Mie
- [Friedrich 1995] FRIEDRICH, Steffen: Standortbestimmung der Informatik in der Schule. In: **[Schubert 1995]**, S. 33–39
- [Friedrich 1998] FRIEDRICH, Steffen: Schulinformatik – eine endlose Debatte. In: **[Koerber und Peters 1998]**, S. 75–88. – <http://log-in.fachdid.fu-berlin.de/IBiD/index.html> – geprüft 17. Mai 2002
- [Fuhr 2000] FUHR, Norbert. *Informationssysteme Stammvorlesung im WS 99/00 (IR-Teil)*. [http://ls6-www.informatik.uni-dortmund.de/ir/teaching/lectures/is\\_ws99-00/folien/irskall.ps.gz](http://ls6-www.informatik.uni-dortmund.de/ir/teaching/lectures/is_ws99-00/folien/irskall.ps.gz) – geprüft: 25. September 2002. Januar 2000

- [Funken u. a. 1996] FUNKEN, Christiane ; HAMMERICH, Kurt ; SCHINZEL, Britta: *Geschlecht, Informatik und Schule oder: Wie Ungleichheit der Geschlechter durch Koedukation neu organisiert wird*. 1. Aufl. Sankt Augustin : Academia Verlag, 1996
- [Gagné 1962] GAGNÉ, Robert M.: Military training and principles of learning. In: *American Psychologist* 17 (1962), S. 263–276
- [Gallin und Ruf 1990] GALLIN, Peter ; RUF, Urs: *Sprache und Mathematik in der Schule. Auf eigenen Wegen zur Fachkompetenz. Illustriert mit sechzehn Szenen aus der Biographie von Lernenden*. Zürich : Verlag Lehrerinnen und Lehrer, 1990
- [Gallin und Ruf 1998] GALLIN, Peter ; RUF, Urs: *Sprache und Mathematik in der Schule*. Seelze : Friedrich Verlag, 1998
- [Gamm 1983] GAMM, Hans-Jochen: *Materialistisches Denken und pädagogisches Handeln*. Frankfurt a. M. : Campus Verlag, 1983
- [Gamma u. a. 1996] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; JOHN, Vlissides: *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. 1. Aufl. Bonn : Addison-Wesley, 1996. – Design Patterns, 1995, Deutsche Übersetzung von Dirk Riehle
- [Gaudig 1921] GAUDIG, Hugo: Kritik der Formalstufen – Die Methode der freien geistigen Arbeit. In: [Geißler 1970], S. 35–47. – aus: *Didaktische Präludien*, 2. Aufl. Leipzig, Berlin (Teubner), 1921, S. 1ff
- [Geißler 1970] GEISSLER, Georg (Hrsg.): *Das Problem der Unterrichtsmethode in der pädagogischen Bewegung*. Weinheim : Beltz Verlag, 1970 (Kleine Pädagogische Texte 18). – 1. Aufl. 1952
- [Genrich 1975a] GENRICH, Hartmann J.: Belästigung des Menschen durch Computer. In: MÜHLBACHER, Jörg (Hrsg.): *GI – 5. Jahrestagung, Dortmund, 8.-10. Oktober 1975*. Heidelberg : Springer, Oktober 1975a (Lecture Notes in Computer Science 34), S. 94–105
- [Genrich 1975b] GENRICH, Hartmann J.: Belästigung des Menschen durch Computer. In: *Der GMD-Spiegel* (1975), Dezember, Nr. 5, S. 32–45
- [Gesellschaft für Informatik 2000] GESELLSCHAFT FÜR INFORMATIK: Empfehlung der Gesellschaft für Informatik e.V. für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. In: *Informatik Spektrum* 23 (2000), Dezember, Nr. 6, S. 378–382
- [Giegerich 1989] GIEGERICH, Robert: *Grundlagen der Informatik für Naturwissenschaftler und Ingenieure*. 2. Aufl. Dortmund : Universität Dortmund, 1989. – Skriptum zur Vorlesung Technische Informatik im Wintersemester 1989/90
- [Giloi 1990] GILOI, Wolfgang K.: Konrad Zuses Plankalkül als Vorläufer moderner Programmiermodelle / ZIB – Konrad-Zuse-Zentrum für Informationstechnik Berlin. 1990 ( TR 90-13). – Technical Report. <ftp://ftp.zib.de/pub/zib-publications/reports/TR-90-13.ps.Z> – geprüft: 20. Mai 2002
- [von Glasersfeld 1989] GLASERSFELD, Ernst von: Kognition, Konstruktion des Wissens und Unterricht. In: *Wege des Wissens. Konstruktivistische Erkundungen durch unser Denken* [von Glasersfeld 1997b], S. 172–197. – überarbeiteter Fassung des Beitrages „Cognition, Construction of Knowledge, and Teaching.“ In: *Synthese* 1989, 80(1), pp. 121-140.
- [von Glasersfeld 1995] GLASERSFELD, Ernst von: Aspekte einer konstruktivistischen Didaktik. In: *Wege des Wissens. Konstruktivistische Erkundungen durch unser Denken* [von Glasersfeld 1997b], S. 165–171. – Zuerst erschienen in [LSW 1995a]

- [von Glasersfeld 1997a] GLASERSFELD, Ernst von: *Radikaler Konstruktivismus. Ideen, Ergebnisse, Probleme*. Frankfurt a. M. : Suhrkamp, 1997a (stw 1326). – Titel der Originalausgabe: *Radical Constructivism. A Way of Knowing and Learning* – London : The Falmer Press 1995. Übersetzung aus dem Englischen von Wolfram Köck
- [von Glasersfeld 1997b] GLASERSFELD, Ernst von: *Wege des Wissens. Konstruktivistische Erkundungen durch unser Denken*. Heidelberg : Carl-Auer-Systeme Verlag, 1997b. – Übersetzungen aus dem Englischen von Wolfram Köck
- [Glöckel 1996] GLÖCKEL, Hans: *Vom Unterricht*. 3. überarbeitete und ergänzte Aufl. Bad Heilbrunn : Klinkhardt, 1996
- [Goldberg und Robson 1983] GOLDBERG, Adele ; ROBSON, Dave: *Smalltalk-80: The Language and Its Implementation*. Addison Wesley, 1983. – [http://users.ipa.net/~dwighth/smalltalk/bluebook/bluebook\\_imp\\_toc.html](http://users.ipa.net/~dwighth/smalltalk/bluebook/bluebook_imp_toc.html) – geprüft: 16. Dezember 2002
- [Good 2002] GOOD, Tom: *Walk a directory tree using a generator*. Python Cookbook. May 2002. – <http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/105873> – geprüft: 16. Dezember 2002
- [Goos 1979] GOOS, Gerhard: *Editorial Informatik an der Schule?* In: *Informatik Spektrum* 2 (1979), Februar, Nr. 1, S. 1–3
- [Görke 1998] GÖRKE, Winfried (Hrsg.) ; Fakultätentag Informatik (Veranst.): *25 Jahre Fakultätentag Informatik – 1973–1998*. Digitale Fassung vom Dezember 2000 erhältlich unter [http://goethe.ira.uka.de/~goerke/Brosch\\_Faktag\\_erg.pdf](http://goethe.ira.uka.de/~goerke/Brosch_Faktag_erg.pdf) – geprüft: 13. Mai 2002 : Fakultätentag Informatik, 1998
- [Gorny 1984] GORNY, Peter: Informatik in Schule und Ausbildung im internationalen Vergleich. In: **[Arlt und Haefner 1984]**, S. 37–46
- [Gorny 1998] GORNY, Peter: Didaktisches Design telematik-gestützter Lernsoftware. In: **[Koerber und Peters 1998]**, S. 127–155. – <http://www-cg-hci.informatik.uni-oldenburg.de/resources/DidDesign.pdf> – geprüft: 14. Juli 2002
- [Grell und Grell 1996] GRELL, Joachim ; GRELL, Monika: *Unterrichtsrezepte*. 11. Aufl. Weinheim : Beltz Verlag, 1996 (Beltz Grüne Reihe)
- [Gruska und Vollmar 1997] GRUSKA, Jozef ; VOLLMAR, Roland: Towards Adjusting Informatics Education to Information Era. In: **[Freksa u. a. 1997]**, S. 49–67
- [Gudjons 1986] GUDJONS, Herbert: *Handlungsorientiert Lehren und Lernen. Projektarbeit und Schüleraktivität*. Bad Heilbrunn : Klinkhardt, 1986
- [Gudjons 2001] GUDJONS, Herbert: *Handlungsorientiert lehren und lernen. Schüleraktivierung – Selbsttätigkeit – Projektarbeit*. 6. überarb. und erw. Aufl. Bad Heilbrunn : Klinkhardt, 2001 (Erziehen und Unterrichten in der Schule)
- [Guzdial und Soloway 2002] GUZDIAL, Mark ; SOLOWAY, Elliot: Teaching the Nintendo generation to program. In: *CACM* 45 (2002), April, Nr. 4, S. 17–21
- [Häfeli und Lamprecht 2001] HÄFELI, Christian ; LAMPRECHT, Reto: *Interactive Learning Components for the Study of Finite Automata, Diplomarbeit*. Zürich : ETH, March 2001. – <http://www.tedu.ethz.ch/exorciser/docu/ifa01.pdf> – geprüft: 16. Dezember 2002
- [Haller 1990] HALLER, M. (Hrsg.): *Weizenbaum contra Haefner - Sind Computer die besseren Menschen?* Zürich : Pendo, 1990

- [Händlykken und Nygaard 1981] HÄNDLYKKEN, P. ; NYGAARD, Kristen: The DELTA System Description Language – Motivation, Main Concepts and Experience from Use. In: HÜNKE, H. (Hrsg.): *Software Engineering Environments Proceedings of the Symposium held in Lahnstein, Germany, June 16-20, 1980*. Amsterdam : North-Holland, 1981
- [Harbison 1990] HARBISON, Sam: Modula-3. In: *Byte* 15 (1990), November, Nr. 12, S. 385–392
- [Harrer und Schneider 2002] HARRER, Andreas ; SCHNEIDER, Markus: Didaktische Betrachtungen zur Unterrichtung von Software-Mustern im Hochschulbereich. In: [Schubert u. a. 2002], S. 67–76
- [Hartmann und Nievergelt 2002] HARTMANN, Werner ; NIEVERGELT, Jürg: Informatik und Bildung zwischen Wandel und Beständigkeit. In: *Informatik Spektrum* 25 (2002), Dezember, Nr. 6, S. 465–476
- [Hauf-Tulodziecki u. a. 1999] HAUF-TULODZIECKI, Annemarie ; BARTSCH, Paul D. ; BECKER, Karl-Heinz ; HERZIG, Bardo ; LEHMANN, Gabriele ; MAGENHEIM, Johannes ; SCHELHOWE, Heidi ; SIEGEL, Christian ; WAGNER, Wolf-Rüdiger: *Informatische Bildung und Medienerziehung. Empfehlung der Gesellschaft für Informatik e.V. erarbeitet von einem Arbeitskreis des Fachausschusses „Informatische Bildung in Schulen“ (7.3)*. Oktober 1999. – [http://www.gi-ev.de/informatik/publikationen/empfehlung\\_991206.shtml](http://www.gi-ev.de/informatik/publikationen/empfehlung_991206.shtml) – geprüft: 16. Dezember 2002
- [Häußler u. a. 1998] HÄUSSLER, Peter (Hrsg.) ; BÜNDER, Wolfgang (Hrsg.) ; DUIT, Reinders (Hrsg.) ; GRÄBER, Wolfgang (Hrsg.) ; MAYER, Jürgen (Hrsg.): *Naturwissenschaftsdidaktische Forschung - Perspektiven für die Unterrichtspraxis*. Kiel : Institut für die Pädagogik der Naturwissenschaften, 1998
- [Hechenleitner 2001] HECHENLEITNER, Andrea: *Schulversuch „Europäisches Gymnasium“*. November 2001. – <http://www.isb.bayern.de/gym/informat/schulver/lp-egy.htm> – geprüft: 27. November 2002
- [Heimann u. a. 1970] HEIMANN, Paul ; OTTO, Gunter ; SCHULZ, Wolfgang: *Auswahl Reihe B*. Bd. 1/2: *Unterricht: Analyse und Planung*. 5. Aufl. Hannover : Schroedel-Verlag, 1970
- [Hein 1999] HEIN, Hans-Werner: „Big Brother is scanning you“. In: *Spektrum der Wissenschaft* (1999), März, Nr. 3, S. 106–109
- [Herbart 1913] HERBART, Johann F.: Begriff der Vielseitigkeit – Stufen des Unterrichts – Gang des Unterrichts. In: [Geißler 1970], S. 19–28. – aus: Willmann-Fritzsch: Joh. Friedr. Herbarts Päd. Schriften – Allgemeine Pädagogik. 2. Buch, 3. Ausgabe. Osterwieck 1913ff.
- [Hermann 1980] HERMANN, Ursula: *Großes Sprach- und Fremdwörterbuch. Rechtschreibung, Fremdwörter, Grammatik*. München : Lexikographisches Institut, 1980. – Lizenzausgabe für die Büchergilde Gutenberg
- [Herzig 2001] HERZIG, Bardo: Medienbildung und Informatik – Zur Fundierung einer integrativen Medienbildungstheorie. In: [Keil-Slawik und Magenheimer 2001], S. 107–119
- [Hesse 1943] HESSE, Hermann: *Das Glasperlenspiel*. 1972 – Jubiläumsausgabe zum hundersten Geburtstag von Hermann Hesse. Frankfurt a. M. : Suhrkamp Verlag, 1943. – Originalausgabe: 1943 Fretz & Wasmuth AG, Zürich
- [Hessisches Kultusministerium 2002] HESSISCHES KULTUSMINISTERIUM (Hrsg.): *Lehrplan Informatik – Gymnasialer Bildungsgang Jahrgangsstufe 11 bis 13 Stand: 03.06.2002*. Wiesbaden : KM Hessen, Juni 2002. – [http://www.hessisches-kultusministerium.de/downloads/lehrplngym/Informatik\\_SEK\\_II.pdf](http://www.hessisches-kultusministerium.de/downloads/lehrplngym/Informatik_SEK_II.pdf) – geprüft: 16. November 2002
- [Hildebrandt 2002] HILDEBRANDT: *Veranstaltung zur Bildungsgangdidaktik – Folienskript*. Juni 2002. – – nach [Effe-Stumpf, Gertrud u. a.] <http://www.uni-duisburg.de/~hd292hi/Seminare/GesamtfolienBildungsgangdidaktik.pdf> – geprüft: 11. September 2002

- [Hilger 2002] HILGER, Stefan: *Einführung in die Physik Didaktik*. März 2002. – <http://mathsrv.ku-eichstaett.de/MGF/homes/didphy/skripten/DID.pdf> – geprüft: 2. September 2002
- [Hinck u. a. 2001] HINCK, Daniela ; KÖHLER, Michael ; LANGER, Roman ; MOLDT, Daniel ; RÖLKE, Heiko: *Organisation etablierter Machtzentren: Modellierungen und Reanalysen zu Norbert Elias*. Hamburg : Fachbereich Informatik und Institut für Soziologie der Universität, September 2001 (Arbeitsberichte des Forschungsprogramms Agieren in sozialen Kontexten FBI-HH-306/01). – <http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/publ/elias.ps> – geprüft: 31. Juli 2002
- [Hopcroft und Ullmann 1988] HOPCROFT, John E. ; ULLMANN, Jeffrey D.: *Einführung in die Automaten-theorie, Formale Sprachen und Komplexitätstheorie*. Bonn : Addison-Wesley, 1988. – englische Originalausgabe: Introduction to automata theory languages and computation; 1979 Addison-Wesley
- [Hoppe und Luther 1997] HOPPE, Heinz U. (Hrsg.) ; LUTHER, Wolfram (Hrsg.): *Informatik und Lernen in der Informationsgesellschaft*. Berlin, Heidelberg : Springer, September 1997 (Informatik aktuell)
- [Hoppe und Luther 1996a] HOPPE, Heinz U. ; LUTHER, Wolfram J.: Informatik und Schule – Ein Fach im Spiegel neuer Entwicklungen der Fachdidaktik. In: *LOG IN* 16 (1996), Nr. 1, S. 8–14
- [Hoppe und Luther 1996b] HOPPE, Heinz U. ; LUTHER, Wolfram J.: Informatik und Schule im Spiegel neuer Entwicklungen in der Fachdidaktik / Institut für Informatik. Gerhard-Mercator-Universität-Gesamthochschule, Januar 1996b ( Duisburg). – Schriftenreihe Informatik – Bericht Nr. SI-14
- [Hubwieser 1997] HUBWIESER, Peter: Wie soll der Unterricht ablaufen? In: [Hoppe und Luther 1997], S. 216
- [Hubwieser 1999] HUBWIESER, Peter: Informatik als Pflichtfach an bayerischen Gymnasien. In: [Schwill 1999], S. 165–174
- [Hubwieser 2000] HUBWIESER, Peter: *Didaktik der Informatik – Grundlagen, Konzepte, Beispiele*. 1. Aufl. Berlin : Springer, April 2000
- [Hubwieser und Broy 1996] HUBWIESER, Peter ; BROY, Manfred: Der informationszentrierte Ansatz - Ein Vorschlag für eine zeitgemäße Form des Informatikunterrichtes am Gymnasium / Technische Universität München - Fakultät für Informatik. München, Mai 1996 ( TUM-I9624). – Forschungsbericht. <http://wwwbib.informatik.tu-muenchen.de/infberichte/1996/TUM-I9624.ps> – geprüft: 21. Mai 2002
- [Hubwieser und Broy 1997] HUBWIESER, Peter ; BROY, Manfred: Grundlegende Konzepte von Informations- und Kommunikationssystemen für den Informatikunterricht. In: [Hoppe und Luther 1997], S. 40–50
- [Hubwieser u. a. 2001a] HUBWIESER, Peter ; FISCHER, Rudi ; HEUSSER, Theo ; JANTZEN, Fried-Thorsten ; LEIPHOLZ-SCHUMACHER, Barbara ; MAYRHOFER, Franz ; MODELL, Rüdiger ; PUHLMANN, Hermann ; RÖHNER, Gerhard ; SCHOTT, Walter ; STIMM, Hermann ; TIMMERMANN, Bettina ; WIEDEMANN, Albert ; WINHARD, Ferdinand ; WITT, Hans: *Empfehlungen der Gesellschaft für Informatik e.V. zur Planung und Betreuung von Rechnersystemen an Schulen*. Erarbeitet von der GI-Fachgruppe 7.3.1 – Informatiklehrer und -lehrerinnen – Beilage zur LOG IN 21(2001) Heft 1. Juni 2001a. – <http://nw.schule.de/gi/EmpfehlungenSysBetr.PDF> – geprüft: 29. Dezember 2002
- [Hubwieser u. a. 2001b] HUBWIESER, Peter ; HUMBERT, Ludger ; SCHUBERT, Sigrid: Evaluation von Informatikunterricht. In: [Keil-Slawik und Magenheimer 2001], S. 213–215
- [Humbert 1999] HUMBERT, Ludger: Grundkonzepte der Informatik und ihre Umsetzung im Informatikunterricht. In: [Schwill 1999], S. 175–189
- [Humbert 2000a] HUMBERT, Ludger: Automatisierte Aktualisierung von Webseiten. In: *LOG IN* 20 (2000), Nr. 3/4, S. 63–68

- [Humbert 2000b] HUMBERT, Ludger: Umsetzung von Grundkonzepten der Informatik zur fachlichen Orientierung im Informatikunterricht. In: *informatica didactica* (2000), Juli, Nr. 1. – Herausgeber: Sigrid Schubert, Andreas Schwill <http://didaktik.cs.uni-potsdam.de/InformaticaDidactica/Issue1/Humbert> Ausgewählte Beiträge zur Tagung INFOS99 – 8. GI-Fachtagung Informatik und Schule
- [Humbert 2001a] HUMBERT, Ludger: Informatik lehren – zeitgemäße Ansätze zur nachhaltigen Qualifikation aller Schülerinnen. In: [Keil-Slawik und Magenheim 2001], S. 121–132. – [http://www.ham.nw.schule.de/pub/bscw.cgi/d23843/INFOS\\_2001\\_Informatik-lehren.pdf](http://www.ham.nw.schule.de/pub/bscw.cgi/d23843/INFOS_2001_Informatik-lehren.pdf) – geprüft: 16. Dezember 2002
- [Humbert 2001b] HUMBERT, Ludger: Interviews mit Informatik-Lehrkräften. In: *LOG IN* 21 (2001), Nr. 3/4, S. 51–53
- [Humbert 2001c] HUMBERT, Ludger: Theoretischer und empirischer Vergleich zum Bild der Wissenschaft Informatik in der Schule. In: *informatica didactica* (2001), April, Nr. 3. – <http://didaktik.cs.uni-potsdam.de/InformaticaDidactica/Issue3> Ausgewählte Beiträge der Tagung „IAB2000 - Informatik und Ausbildung“
- [Humbert 2002a] HUMBERT, Ludger: Das Modulkonzept – ein zeitgemäßer Ansatz zur informatischen Bildung für alle Schülerinnen. In: *informatica didactica* (2002), November, Nr. 5. – <http://didaktik.cs.uni-potsdam.de/InformaticaDidactica/Issue5> Ausgewählte Beiträge der Tagung „INFOS2001 – 9. GI-Fachtagung Informatik und Schule, Paderborn“
- [Humbert 2002b] HUMBERT, Ludger: *Exploration im Informatikunterricht*. Zusammenfassung: [http://bscw.gmd.de/pub/german.cgi/d33570242-1/\\*/Exploration\\_Abstract.html](http://bscw.gmd.de/pub/german.cgi/d33570242-1/*/Exploration_Abstract.html) Folien: [http://bscw.gmd.de/pub/german.cgi/d34056562/Exploration\\_SCREEN.pdf](http://bscw.gmd.de/pub/german.cgi/d34056562/Exploration_SCREEN.pdf). März 2002b. – 9. Fachdidaktisches Gespräch zur Informatik – Königstein (Sächsische Schweiz) – Einblicke in aktuelle Arbeiten der Informatikdidaktikgruppe an der Universität Dortmund
- [Humbert 2002c] HUMBERT, Ludger: Informatik – übergreifende, einzigartige Metawissenschaft? Überlegungen und fachdidaktischer Kontext. In: [Schubert u. a. 2002], S. 109–118
- [Humbert 2002d] HUMBERT, Ludger: Let's teach informatics - empowering pupils, students and teachers. In: WEERT, Tom van (Hrsg.); MUNRO, Robert (Hrsg.): *Proceedings of Conference on Social, Ethical and Cognitive Issues of Informatics and ICT*. Dordrecht : Kluwer Academic Publishers, July 2002d. – Open IFIP-GI-Conference – July 22-26, 2002, University of Dortmund, Germany – to be published in 2003
- [Humbert 2002e] HUMBERT, Ludger: *Which programming language supports my concepts for education in informatics at school?* July 2002e. – Abstract: <http://didaktik-der-informatik.de/dortmund2002/nrw/humbert.html> – presentation, manuscript: <http://www.ham.nw.schule.de/pub/bscw.cgi/0/20866>
- [Humbert u. a. 2000] HUMBERT, Ludger ; MAGENHEIM, Johannes ; SCHUBERT, Sigrid: *Projekt MUE: Multimediale Evaluation in der Informatiklehrausbildung*. <http://didaktik.cs.uni-potsdam.de/HyFISCH/WorkshopLehrerbildung2000/Papers/Schubert.pdf.zip>. Juli 2000. – Beitrag zum Workshop zur Lehrerausbildung, GI-Jahrestagung 2000, Berlin, 19. September 2000
- [Humbert und Pohlmann 1999] HUMBERT, Ludger ; POHLMANN, Dietrich: Resümee des MNU-GI-Gesprächs anlässlich der 90. Hauptversammlung der MNU 1999 in Saarbrücken. In: *LOG IN* 19 (1999), August, Nr. 3/4, S. 7
- [Humbert und Schubert 1999] HUMBERT, Ludger ; SCHUBERT, Sigrid: Gesamtkonzept der informatischen Bildung – Workshop. In: BEIERSDÖRFER, Kurt (Hrsg.) ; ENGELS, Gregor (Hrsg.) ; SCHÄFER, Wilhelm (Hrsg.): *Informatik '99 – Informatik überwindet Grenzen*. Berlin : Springer, September 1999. – [http://www.ham.nw.schule.de/pub/bscw.cgi/d29357/9\\_Oktober\\_2000\\_Humbert\\_Schubert.pdf](http://www.ham.nw.schule.de/pub/bscw.cgi/d29357/9_Okttober_2000_Humbert_Schubert.pdf) – geprüft: 14. Dezember 2002, S. 344–346

- [Humbert und Schubert 2002] HUMBERT, Ludger ; SCHUBERT, Sigrid: Fachliche Orientierung des Informatikunterrichts in der Sekundarstufe II / Fachbereich Informatik, Universität Dortmund. 2002 ( 771). – Forschungsbericht. [http://www.ham.nw.schule.de/pub/bscw.cgi/d23846/Uni\\_Do\\_cs\\_ddi\\_Forschungsbericht\\_771.pdf](http://www.ham.nw.schule.de/pub/bscw.cgi/d23846/Uni_Do_cs_ddi_Forschungsbericht_771.pdf) – geprüft: 14. Dezember 2002
- [Hurrelmann 1975] HURRELMANN, Klaus: *Erziehungssystem und Gesellschaft*. Reinbek : Rowohlt, 1975
- [Hyman 1987] HYMAN, Anthony: *Charles Babbage: 1791-1871; Philosoph, Mathematiker Computerpionier*. Stuttgart : Klett-Cotta, 1987
- [IEEE LTSC 2002] IEEE LTSC (Hrsg.): *Draft Standard for Learning Object Metadata, Version 6.4*. New York : Institute of Electrical and Electronics Engineers (IEEE); Learning Technology Standards Committee (LTSC), March 2002. – [http://ltsc.ieee.org/doc/wg12/LOM\\_WD6\\_4.pdf](http://ltsc.ieee.org/doc/wg12/LOM_WD6_4.pdf) – geprüft: 6. Oktober 2002
- [Illich 1986] ILLICH, Ivan ; DUVE, Freimut (Hrsg.): *Selbstbegrenzung – Eine politische Kritik der Technik «Tools for Conviviality»*. erweiterte Aufl. Reinbek : Rowohlt, Juni 1986
- [Ingalls u. a. 1988] INGALLS, Dan ; WALLACE, Scott ; CHOW, Yu-Ying ; LUDOLPH, Frank ; DOYLE, Ken: Fabrik. A Visual Programming Environment. In: *Norman K. Meyrowitz (Ed.): Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '88) Conference Proceedings. SIGPLAN Notices* 23 (1988), November, Nr. 11, S. 176–190. – 3. OOPSLA San Diego, California <http://www.ipa.net/~dwighth/smalltalk/Fabrik/Fabrik.html> – geprüft: 11. August 2002
- [Issing und Schaumburg 2002] ISSING, Ludwig ; SCHAUMBURG, Heike: *Lernen mit Laptops*. Gütersloh : Bertelsmann Stiftung, November 2002. – Zusammenfassung (Executive Summary) S. 11-19, <http://www.bertelsmann-stiftung.de/documents/ACFUMN6S3.pdf> – geprüft: 14. Dezember 2002
- [Jacobson u. a. 1992] JACOBSON, Ivar ; CHRISTERSON, Magnus ; JONSSON, Patrik ; ÖVERGAARD, Gunnar: *Object-Oriented Software Engineering: A Use Case Driven Approach*. Reading : Addison-Wesley, 1992
- [Jähnichen und Herrmann 2002] JÄHNICHEN, Stefan ; HERRMANN, Stephan: Was, bitte, bedeutet Objektorientierung? In: *Informatik-Spektrum* 25 (2002), August, Nr. 4, S. 266–276. – <http://link.springer.de/link/service/journals/00287/bibs/2025004/20250266.htm> – geprüft: 9. August 2002
- [Jochum 1998] JOCHUM, Heiko: *Objektorientierung zur Analyse, zum Design und zur Programmierung am Beispiel eines Strategiespiels mit einem Schwerpunkt in arbeitsteiliger Gruppenarbeit*. 1998. – 2. Staatsarbeit für das Lehramt für die Sekundarstufe II, Informatik <http://bscw.hagen.de/pub/german.cgi/d204238/> – geprüft: 10. August 2001
- [Kaasbøll 1998a] KAASBØLL, Jens J.: Exploring didactic models for programming. In: *NIK'98 – Norwegian Computer Science Conference*. Kristiansand, Norwegen, November 1998a. – <http://ifi.uio.no/~jensj/NIK98.ps> – geprüft: 10. August 2002, S. 195–203
- [Kaasbøll 1998b] KAASBØLL, Jens J.: Teaching critical thinking and problem defining skills. In: *Education and Information Technologies – Official Journal of the IFIP Technical Committee on Education (TC 3)* 3 (1998), Nr. 2, S. 101–117. – <http://www.ifi.uio.no/~jensj/critical.ps> – geprüft: 10. August 2002
- [Kanders u. a. 1997] KANDERS, Michael ; RÖSNER, Ernst ; ROLFF, Hans-Günter ; BUNDESMINISTERIUM FÜR BILDUNG, WISSENSCHAFT, FORSCHUNG UND TECHNOLOGIE (Hrsg.): *Das Bild der Schule aus der Sicht von Schülern und Lehrern*. Bonn : BMBF, Juli 1997
- [Karbach 1998] KARBACH, Manfred: *Erziehungswissenschaft: Anmerkungen zum Wort Evaluation*. Juni 1998. – <http://schulen.hagen.de/GSGE/ew/EvalW.html> – geprüft: 27. November 2002

- [Kay 1993] KAY, Alan C.: The Early History of Smalltalk. In: *ACM SIGPLAN Notices* 28 (1993), March, S. 69–95
- [Kay 2002] KAY, Alan C.: Graphical User Interfaces. In: [Broy und Denert 2002], S. 230–231. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Keene 1989] KEENE, Sonya E.: *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*. Addison-Wesley, 1989
- [Keil-Slawik und Magenheimer 2001] KEIL-SLAWIK, Reinhard (Hrsg.) ; MAGENHEIMER, Johannes (Hrsg.): *Informatik und Schule – Informatikunterricht und Medienbildung INFOS 2001 – 9. GI-Fachtagung 17.–20. September 2001, Paderborn*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, September 2001 (GI-Edition – Lecture Notes in Informatics – Proceedings P-8)
- [Keller 1998] KELLER, Christel: *Der Begriff „Globale Informationsgesellschaft“: Wissenschaftliche Theorie – Politisches Programm – Globalisierte Geschäftssphäre – Zur politischen Steuerung der Entwicklung und nationalökonomischen Nutzung der Informationstechnik*, Universität Bremen, Diss., Juni 1998. – <http://www.informatik.uni-tuebingen.de/FILES/bibliothek/wsi-98-8.pdf> – geprüft: 13. Juli 2002
- [Kerschensteiner 1927f] KERSCHENSTEINER, Georg: Kritik der herbartianischen Methode und die produktive Arbeit als neues methodisches Prinzip – Analyse des Arbeitsprozesses. In: [Geißler 1970], S. 50–57. – aus: *Grundfragen der Schulorganisation*, 5. Aufl. Leipzig, Berlin (Teubner), 1927, S. 53ff; *Wesen und Wert des naturwissenschaftlichen Unterrichts*, 3. Aufl. Leipzig, Berlin (Teubner), 1928, S. 48ff
- [Kilpatrick 1918] KILPATRICK, William H.: The Problem-Project. Attack in Organization, Subject-Matter, and Teaching. In: *Addresses and Proceedings of the Fiftysixth Annual Meeting* National Education Association of the United States, 1918, S. 528–531
- [Kim und Toole 1999] KIM, Eugene E. ; TOOLE, Betty A.: Ada und der erste Computer. In: *Spektrum der Wissenschaft* (1999), Juli, Nr. 7, S. 80–85
- [Klafki 1985a] KLAFKI, Wolfgang: Konturen eines neuen Allgemeinbildungskonzepts. In: *Neue Studien zur Bildungstheorie und Didaktik: Beiträge zur kritisch-konstruktiven Didaktik* [Klafki 1985b], S. 12–30
- [Klafki 1985b] KLAFKI, Wolfgang: *Neue Studien zur Bildungstheorie und Didaktik: Beiträge zur kritisch-konstruktiven Didaktik*. Weinheim, Basel : Beltz Verlag, 1985b
- [Klieme und Siegle 2001] KLIEME, Eckhard ; SIEGLE, Thilo: Unterrichtsmuster: Identifizierung, Bedingungen und Wirkungen. In: *60. Tagung der Arbeitsgruppe für Empirisch Pädagogische Forschung (AEPF) – Abstractband*. Bamberg : Universität, März 2001. – [http://wipaed.sowi.uni-bamberg.de/aepf/abstracs/Abstractband\\_PEV.pdf](http://wipaed.sowi.uni-bamberg.de/aepf/abstracs/Abstractband_PEV.pdf) – geprüft 9. September 2002, S. 84
- [Knoll 1997] KNOLL, Michael: The project method: Its vocational education origin and international development. In: *Journal of Industrial Teacher Education* 34 (1997), Spring, Nr. 3, S. 59–80. – <http://scholar.lib.vt.edu/ejournals/JITE/v34n3/Knoll.html> – geprüft: 4. September 2002
- [Knoll 1999] KNOLL, Michael: *Die Rezeption der 'Projektidee' in der schulpädagogischen Literatur*. Februar 1999. – Erweiterte Fassung eines Vortrages, gehalten auf dem Professionspolitischen Kongreß der Deutschen Gesellschaft für Erziehungswissenschaft (DGfE) <http://www.fb12.uni-dortmund.de/dgfeDyn/Beitrag.po?ID=317> – geprüft: 4. September 2002

- [Knoll 2000] KNOLL, Michael: Grundmodelle des Projektunterrichts. Versuch zur Klärung eines unübersichtlichen Konzepts. In: *Pädagogisches Handeln – Wissenschaft und Praxis im Dialog* 4 (2000), Nr. 1. – [http://paedagogischeshandeln.de/ForPrax1\\_2000.htm](http://paedagogischeshandeln.de/ForPrax1_2000.htm) – geprüft: 4. September 2002
- [Knöß 1989] KNÖSS, Petra: *Fundamentale Ideen der Informatik im Mathematikunterricht*. Wiesbaden : Deutscher Universitäts-Verlag, 1989
- [Korber 1986] KOERBER, Bernhard: Leistungsmessung bei der Projektarbeit im Informatikunterricht. In: *LOG IN* 6 (1986), Nr. 4, S. 20–23
- [Korber und Peters 1998] KOERBER, Bernhard (Hrsg.) ; PETERS, Ingo-Rüdiger (Hrsg.): *Informatische Bildung in Deutschland – Perspektiven für das 21. Jahrhundert. Festschrift für Professor Dr. Wolfgang Arlt*. Berlin : LOG IN Verlag, 1998 . – <http://log-in.fachdid.fu-berlin.de/IBiD/index.html> – geprüft 17. Mai 2002
- [Korber und Reker 1982] KOERBER, Bernhard ; REKER, Jörg: Projektarbeit im Informatikunterricht. In: HAAS, Hans W. (Hrsg.) ; WILDENBERG, Detlef (Hrsg.): *Informatik für Lehrer – Studentexte und Handreichungen für den Unterricht* Bd. 2 : Komplexere Probleme und Didaktik der Schulinformatik. München : Oldenbourg Verlag, 1982
- [Köhler u. a. 2001] KÖHLER, Michael ; MOLDT, Daniel ; RÖLKE, Heiko: Modelling a sociological case study. In: *Sozionik aktuell* (2001), Nr. 3. – <http://www.informatik.uni-hamburg.de/TGI/forschung/projekte/sozionik/journal/3/masho-4.ps.gz> – geprüft: 31. Juli 2002
- [Korbmacher 1992] KORBMACHER, Karlheinz: *Eine didaktische Konzeption handlungsorientierten Lernens*. Lippstadt : Korbmacher, 1992. – <http://www.luk-korbmacher.de/Schule/Buwi/texte/lernen/hand.htm> – geprüft: 8. September 2002
- [Krabbel und Kuhlmann 1994] KRABBEL, Anita ; KUHLMANN, Bettina ; BRUNNSTEIN, Klaus (Hrsg.) ; OBERQUELLE, Horst (Hrsg.): *Zur Selbstverständnis-Diskussion in der Informatik / Universität Hamburg, Fachbereich Informatik. D-22527 Hamburg, Vogt-Kölln-Strasse 30, Mai 1994 (FBI-HH-B-169/94)*. – Forschungsbericht
- [Kraimer u. a. 2002] KRAINER, Konrad ; JUNGWIRTH, Helga ; KÜHNELT, Helmut ; STADLER, Helga: Kurzdarstellung von Reformansätzen zur Weiterentwicklung des mathematisch-naturwissenschaftlichen Unterrichts in ausgewählten Ländern – Innovations in Mathematics and Science Teaching (IMST TOP 4) / Interuniversitäres Institut für Interdisziplinäre Forschung und Fortbildung (IFF) – Abteilung “Schule und gesellschaftliches Lernen“. Klagenfurt, Mai 2002. – Forschungsbericht. im Auftrag des Österreichischen Bundesministeriums für Unterricht und kulturelle Angelegenheiten [http://imst.uni-klu.ac.at/was\\_ist\\_imst/\\_content/vorprojekt/IMSTTOP4/InhaltTOP4.htm](http://imst.uni-klu.ac.at/was_ist_imst/_content/vorprojekt/IMSTTOP4/InhaltTOP4.htm) – geprüft: 10. August 2002
- [Krämer 1988] KRÄMER, Sybille: *Symbolische Maschinen: die Idee der Formalisierung in geschichtlichem Abriß*. Darmstadt : Wissenschaftliche Buchgesellschaft, 1988
- [Krämer 1997] KRÄMER, Sybille: Werkzeug – Denkzeug – Spielzeug. Zehn Thesen über unseren Umgang mit Computern. In: [Hoppe und Luther 1997], S. 7–13
- [Kugler 1986] KUGLER, H.-J. (Hrsg.) ; IFIP (Veranst.): *Information Processing 86*. North-Holland : Elsevier Science Publishers B.V., September 1986
- [Kuhlen 2002] KUHLEN, Rainer: „Privatisierung des Wissens“ Gutachten in Auftrag gegeben von der Enquete-Kommission des Deutschen Bundestages „Globalisierung der Weltwirtschaft – Herausforderungen und Antworten“. Konstanz, Berlin : Universität, Februar 2002. – Die Rechte an diesem Gutachten hat der Deutsche Bundestag. <http://www.inf-wiss.uni-konstanz.de/People/RK/gutachten//gutachten-enquete-v3-080302.pdf> – geprüft: 19. September 2002

- [Kuhn 1969] KUHN, Thomas S.: *Die Struktur wissenschaftlicher Revolutionen*. 15. Aufl. Reinbek : Rowohlt, 1969. – Originalausgabe 1962: *The Structure of Scientific Revolutions* : University of Chicago
- [Kultusministerium Thüringen 1999] KULTUSMINISTERIUM THÜRINGEN (Hrsg.): *Lehrplan für das Gymnasium, Informatik (Jahrgangsstufe 11 – 12)*. Saalfeld : Satz + Druck Centrum Saalfeld, Mai/Juni 1999. – [http://www.thillm.th.schule.de/lehrplan/lehrpl\\_gy/99gyinfo.pdf](http://www.thillm.th.schule.de/lehrplan/lehrpl_gy/99gyinfo.pdf) – geprüft: 20. August 2002
- [Kultusministerkonferenz 1999] KULTUSMINISTERKONFERENZ: *Vereinbarung zur Gestaltung der gymnasialen Oberstufe in der Sekundarstufe II*. <http://www.kmk.org/>. Oktober 1999. – Beschluss der KMK vom 7. Juli 1972 i. d. F. vom 22. Oktober 1999
- [van Leeuwen und Wiedermann 2000] LEEUWEN, Jan van ; WIEDERMANN, Jiří: *On the Power of Interactive Computing*. In: LEEUWEN, Jan van (Hrsg.) ; WATANABE, Osamu (Hrsg.) ; HAGIYA, Masami (Hrsg.) ; MOSSES, Peter D. (Hrsg.) ; ITO, Takayasu (Hrsg.): *IFIP TCS*. Berlin, Heidelberg : Springer, 2000 (Lecture Notes in Computer Science 1872). – <http://link.springer.de/link/service/series/0558/bibs/1872/18720619.htm> – geprüft: 17. Juli 2002, S. 619–623
- [Lehmann 1985] LEHMANN, Eberhard: *Projektarbeit im Informatikunterricht – Entwicklung von Softwarepaketen und Realisierung in PASCAL*. Stuttgart : B.G. Teubner, 1985 (Mikro-Computer-Praxis)
- [Linkweiler 2002] LINKWEILER, Ingo: *Eignet sich die Skriptsprache Python für schnelle Entwicklungen im Softwareentwicklungsprozess? – Eine Untersuchung der Programmiersprache Python im softwaretechnischen und fachdidaktischen Kontext*. Dortmund, Universität, Fachbereich Informatik, Fachgebiet Didaktik der Informatik, Diplomarbeit, November 2002. – <http://www.ingo-linkweiler.de/diplom/Diplomarbeit.pdf> – geprüft: 3. Dezember 2002
- [Lockemann 1986] LOCKEMANN, Peter C.: *Konsistenz, Konkurrenz, Persistenz – Grundbegriffe der Informatik?* In: *Informatik Spektrum* 9 (1986), Oktober, Nr. 5, S. 300–305
- [Loser und Terhart 1977] LOSER, F. (Hrsg.) ; TERHART, Ewald (Hrsg.): *Theorien des Lehrens*. Stuttgart : Klett, 1977
- [LSW 1991] LSW (Hrsg.): *Orientierungshilfe zur Ausstattung von allgemeinbildenden Schulen mit Hard- und Software*. 2. Aufl. LSW Soest, Beratungsstelle für Neue Technologien (BfNT) : Soester Verlagskontor, Februar 1991. – LSW – Landesinstitut für Schule und Weiterbildung (Soest, Nordrhein-Westfalen)
- [LSW 1993] LSW (Hrsg.): *Orientierungshilfe zur Ausstattung von allgemeinbildenden Schulen mit Hard- und Software*. 4. Aufl. LSW Soest, Beratungsstelle für Neue Technologien (BfNT) : Soester Verlagskontor, Oktober 1993
- [LSW 1995a] LSW (Hrsg.): *Lehren und Lernen als konstruktive Tätigkeit. Beiträge zu einer konstruktivistischen Theorie des Unterrichts*. 1. Aufl. Bönen : Verlag für Schule und Weiterbildung, 1995a (Curriculumentwicklung 4108). – LSW – Landesinstitut für Schule und Weiterbildung (Soest, Nordrhein-Westfalen)
- [LSW 1995b] LSW (Hrsg.): *Orientierungshilfe zur Ausstattung Kaufmännischer Schulen mit Neuen Technologien*. Soest : Landesinstitut für Schule und Weiterbildung (LSW) – Nordrhein-Westfalen, Herbst 1995b. – Redaktion: Karlheinz Korbmacher
- [LSW 1995c] LSW (Hrsg.): *Orientierungshilfe zur Ausstattung von allgemeinbildenden Schulen mit Hard- und Software*. 5. Aufl. Bönen : Verlag für Schule und Weiterbildung DruckVerlag Kettler, November 1995c. – Bearbeiter: Wolfgang Weber
- [Luft 1992] LUFT, Alfred L.: *Grundlagen einer Theorie der Informatik: «Wissen» und «Information» bei einer Sichtweise der Informatik als Wissenstechnik*. In: [Coy u. a. 1992], S. 49–70

- [Magenheim u. a. 2000] MAGENHEIM, Johannes ; FREYE, Thomas ; SCHUBERT, Sigrid ; HUMBERT, Ludger: *Computergestützte Evaluation von Lehr-/ Lernprozessen in Hochschule und Studienseminar*. <http://www.learn-line.nrw.de/angebote/zukunftlehren/info/pdf/ws4magfsh.pdf>. März 2000. – erreichbar über <http://www.learn-line.nrw.de/angebote/zukunftlehren/info/workshops.html>
- [Magenheim u. a. 1999] MAGENHEIM, Johannes ; SCHULTE, Carsten ; HAMPEL, Thorsten: Dekonstruktion von Informatiksystemen als Unterrichtsmethode – Zugang zu objektorientierten Sichtweisen im Informatikunterricht. In: [Schwill 1999], S. 149–164
- [Mainzer 1979] MAINZER, Klaus: Entwicklungsfaktoren der Informatik in der Bundesrepublik Deutschland. In: DAELE, Wolfgang van d. (Hrsg.) ; KROHN, Wolfgang (Hrsg.) ; WEINGART, Peter (Hrsg.): *Geplante Forschung. Vergleichende Studien über den Einfluß politischer Programme auf die Wissenschaftsentwicklung*. Frankfurt a. M. : Suhrkamp Verlag, 1979, S. 117–180
- [Malsch 1998] MALSCH, Thomas (Hrsg.): *Sozionik. Soziologische Ansichten über künstliche Sozialität*. Berlin : edition sigma, 1998
- [Mandl und Reinmann-Rothmeier 1995] MANDL, Heinz ; REINMANN-ROTHMEIER, Gabi: Unterrichten und Lernumgebungen gestalten / Ludwig-Maximilians-Universität, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie. München, 1995 ( Nr. 60 ). – Forschungsbericht. <http://www.ph-weingarten.de/homepage/faecher/psychologie/konrad/rl/Reinma.rtf> – geprüft: 6. April 2002
- [Maturana 1994] MATURANA, Humberto R.: *Was ist erkennen?* München : Piper Verlag, 1994
- [Maturana und Varela 1992] MATURANA, Humberto R. ; VARELA, Francisco J.: *Der Baum der Erkenntnis. Die biologischen Wurzeln des menschlichen Erkennens*. 4. Aufl. München : Goldmann Verlag, Oktober 1992. – Originalausgabe: 1984 *El árbol del conocimiento*; einzig berechtigte Übersetzung Scherz Verlag, Bern; 1987
- [Maughan u. a. 1980] MAUGHAN, Barbara ; MORTIMER, Peter ; RUTTER, Michael: *Fünfzehntausend Stunden. Schulen und ihre Wirkung auf die Kinder*. Weinheim, Basel : Beltz und W. Gelberg, 1980
- [Maurer 2000] MAURER, Hermann: Overflow / Prognosen und Thesen . . . nicht nur zum Schmunzeln. In: *Informatik-Spektrum* 23 (2000), Februar, Nr. 1, S. 51–59
- [Menzel 1984] MENZEL, Klaus: Allgemeinbildung im Umgang mit dem Computer – Projekte und Ansätze Baden-Württemberg. In: [Arlt und Haefner 1984], S. 323–326
- [Meyer 1987] MEYER, Hilbert: *Unterrichtsmethoden*. Bd. II: Praxisband. 1. Aufl. Frankfurt a. M. : Scriptor Verlag, 1987
- [Meyer 1988] MEYER, Hilbert: *Unterrichtsmethoden*. Bd. I: Theorieband. 2. Aufl. Frankfurt a. M. : Scriptor Verlag, 1988
- [Meyer 1997] MEYER, Hilbert: *Schulpädagogik*. Bd. II: Für Fortgeschrittene. Berlin : Cornelsen Lehrbuch, 1997
- [Micheuz 2002] MICHEUZ, Peter: *Schulinformatik in Österreich*. März/Juni 2002. – <http://www.gym1.at/schulinformatik/organisation/lehrplanentwerfe/situation-oest.pdf> – geprüft: 20. Oktober 2002
- [Ministerium für Bildung, Wissenschaft und Kultur – Mecklenburg-Vorpommern 2001] MINISTERIUM FÜR BILDUNG, WISSENSCHAFT UND KULTUR – MECKLENBURG-VORPOMMERN (Hrsg.): *Rahmenplan Gymnasiale Oberstufe Informatik – Jahrgangsstufen 11 bis 13 – Erprobungsfassung*. Schwerin, Mecklenburg-Vorpommern : KM M-V, Oktober 2001. – [http://www.tedu.ethz.ch/didaktik/id2\\_docs/rahmenlehrplan\\_mv.pdf](http://www.tedu.ethz.ch/didaktik/id2_docs/rahmenlehrplan_mv.pdf) – geprüft: 16. November 2002

- [Modrow 1998] MODROW, Eckart: *Automaten, Schaltwerke, Sprachen - Lehr- und Übungsbuch der technischen und theoretischen Informatik*. Bonn : Dümmler, 1998 (Bausteine Informatik)
- [Möller 1973] MÖLLER, Christine: *Technik der Lernplanung*. 4. überarbeitete Auflage. Weinheim : Beltz, 1973
- [Montessori 1926] MONTESSORI, Maria: Mein experimenteller Beitrag. In: [Geißler 1970], S. 90–102. – aus: M. Montessori: Montessorierziehung für Schulkinder I. Stuttgart (J. Hoffmann), S. 72ff.
- [Mössenböck 1992] MÖSSENBÖCK, Hans-Peter: *Objektorientierte Programmierung in Oberon-2*. 1. Aufl. Berlin : Springer, 1992
- [MSWWF 1999] MSWWF (Hrsg.): *Richtlinien und Lehrpläne für die Sekundarstufe II - Gymnasium/Gesamtschule in Nordrhein-Westfalen - Informatik*. 1. Aufl. Frechen : Ritterbach Verlag, Juni 1999. – MSWWF (Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen)
- [Mulder und van Weert 2000] MULDER, Fred ; WEERT, Tom J.: *IFIP/UNESCO Informatics Curriculum Framework 2000 – Building effective higher Education Informatics Curricula in a Situation of Change*. Paris : UNESCO, 2000. – ICF-2000 <http://poe.netlab.csc.villanova.edu/ifip32/ICF2000.htm> – geprüft: 8. Oktober 2002
- [Münker 1996] MÜNKER, Stefan: Die Schwierigkeit mit der Schnelligkeit – Über den Deutschen Kongreß für Philosophie in Leipzig. In: *Telepolis* (1996), Oktober. – <http://www.heise.de/tp/deutsch/inhalt/konf/2059/2.html> – geprüft: 20. Mai 2002
- [Nake 2001] NAKE, Frieder: AG Semiotische Aufregung: Denn eben wo Begriffe fehlen .... In: [Nake u. a. 2001], S. 10–14
- [Nake u. a. 2001] NAKE, Frieder (Hrsg.) ; ROLF, Arno (Hrsg.) ; SIEFKES, Dirk (Hrsg.): *Informatik: Aufregung zu einer Disziplin. Arbeitstagung mit ungewissem Ausgang*. Berlin : Technische Universität, Dezember 2001 (<http://tal.cs.tu-berlin.de:80/siefkes/Heppenheim/Heppenheimerbericht.pdf> – geprüft: 13. Juli 2002)
- [Nassi und Shneiderman 1973] NASSI, Isaac ; SHNEIDERMAN, Ben: Flowchart Techniques for Structured Programming. In: *ACM SIGPLAN Notices* 8 (1973), August, Nr. 8, S. 12–26
- [Naumann 2001] NAUMANN, Friedrich: *Vom Abakus zum Internet – Die Geschichte der Informatik*. Darmstadt : Primus Verlag, 2001
- [Naur und Randell 1969] NAUR, Peter (Hrsg.) ; RANDELL, Brian (Hrsg.): *Software Engineering. Report on a conference sponsored by the NATO Science Committee*. Brussels 39, Belgium : NATO, Scientific Affairs Division, January 1969 . – Garmisch, Garmay, 7<sup>th</sup> to 11<sup>th</sup> October 1968
- [von Neumann 1945] NEUMANN, John von: First Draft of a Report on the EDVAC. In: *University of Pennsylvania* (1945). – <http://www.histech.rwth-aachen.de/www/quellen/vnedvac.pdf> – geprüft: 14. Juli 2002
- [Nievergelt 1991] NIEVERGELT, Jürg: *Was ist Informatik-Didaktik?* Oldenburg : Universität, 1991. – Sonderdruck 4. Fachtagung „Informatik und Schule“
- [Nievergelt 1993] NIEVERGELT, Jürg: Was ist Informatik-Didaktik? Gedanken über die Fachkenntnisse des Informatiklehrers. In: *Informatik Spektrum* 16 (1993), Februar, Nr. 1, S. 3–10. – <http://www.informatik.uni-leipzig.de/~compalg/ili/nieverg.txt> – geprüft: 23. März 2002
- [Norm DIN 66 261 ] NORM DIN 66 261. *Informationsverarbeitung; Sinnbilder nach Nassi-Shneiderman und ihre Anwendung in Programmablaufplänen*

- [Norris u. a. 2002] NORRIS, Cathleen ; SOLOWAY, Elliot ; SULLIVAN, Terry: Examining 25 years of technology in U.S. education. In: *CACM* 45 (2002), August, Nr. 8, S. 15–18. – COLUMN: Log on education
- [Nortmeyer 2002] NORTMEYER, Isolde: *Evaluation – Deutsches Fremdwörterbuch (DFWB)*. Neubearbeitung des DFWB, erschienen bei de Gruyter. April 2002. – <http://www.ids-mannheim.de/lexik/fremdwort/artikel/Evaluation.pdf> – geprüft: 27. November 2002
- [Nuttelmann 1999] NUTTELMANN, Sanna: Geheime Mitteilungen? – Ein Unterrichtsbeispiel zum Thema Kryptologie in einem Leistungskurs Informatik. In: [Schwill 1999], S. 219–231
- [Nygaard 1986] NYGAARD, Kristen: Program Development as a Social Activity. In: [Kugler 1986], S. 189–198. – [http://www.ifi.uio.no/~kristen/PDF\\_MAPPE/F\\_PDF\\_MAPPE/F\\_IFIP\\_86.pdf](http://www.ifi.uio.no/~kristen/PDF_MAPPE/F_PDF_MAPPE/F_IFIP_86.pdf) – geprüft: 14. Juli 2002
- [OECD 2001] OECD (Hrsg.): *Lernen für das Leben. Erste Ergebnisse der internationalen Schulleistungstudie PISA 2000*. Paris : Organisation for Economic Co-operation and Development (OECD), 2001
- [Opp und Freytag 1997] OPP, Günther ; FREYTAG, Andreas: Warum Lehrerinnen und Lehrer nicht tun, wozu sie von allen Seiten aufgefordert werden. In: HEIMLICH, Ulrich (Hrsg.): *Zwischen Aussonderung und Integration. Schülerorientierte Förderung bei Lern- und Verhaltensschwierigkeiten*. Neuwied, Berlin : Luchterhand, 1997, S. 270–281
- [Ortmann und Koch 2001] ORTMANN, Tobias ; KOCH, Ludger: *Werkzeuge für objektorientiertes Modellieren im Informatikunterricht*. März 2001. – Projektarbeit im Rahmen des Informatik-Lehramtsstudiums (Vertiefung in der Didaktik der Informatik) am Fachbereich Informatik der Universität Dortmund  
[http://www.ham.nw.schule.de/pub/bscw.cgi/d29348/Ortmann\\_Koch\\_Projektarbeit010301.pdf](http://www.ham.nw.schule.de/pub/bscw.cgi/d29348/Ortmann_Koch_Projektarbeit010301.pdf) – geprüft: 8. Dezember 2002
- [Ottmann und Widmayer 1990] OTTMANN, Thomas ; WIDMAYER, Peter: *Informatik. Bd. 70: Algorithmen und Datenstrukturen*. Mannheim : B.I. Wissenschaftsverlag, 1990
- [Otto 1913] OTTO, Berthold: Gesamtunterricht. In: [Geißler 1970], S. 67–78. – Vortrag B. Ottos, gehalten am 21.10.1913 in seiner Hauslehrerschule. 1. pädagogische Flugschrift des Berthold Otto-Vereins, Berlin-Lichterfelde (Vlg. des Hauslehrers)
- [Padawitz 2000] PADAWITZ, Peter: *Einführung ins funktionale Programmieren – Vorlesungsskriptum Sommersemester 1999*. Dortmund : Universität Dortmund, 2000. – LV Grundlagen und Methoden funktionaler Programmierung <http://ls5-www.cs.uni-dortmund.de/padawitz.html>
- [Parnas 1986] PARNAS, David L.: Software Wars. Offener Brief an Mr. H. Offut, Ministerium für Verteidigung, Washington. In: *Kursbuch 83, Rotbuch Verlag, Berlin* (1986), März, S. 49–69. – Aus dem Amerikanischen von Philip Bacon
- [Parnas 1989] PARNAS, David L.: Respond to Dijkstra: “On the cruelty of Really Teaching Computing Science”. In: [Denning 1989], S. 1405–1406. – *CACM* 32 (1989), Nr. 12
- [Parnas 2002] PARNAS, David L.: The Secret History of Information Hiding. In: [Broy und Denert 2002], S. 398–409. – sd&m Konferenz, 28., 29. Juni 2001 Webseite zur Konferenz mit Hintergrundmaterial zu den Teilnehmern und Verweisen auf die Vorträge als Videodatenströme: <http://www.sdm.info/de/termine/sdmkonf-2001/index.html> – geprüft: 16. Dezember 2002
- [Pawlowski 2002] PAWLOWSKI, Jan M.: Modellierung didaktischer Konzepte. In: SCHUBERT, Sigrid (Hrsg.) ; REUSCH, Bernd (Hrsg.) ; JESSE, Norbert (Hrsg.): *Informatik bewegt – Informatik 2002, 32. Jahrestagung der GI 30. Sept. – 3. Okt. 2002 in Dortmund*. Bonn : Köllen Druck + Verlag GmbH, Oktober 2002 (GI-Edition – Lecture Notes in Informatics – Proceedings P-19). – [http://www.rz.uni-frankfurt.de/neue\\_medien/standardisierung/pawlowski\\_text.pdf](http://www.rz.uni-frankfurt.de/neue_medien/standardisierung/pawlowski_text.pdf) – geprüft: 12. Oktober 2002, S. 369–374

- [Penrose 1991] PENROSE, Roger: *Computerdenken*. Heidelberg : Spektrum Verlag, 1991. – The Emperor's New Mind: Concerning Computers, Minds and the Laws of Physics; Amerikanische Erstausgabe bei Oxford University Press, 1989
- [Perrochon 1996] PERROCHON, Louis: *School goes Internet: das Buch für mutige Lehrerinnen und Lehrer*. 1. Aufl. Heidelberg : dpunkt, 1996
- [Peschke 1989] PESCHKE, Rudolf: Die Krise des Informatikunterrichts in den neunziger Jahren. In: STETTER, Franz (Hrsg.) ; BRAUER, Wilfried (Hrsg.): *Informatik und Schule 1989: Zukunftsperspektiven der Informatik für Schule und Ausbildung*. Berlin, Heidelberg : Springer, 1989 (Informatik-Fachberichte 220), S. 89–98
- [Petri 1983] PETRI, Carl A.: Zur 'Vermenschlichung' des Computers. In: *Der GMD-Spiegel* (1983), Nr. 3/4, S. 42–44
- [Piaget 1948] PIAGET, Jean: *Psychologie der Intelligenz*. Zürich, Stuttgart : Rascher, 1948
- [Pólya 1945] PÓLYA, George: *How to Solve It: A New Aspect of Mathematical Method*. Princeton, NJ : Princeton University Press, 1945
- [Pólya 1967] PÓLYA, George: *Vom Lösen mathematischer Aufgaben*. Bd. 1 und 2. Basel : Birkhäuser, 1966, 1967
- [Popper 1972] POPPER, Karl R.: *Logik der Forschung*. 2. Aufl. Tübingen : J. C. B. Mohr, 1972
- [Prokop 2002] PROKOP, Dieter: *Die Unzufriedenheit mit den Medien. Das Theorie-Erzählbuch der neuen kritischen Medienforschung*. Hamburg : VSA-Verlag, 2002
- [Pütt 1982] PÜTT, Heinz: *Projektunterricht und Vorhabengestaltung*. 1. Aufl. Essen : Neue Deutsche Schule Verlagsgesellschaft, 1982 (neue pädagogische bemühungen 90)
- [Quibeldey-Cirkel 1994] QUIBELDEY-CIRKEL, Klaus: *Das Objekt-Paradigma in der Informatik*. Stuttgart : Teubner-Verlag, 1994
- [Quibeldey-Cirkel 1995] QUIBELDEY-CIRKEL, Klaus: Quo vadis, Informatik? Aspekte einer objektorientierten Entwurfslehre. In: *Objekt-Spektrum 2* (1995), Nr. 1, S. 30–36
- [Rauterberg 1992] RAUTERBERG, Matthias: Partizipative Modellbildung zur Optimierung der Softwareentwicklung. In: STUDER, Rudi (Hrsg.): *Informationssysteme und Künstliche Intelligenz: Modellierung – 2. Workshop Ulm, 24.–26. Februar 1992 – Proceedings*. Berlin Heidelberg New York London Paris Tokyo Hong Kong Barcelona Budapest : Springer, 1992. – <http://www.ip0.tue.nl/homepages/mrauterb/publications/GIWS92paper.pdf> – geprüft: 3. August 2002, S. 113–128
- [Rechenberg 1991] RECHENBERG, Peter: *Was ist Informatik? Eine allgemeinverständliche Einführung*. 1. Aufl. München : Carl Hanser Verlag, 1991
- [Reich 2000] REICH, Gabriele (Hrsg.): *Bundswettbewerb Informatik – Aufgaben und Lösungen 12., 13. Wettbewerb 1993/1994, 1995/1996*. GMD – Forschungszentrum Informationstechnik GmbH, Mai 2000 (GMD Report 97)
- [Reichert u. a. 2000] REICHERT, Raimond ; NIEVERGELT, Jürg ; HARTMANN, Werner: Ein spielerischer Einstieg in die Programmierung mit Java. Kara to Java – erste Schritte beim Programmieren. In: *Informatik Spektrum 23* (2000), Oktober, Nr. 5, S. 309–315
- [Rein 1893] REIN, Wilhelm: Theorie des Lehrverfahrens. In: [Geißler 1970], S. 29–34. – aus: Pädagogik im Grundriß. 2. Aufl. Stuttgart (Götschen) 1893. S. 107ff
- [Reiser 1991] REISER, Martin: *The Oberon System: user guide and programmer's manual*. New York : ACM Press, 1991

- [Reiser und Wirth 1992] REISER, Martin ; WIRTH, Niklaus: *Programming in Oberon: steps beyond Pascal and Modula*. New York : ACM Press, 1992
- [Reiser und Wirth 1994] REISER, Martin ; WIRTH, Niklaus: *Programmieren in Oberon: Das neue Pascal*. New York : ACM Press, 1994
- [Reisin 1992] REISIN, Fanny-Michaela: *Kooperative Gestaltung in partizipativen Softwareprojekten*. Berlin : Peter Lang, 1992
- [Rickert u. a. 2001] RICKERT, Wibke ; CREMER, Thomas ; DESCHEPPER, Patrick ; HUMBERT, Ludger: Qualifizierung von Informatiklehrern in der zweiten Ausbildungsphase der Lehrerbildung. In: **[Keil-Slawik und Magenheimer 2001]**, S. 223–226
- [Rittershofer 2001] RITTERSHOFER, Andreas: *Überlegungen zu netzwerkfähiger Software für die Schule – Version 2*. Februar 2001. – – Version 2, <http://www.rittershofer.de/info/schsoft/inss.pdf> – geprüft: 6. Dezember 2002
- [Robbins und Mayer 1998] ROBBINS, Jason ; MAYER, Niels: *ArgoUML: The Cognitive CASE Tool*. BSD License. 1998. – <http://argouml.tigris.org/> – geprüft: 7. Dezember 2002
- [Robson 1981] ROBSON, David: Object-Oriented Software Systems. In: *Byte* 6 (1981), August, Nr. 8, S. 74–86
- [Rojas u. a. 2000a] ROJAS, Raúl ; GÖKTEKIN, Cüneyt ; FRIEDLAND, Gerald ; KRÜGER, Mike ; KUNISS, Denis ; LANGMACK, Olaf: Plankalkül: The First High-Level Programming Language and its Implementation / Freie Universität Berlin – Institut für Informatik und Feinarbeit. FB Mathematik und Informatik Takustr. 9 14195 Berlin, Februar 2000a ( B-3/2000). – Technical Report. <http://www.zib.de/zuse/Inhalt/Programme/Plankalkuel/Plankalkuel-Report/techreport.pdf> – geprüft: 11. Juli 2002
- [Rojas u. a. 2000b] ROJAS, Raúl ; GÖKTEKIN, Cüneyt ; FRIEDLAND, Gerald ; KRÜGER, Mike ; SCHARF, Ludmila ; KUNISS, Denis ; LANGMACK, Olaf: Konrad Zuses Plankalkül – Seine Genese und eine moderne Implementierung / Freie Universität Berlin – Institut für Informatik and Transform/Feinarbeit. FB Mathematik und Informatik Takustr. 9 14195 Berlin, 2000b. – Forschungsbericht. <http://www.zib.de/zuse/Inhalt/Programme/Plankalkuel/Genese/Genese.pdf> – geprüft: 11. Juli 2002
- [Roth 1976] ROTH, Heinrich: *Pädagogische Psychologie des Lehrens und Lernens*. 15. Aufl. Hannover : Schroedel, 1976
- [Sannella 1997] SANNELLA, Donald: What Does the Future Hold for Theoretical Computer Science? In: *Proc. 7th Intl. Joint Conf. on Theory and Practice of Software Development (TAPSOFT'97)*. Berlin : Springer, 1997 (Lecture Notes in Computer Science 1214). – <ftp://ftp.dcs.ed.ac.uk/pub/dts/tapsoft97.ps> – geprüft 2. August 2002, S. 15–19
- [Schmucker 1986] SCHMUCKER, Kurt J.: *Object Oriented Programming for the Macintosh*. Hasbrouck Heights, New Jersey : Hayden, 1986
- [Schneider 2002] SCHNEIDER, Markus: Rekursive Strukturen in Einführungsvorlesungen der Informatik. In: **[Schubert u. a. 2002]**, S. 77–86
- [Schöning 1995] SCHÖNING, Uwe: Complexity Theory and Interaction. In: HERKEN, Rolf (Hrsg.): *The Universal Turing Machine – A Half-Century Survey* Bd. II. Berlin : Kammerer & Unverzagt, 1995, S. 519–536
- [Schöning 2002] SCHÖNING, Uwe: *Ideen der Informatik. Grundlegende Modelle und Konzepte*. München, Wien : Oldenbourg, 2002

- [Schubert 1991] SCHUBERT, Sigrid: Fachdidaktische Fragen der Schulinformatik und (un)mögliche Antworten. In: GORNY, Peter (Hrsg.): *Informatik: Wege zur Vielfalt beim Lehren und Lernen* Bd. 292. Berlin, Heidelberg : Springer, 1991, S. 27–33
- [Schubert 1992] SCHUBERT, Sigrid: Logische Programmierung. In: **[Bosler 1992]**, S. 171–179
- [Schubert 1995] SCHUBERT, Sigrid (Hrsg.): *Innovative Konzepte für die Ausbildung*. Berlin, Heidelberg : Springer, 1995 (Informatik aktuell)
- [Schubert 2001a] SCHUBERT, Sigrid: *Skript zur Vorlesung Didaktik der Informatik I (für Sekundarstufe II)*. [http://didaktik-der-informatik.de/lehre/grundstudium/skript\\_html](http://didaktik-der-informatik.de/lehre/grundstudium/skript_html). Oktober 2001a. – Universität Dortmund, Didaktik der Informatik – geprüft: 18. November 2002
- [Schubert 2001b] SCHUBERT, Sigrid: Stand der Fachdidaktik Informatik. In: SEIFFERT, Monika (Hrsg.): *Tagungsband „Fachtagung Informatik“*. Hamburg : Behörde für Schule, 2001b. – [http://didaktik-der-informatik.de/ddi\\_bib/forschung/pub/schubert-hamburg-2001.pdf](http://didaktik-der-informatik.de/ddi_bib/forschung/pub/schubert-hamburg-2001.pdf), S. II.1–II.7
- [Schubert u. a. 2002] SCHUBERT, Sigrid (Hrsg.) ; MAGENHEIM, Johannes (Hrsg.) ; HUBWIESER, Peter (Hrsg.) ; BRINDA, Torsten (Hrsg.) ; Gesellschaft für Informatik Didaktik-Workshop (Veranst.): *Forschungsbeiträge zur „Didaktik der Informatik“ – Theorie, Praxis, Evaluation. 1. GI-Workshop DDI'02 (Schwerpunkt: Modellierung in der informatischen Bildung, 10.–11. Okt. 2002 in Witten-Bommerholz*. Bonn : Gesellschaft für Informatik, Köllen Druck + Verlag GmbH, Oktober 2002 (GI-Edition – Lecture Notes in Informatics – Proceedings P 22)
- [Schubert und Schwill 1996] SCHUBERT, Sigrid ; SCHWILL, Andreas: Informatik in der Schule – ein Fach im Wandel. Informationen vom 3. Fachdidaktischen Gespräch zur Informatik an der Technischen Universität Dresden. In: *LOG IN* 16 (1996), Nr. 2, S. 32–33. – Ergebnisse der Arbeitsgruppe: Strukturierung des Schulfachs Informatik aus der Sicht der Fachwissenschaft
- [Schulmeister 2002] SCHULMEISTER, Rolf: *Grundlagen hypermedialer Lernsysteme. Theorie – Didaktik – Design*. 3. Aufl. München, Wien : Oldenbourg, 2002
- [Schulte 2001] SCHULTE, Carsten: Vom Modellieren zum Gestalten – Objektorientierung als Impuls für einen neuen Informatikunterricht? In: *informatica didactica* (2001), Juli, Nr. 3. – <http://didaktik.cs.uni-potsdam.de/InformaticaDidactica/Issue3> Ausgewählte Beiträge der Tagung „IAB2000 - Informatik und Ausbildung“
- [Schulz-Zander u. a. 1993] SCHULZ-ZANDER, Renate ; BRAUER, Wilfried ; BURKERT, Jürgen ; HEINRICHS, U. ; HILTY, Lorenz M. ; HÖLZ, I. ; KEIDEL, K. ; KLAGES, Albrecht ; KOERBER, Bernhard ; MEYER, M. ; PESCHKE, Rudolf ; PFLÜGER, Jörg ; REINEKE, Vera ; SCHUBERT, Sigrid: Veränderte Sichtweisen für den Informatikunterricht. GI-Empfehlungen für das Fach Informatik in der Sekundarstufe II allgemeinbildender Schulen. In: TROITZSCH, Klaus G. (Hrsg.): *Informatik als Schlüssel zur Qualifikation*. Berlin, Heidelberg : Springer, 1993 (Informatik aktuell). – Gesellschaft für Informatik e. V., S. 205–218
- [Schwabe u. a. 2001] SCHWABE, Gerhard (Hrsg.) ; STREITZ, Norbert (Hrsg.) ; UNLAND, Rainer (Hrsg.): *CSCW-Kompendium – Lehr- Handbuch zum computerunterstützten kooperativen Arbeiten*. Heidelberg : Springer, 2001
- [Schwill 1993] SCHWILL, Andreas: Fundamentale Ideen der Informatik. In: *Zentralblatt für Didaktik der Mathematik* 25 (1993), Nr. 1, S. 20–31. – <http://www.informatikdidaktik.de/Forschung/Schriften/ZDM.pdf> – geprüft: 4. August 2002
- [Schwill 1999] SCHWILL, Andreas (Hrsg.): *Informatik und Schule – Fachspezifische und fachübergreifende didaktische Konzepte*. Berlin : Springer, September 1999 (Informatik aktuell)
- [Schwill 2001] SCHWILL, Andreas: Ab wann kann man mit Kindern Informatik machen? Eine Studie über die informatischen Fähigkeiten von Kindern. In: **[Keil-Slawik und Magenheimer 2001]**, S. 13–30

- [Sedgewick 1992] SEDGEWICK, Robert: *Algorithmen*. 4. unveränderter Nachdruck, 1998. Reading, Mass. : Addison-Wesley, 1992
- [Seiffert 2000] SEIFFERT, Monika. *Abituraufgaben zur Informatik*. <http://koenigstein.inf.tu-dresden.de/00/seiffert/abitur.html>. März 2000
- [Sezgen 2001] SEZGEN, Beytullah: *Testgenerierung und Auswertung – PyLZK*. Dortmund, Universität Dortmund, Fachbereich Informatik, Lehrstuhl XII, Fachgebiet Didaktik der Informatik, Betriebliche Projektarbeit, 2001. – <http://bscw.gmd.de/pub/german.cgi/0/33337001> – geprüft: 12. Dezember 2001
- [Sezgen 2002] SEZGEN, Beytullah: *PyNetzwerkmonitor*. Dortmund, Essen, Universität Dortmund, Fachbereich Informatik, Lehrstuhl XII, Fachgebiet Didaktik der Informatik, Bericht zur betrieblichen Projektarbeit zur Erlangung des Abschlusses als Fachinformatiker – Fachrichtung Anwendungsentwicklung, 2002. – <http://bscw.gmd.de/pub/german.cgi/0/40074176> – geprüft: 29. Januar 2003
- [Shannon 1948] SHANNON, Claude E.: A Mathematical Theory of Communication. In: *Bell System Technical Journal* 27 (1948), July, October, S. pp. 379–423 and pp. 623–656. – <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.tar.gz> <http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html> – geprüft: 14. Juli 2002
- [Shannon und Weaver 1973] SHANNON, Claude E. ; WEAVER, Warren: *Informationstheorie*. Stuttgart : B.G. Teubner Verlag, 1973. – deutsche Übersetzung von [Shannon 1948]
- [Shaw 1992] SHAW, Mary: *We Can Teach Software Better*. September 1992. – Computing Research News, 4, 4, September 1992, pp. 2, 3, 4, 12; Reprinted in Journal of Computer Science Education, 7, 3, Spring 1993, pp. 4-7 <http://spoke.compose.cs.cmu.edu/shaweb/edparts/crn.htm> – geprüft 4. August 2002
- [Siefkes 2001] SIEFKES, Dirk: Informatikobjekte entstehen durch Hybridisierung. Techniken der Softwareentwicklung und Entwicklung der Softwaretechnik. In: BAUKNECHT, Kurt (Hrsg.) ; BRAUER, Wilfried (Hrsg.) ; MÜCK, Thomas (Hrsg.): *Informatik 2001: Wirtschaft und Wissenschaft in der Network Economy? Visionen und Wirklichkeit – Tagungsband Bd. 2 Bde*. Berlin : Springer, September 2001. – <http://tal.cs.tu-berlin.de/siefkes/texte/2002/Eso.html> – geprüft: 14. Juli 2002, S. 798–803
- [Skinner 1974] SKINNER, Burrhus F.: *Futurum Zwei ‘Walden Two’*. Die Vision einer aggressionsfreien Gesellschaft. Reinbek : Rowohlt, 1974
- [Snelting 1998] SNELTING, Gregor: Paul Feyerabend und die Softwaretechnologie. In: *Informatik Spektrum* 21 (1998), Oktober, Nr. 5, S. 273–276
- [Spinellis u. a. 1994] SPINELLIS, Diomidis ; DROSSOPOULOU, Sophia ; EISENBACH, Susan: Language and Architecture Paradigms as Object Classes: A Unified Approach Towards Multiparadigm Programming. In: GUTKNECHT, Jürg (Hrsg.): *Programming Languages and System Architectures*. Berlin, Heidelberg : Springer-Verlag, March 1994 (Lecture Notes in Computer Science 782), S. 191–207
- [Spohrer 2002] SPOHRER, Matthias: Zustandsorientierte Modellierung – Grundgedanken zur Umsetzung dieses Themas im Informatikunterricht der Mittelstufe. In: [Schubert u. a. 2002], S. 23–32
- [Stach 1997] STACH, Heike: The Image of Man as a Self-Steering Device as Constituent for the Concept of the Stored-Program-Computer. In: [Eulenhöfer u. a. 1997a], S. 4–11
- [Stachowiak 1973] STACHOWIAK, Herbert: *Allgemeinen Modelltheorie*. Wien : Springer, 1973

- [Städtler u. a. 1997] STÄDTLER, Klaus ; STACH, Heike ; KOREUBER, Mechthild ; EULENHÖFER, Peter ; BRAUN, Anette: Die Rekonstruktion von Orientierungsmustern in Fachtexten aus der Informatik – Ein methodisches Konzept für die Analyse von Wissenschafts- und Technikgeneseprozessen / TU Berlin – Fachbereich Informatik – IFP „Sozialgeschichte der Informatik“. 1997 ( 97-3). – Forschungsbericht. <http://tal.cs.tu-berlin.de/RoteReihe/RR97-03.rtf> – geprüft: 14. April 2002
- [Statistisches Bundesamt 2002] STATISTISCHES BUNDESAMT: *Zahl der Woche vom 6. August 2002 – Durchschnittsalter der Lehrer in Deutschland bei 47 Jahren.* August 2002. – <http://www.destatis.de/presse/deutsch/pm2002/zdw32.htm> – geprüft: 16. Dezember 2002
- [Staudte 1992] STAUDTE, Rainer: Funktionale Programmierung. In: **[Bosler 1992]**, S. 160–171
- [Steinbuch 1957] STEINBUCH, Karl: Informatik: Automatische Informationsverarbeitung. In: *SEG-Nachrichten (Technische Mitteilungen der Standard Elektrik Gruppe) – Firmenzeitschrift* (1957), Nr. 4, S. 171
- [Steinmetz 1999] STEINMETZ, Ralf: *Multimedia-Technologie: Grundlagen, Komponenten und Systeme.* 2. Aufl. Berlin : Springer, 1999
- [Streitberg u. a. 2000] STREITBERG, Sanna ; RUX, Martina ; DANIČIČ, Josef ; EMONTS-GAST, Martin ; GRUBERT, Volker ; HUMBERT, Ludger: *Grundlegende Unterrichtskonzepte der Informatik und ihre Umsetzung in der zweiten Phase der Lehrerinnenausbildung. Zur Verzahnung von Theorie und Praxis.* <http://didaktik.cs.uni-potsdam.de/HyFISCH/WorkshopLehrerbildung2000/Papers/Humbert.pdf>. Juli 2000. – Beitrag zum Workshop zur Lehrerausbildung, GI-Jahrestagung 2000, Berlin, 19. September 2000
- [Tenorth 2000] TENORTH, Heinz-Elmar: *Geschichte der Erziehung. Einführung in die Grundzüge der neuzeitlichen Entwicklung.* Weinheim : Juventa, 2000
- [Tesler 1984] TESLER, Lawrence G.: Programmiersprachen. In: *Spektrum der Wissenschaft, November*, 1984, S. 62–75
- [Thoma 2001] THOMA, Helmut: Wissen und Lernen. Was trägt die Informatik zum Unterricht bei? In: **[Desel 2001]**, S. 99–116
- [Thomas 2001] THOMAS, Marco: Die Vielfalt der Modelle in der Informatik. In: **[Keil-Slawik und Magenheim 2001]**, S. 173–186
- [Thomas 2002] THOMAS, Marco: *Informatische Modellbildung – Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht*, Universität Potsdam Didaktik der Informatik, Dissertation, Juli 2002. – [http://ddi.cs.uni-potsdam.de/Personen/marco/Informatische\\_Modellbildung\\_Thomas\\_2002.pdf](http://ddi.cs.uni-potsdam.de/Personen/marco/Informatische_Modellbildung_Thomas_2002.pdf) – geprüft: 26. Dezember 2002
- [Tolstoj 1862] TOLSTOJ, Leo N.: Die Schule von Jasnaja Poljana. In: **[Blankertz 1980a]**, S. 19–155. – 2. Auflage 1980
- [Tukey 1977] TUKEY, John W.: *Exploratory Data Analysis.* Reading, Massachusetts, USA : Addison-Wesley, 1977
- [Turing 1967] TURING, Alan M.: Kann eine Maschine denken? In: *Kursbuch 2* (1967), März, Nr. 8, S. 106–138
- [Valk 2002] VALK, Rüdiger: *Informatik als Methodendisziplin – am Beispiel interdisziplinärer Arbeit mit der Soziologie.* Februar 2002. – Positionspapier zur Arbeitstagung Bad Hersfeld 2002 Arbeitsgruppe: Informatik als Hybridwissenschaft. [http://waste.informatik.hu-berlin.de:80/peter/theorien-der-informatik/2002/pp\\_valk\\_020206.pdf](http://waste.informatik.hu-berlin.de:80/peter/theorien-der-informatik/2002/pp_valk_020206.pdf) – geprüft: 14. Juli 2002

- [Varela 1990] VARELA, Francisco J. ; BAECKER, Dirk (Hrsg.): *Kognitionswissenschaft, Kognitionstechnik: eine Skizze aktueller Perspektiven*. 1. Aufl. Frankfurt a. M. : Suhrkamp Verlag, 1990 (suhrkamp taschenbuch wissenschaft 882). – Originaltitel: Cognitive Science. A Cartography of Current Ideas 1988, übersetzt von Wolfram Karl Köck
- [Vaupel und Hoffmann 2001] VAUPEL, Wolfgang ; HOFFMANN, Bernd: *Ausstattung für das Lernen mit neuen Medien – Ein Leitfaden für Schulen und Schulträger . Unter Mitarbeit von Claudia Henrich-wark, Detlef Kaenders, Rainer Wulff*. Michelpresse, Düsseldorf. Juni 2001. – Nicht direkt zugreifbar. Über <http://www.e-nitiative.nrw.de/> -> e-dition.nrw [http://www.e-nitiative.nrw.de/ratgeber\\_ausstattung.php](http://www.e-nitiative.nrw.de/ratgeber_ausstattung.php) – geprüft 3. Mai 2002
- [Vohns 2000] VOHNS, Andreas: *Das Messen als fundamentale Idee im Mathematikunterricht der Sekundarstufe I*. Siegen : Universität-Gesamthochschule, Mai 2000. – Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung für das Lehramt für die Sekundarstufe I, <http://www.math.uni-siegen.de:80/didaktik/down1/messen.pdf> – geprüft: 9. September 2002
- [Vollmar 2000] VOLLMAR, Roland: *Von Zielen und Grenzen der Informatik / Universität Karlsruhe (TH)*. 2000. – Bericht. leicht erweiterte Fassung des zur 10-Jahres-Feier der Technischen Fakultät der Universität Bielefeld eingeladenen und am 12.5.2000 gehaltenen Vortrages <http://www.ubka.uni-karlsruhe.de/vvv/ira/2000/14/14.pdf> – geprüft 3. August 2002
- [Vollmer 1999] VOLLMER, Heribert: Was leistet die Komplexitätstheorie für die Praxis? In: *Informatik-Spektrum* 22 (1999), Nr. 15, S. 317–327
- [Vygotski 1978] VYGOTSKI, Lev S.: *Mind in Society. The Development of Higher Psychological Processes*. Cambridge, Mass. : Harvard University Press, 1978
- [Wagenschein 1982] WAGENSCHIN, Martin: *Verstehen lehren. Genetisch – Sokratisch – Exemplarisch*. 7. Aufl. Weinheim : Beltz, 1982
- [Wasserschaff 1995] WASSERSCHAFF, Markus: *Maßgeschneiderte Benutzerschnittstellen zur Unterstützung kooperativer Arbeit*. August 1995. – Diplomarbeit, Universität Bonn - Institut für Informatik
- [van Weert u. a. 1994] WEERT, Tom J. ; BOSLER, Ulrich ; GUBO, Sam ; TAYLOR, Harriet ; ABAS, Zoraini W. ; DUCHÂTEAU, Charles ; MOREL, Raymond ; WAKER, Peter ; IFIP (Hrsg.) ; UNESCO (Hrsg.): *Informatics for secondary education: a curriculum for schools*. Paris : UNESCO, 1994. – <http://unesdoc.unesco.org/images/0009/000973/097323e.pdf> : Produced by working party of the IFIP under auspices of UNESCO. Paris
- [van Weert u. a. 2000] WEERT, Tom J. ; BÜTTNER, Yvonne ; FULFORD, Catherine ; KENDALL, Mike ; DUCHÂTEAU, Charles ; HOGENBIRK, Pieter ; MOREL, Raymond ; IFIP (Hrsg.) ; UNESCO (Hrsg.): *Information and Communication Technology in Secondary Education – A Curriculum for Schools*. Original 1994. Paris : UNESCO, November 2000. – <http://www.edu.ge.ch/cptic/prospective/projets/unesco/en/curriculum2000.pdf> : Produced by working party of the IFIP under auspices of UNESCO. Paris
- [Wegner 1987] WEGNER, Peter: Dimensions of object-based language design. In: MEYROWITZ, Norman K. (Hrsg.): *Proceedings of the Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*. Orlando : ACM, December 1987. – <http://www.cs.ualberta.ca/~duane/courses/605/papers/p168-wegner.pdf> – geprüft: 31. Juli 2002, S. 168–182
- [Wegner 1997] WEGNER, Peter: Why Interaction is More Powerful than Algorithms. In: *CACM* 40 (1997), May, Nr. 5, S. 80–91. – <http://www.cs.brown.edu/people/pw/> – geprüft: 31. Juli 2002
- [Weizenbaum 1977] WEIZENBAUM, Joseph: *Die Macht der Computer und die Ohnmacht der Vernunft*. Frankfurt a. M. : Suhrkamp Verlag, 1977
- [Weizenbaum 1990] WEIZENBAUM, Joseph: Computer und Schule. In: [Weizenbaum 2001], S. 80–97

- [Weizenbaum 1992] WEIZENBAUM, Joseph: Die Sprache des Lernens. In: [Weizenbaum 2001], S. 72–79
- [Weizenbaum 2001] WEIZENBAUM, Joseph ; WENDT, Gunna (Hrsg.) ; KLUG, Franz (Hrsg.): *Computer-macht und Gesellschaft – Freie Reden*. Frankfurt a. M. : Suhrkamp Verlag, 2001
- [von Weizsäcker 1971] WEIZSÄCKER, Carl F.: *Die Einheit der Natur*. München : Carl Hanser Verlag, 1971
- [Werner 2001] WERNER, Peter: Rollenmodellierung. In: MEHNER, Katharina (Hrsg.) ; MEZINI, Mira (Hrsg.) ; PULVERMÜLLER, Elke (Hrsg.) ; SPECK, A. (Hrsg.): *Aspektorientierung – Workshop der GI-Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung – 3./4. Mai 2001*. Paderborn : Universität, Mai 2001 (Technischer Bericht tr-ri-01-223). – <http://i44w3.info.uni-karlsruhe.de/~pulvermu/workshops/aop2001/beitraege/werner.pdf> – geprüft: 10. August 2002
- [Wessner und Pfister 2001] WESSNER, Martin ; PFISTER, Hans-Rüdiger: Kooperatives Lehren und Lernen. In: [Schwabe u. a. 2001], S. 251–263
- [Westram 1999] WESTRAM, Hiltrud: Schule und das neue Medium Internet – nicht ohne Lehrerinnen und Schülerinnen! Universität Dortmund - Fachbereich Erziehungswissenschaften und Biologie. Januar 1999. – Dissertation <http://eldorado.uni-dortmund.de:8080/FB12/inst3/forschung/1999/westram> – geprüft: 16. Dezember 2002
- [Whitehead 1962] WHITEHEAD, Alfred N.: Die Gegenstände des mathematischen Unterrichts. In: *Neue Sammlung* (1962), Nr. 2/3, S. 257–266. – Original: The Mathematical Curriculum, 1913 – übersetzt von Alexander Israel Wittenberg
- [Wilkens 2000] WILKENS, Ulrike: *Das allmähliche Verschwinden der informationstechnischen Grundbildung. Zum Verhältnis von Informatik und Allgemeinbildung*. Aachen : Shaker, 2000 (Berichte aus der Informatik). – Zugl.: Dissertation an der Universität Bremen, 1999
- [Williams 1984] WILLIAMS, G.: Software Frameworks. In: *Byte* 9 (1984), December, Nr. 13, S. 124–127, 394–410. – Lisa Toolkit
- [Wilson u. a. 1993] WILSON, James W. ; FERNANDEZ, Maria L. ; HADAWAY, Nelda: Synthesis of Research on Problem Solving (Chap. 4). In: WILSON, P. S. (Hrsg.): *Research Ideas for the Classroom: High School Mathematics*. New York : MacMillan, 1993. – <http://jwilson.coe.uga.edu/emt725/PSSyn/PSSyn.html> – geprüft: 18. Januar 2003, S. 57–78
- [Winograd und Flores 1986] WINOGRAD, Terry ; FLORES, Fernando: *Understanding Computers and Cognition - A New Foundation for Design*. New York : Ablex Publ. Norwood, 1986. – Deutsch: „Erkenntnis Maschinen Verstehen“, mit einem Nachwort von W. Coy, Rotbuch, Berlin, 1989
- [Wirth 1971] WIRTH, Niklaus: Program Development by Stepwise Refinement. In: *CACM* 14 (1971), April, Nr. 4, S. 221–227
- [Wirth 1974] WIRTH, Niklaus: On the Composition of Well-Structured Programs. In: *Computing Surveys* 6 (1974), December, Nr. 4, S. 247–259
- [Wirth 1978] WIRTH, Niklaus: *Systematisches Programmieren – Eine Einführung*. Stuttgart : B.G. Teubner, 1978
- [Wirth 1985] WIRTH, Niklaus: *Programmieren in Modula-2*. 1. dt. Aufl. Berlin : Springer, 1985
- [Wirth 1988] WIRTH, Niklaus: Type Extensions. In: *ACM Transactions on Programming Languages and Systems* 10 (1988), April, Nr. 2, S. 204–214. – <http://www.cs.colorado.edu/~diwan/5535/assignments/p204-wirth.pdf> – geprüft: 1. August 2002
- [Wirth 1992] WIRTH, Niklaus: Geleitwort. In: [Mössenböck 1992], S. V–VI

- [Wirth 1994] WIRTH, Niklaus: Gedanken zur Software-Explosion. In: *Informatik Spektrum* 17 (1994), Februar, Nr. 1, S. 5–10. – <http://wwwicg.informatik.uni-rostock.de/Lehre/Informatik3/SS2002/Sonstiges/SoftwareWirth.html> – geprüft: 20. September 2002
- [Wirth 1999] WIRTH, Niklaus: An Essay on Programming / Eidgenössische Technische Hochschule, Institut für Computersysteme. Zürich, March 1999 ( 315 ). – Forschungsbericht
- [Witten und Penon 1997] WITTEN, Helmut ; PENON, Johann: Unterrichtlicher Einsatz der Telekommunikation. Erfahrungen und Perspektiven aus der Sicht des Informatikunterrichts. In: [Hoppe und Luther 1997], S. 164–175
- [Wittmann 1981] WITTMANN, Erich C.: *Grundfragen des Mathematikunterrichts*. 6. neu bearbeitete Aufl. Braunschweig : Friedrich Vieweg, 1981
- [Wittmann und Müller 1988] WITTMANN, Erich C. ; MÜLLER, Gerhard: Wann ist ein Beweis ein Beweis? In: BENDER, P. (Hrsg.): *Mathematikdidaktik. Theorie und Praxis. Festschrift für Heinrich Winter*. Bielefeld, 1988. – <http://www.didmath.ewf.uni-erlangen.de/Verschie/Wittmann1/beweis.htm> – geprüft: 23. Mai 2002, S. 237–257
- [Wolff u. a. 1999] WOLFF, Bernd ; FUCHS-KITTOWSKI, Klaus ; KLISCHEWSKI, Ralf ; MÖLLER, Andreas ; ROLF, Arno: Organisationstheorie als Fenster zur Wirklichkeit. In: BECKER, Jörg (Hrsg.) ; KÖNIG, Wolfgang (Hrsg.) ; SCHÜTTE, Reinhard (Hrsg.) ; WENDT, Oliver (Hrsg.) ; ZELEWSKI, Stephan (Hrsg.): *Wirtschaftsinformatik und Wissenschaftstheorie. Bestandsaufnahme und Perspektiven*. Wiesbaden, 1999. – <http://swt-www.informatik.uni-hamburg.de/publications/download.php?id=141>, S. 291–330
- [Wolters 1994] WOLTERS, Angelika: Projekt- und Fächerübergreifender Unterricht. In: BOVET, Gislinde (Hrsg.) ; HUWENDIEK, Volker (Hrsg.): *Leitfaden Schulpraxis - Pädagogik und Psychologie für den Lehrberuf*. Berlin : Cornelsen, 1994. – ISBN 3-464-49116-1, S. 157–196
- [Wulf 1972] WULF, Christoph: Evaluation – ein kritischer Überblick. In: *Neue Sammlung. Göttinger Zeitschrift für Erziehung und Gesellschaft* (1972), Nr. 3, S. 259–284
- [Würmli 1992] WÜRMLI, M. (Hrsg.): *Der Kinderbrockhaus in vier Bänden*. Mannheim : F. A. Brockhaus GmbH, 1992
- [Zemanek 1971] ZEMANEK, Heinz: Was ist Informatik? In: *Elektronische Rechenanlagen* 13 (1971), Nr. 4, S. 157–161
- [Züllighoven 2001] ZÜLLIGHOVEN, Heinz: Softwareentwicklung. In: [Schwabe u. a. 2001], S. 98–107
- [Zuse 1999] ZUSE, Horst: Anmerkungen zum John-von-Neumann-Rechner. In: *FIF-Kommunikation* 16 (1999), Juni, Nr. 2, S. 10–19
- [Zuse 1984] ZUSE, Konrad: *Der Computer – Mein Lebenswerk*. 1. Aufl. Berlin : Springer, 1984



# Tabellenverzeichnis

2.1	Auffassungen und ihre Ausprägung in Sprachklassen . . . . .	21
2.2	Basiskonzepte und Sichten der Software-Entwicklung . . . . .	27
2.3	Notationsmöglichkeiten von Basiskonzepten . . . . .	28
3.1	Formen fächerübergreifenden Unterrichts . . . . .	34
3.2	Geschichtliche Phasen der Projektmethode . . . . .	44
3.3	Merkmale des Projektunterrichts . . . . .	45
3.4	Schritte und Merkmale eines Projektes . . . . .	46
3.5	Praktizierte Unterrichtsformen . . . . .	50
4.2	Empfehlungen der Fachwissenschaft zur Didaktik – Quellen und Einordnung . . . . .	61
4.4	Ausgewählte deutsche fachdidaktische Veröffentlichungen – Quellen und Einordnung . . . . .	72
4.5	Social, Ethical, and Professional Context . . . . .	74
5.1	Arbeitshypothesen für die Expertinnenbefragung . . . . .	84
5.2	Phasierung der Expertinneninterviews . . . . .	85
5.3	„offene“ Fragen – Zuordnung zu Interviewphasen . . . . .	86
5.4	Informatikdidaktische Aktivitäten der Expertinnen . . . . .	88
5.5	Auswertung zu vorgegebenen Orientierungen . . . . .	94
5.6	Auswertung zur fachlichen Basis . . . . .	95
6.1	Arbeitshypothesen und Fragestellungen – Kreuzreferenzen . . . . .	104
6.2	Modulkonzept – Beispielübersicht . . . . .	110
7.1	Zuordnung der Sichten der Informatik im zeitlichen Verlauf . . . . .	119
7.2	Schülerinnen im Längsschnitt: Zuordnung der Sichten der Informatik im zeitlichen Verlauf . . . . .	123
8.1	Informatik – Bayern – Lerninhalte der 6. Jahrgangsstufe laut Lehrplan . . . . .	129
8.2	Hospitationen im Bayerischen Schulversuch – Themenübersicht . . . . .	130
8.3	Lerninhalte Sekundarstufe I × Module . . . . .	134
9.1	Anforderungen und Beispiele – Kreuzreferenzen . . . . .	145

---

11.1	Didaktische Ansätze zur Programmierung in der Hochschulinformatik (international) . . .	153
11.2	Untersuchungsdimensionen des Informatikunterrichts . . . . .	154
E.1	Hospitationen im Bayerischen Schulversuch – Themenübersicht . . . . .	187
F.1	Stoffverteilungsplan Informatikgrundkurs – erstes Halbjahr 11. Jahrgang . . . . .	198
F.2	Das EVA-Prinzip im unterrichtlichen Kontext der Realisierung von Automaten . . . . .	200
F.3	Automatentafel für MuesliAutomat . . . . .	200
F.4	Unterrichtsverlauf – objektorientierte Beschreibung des Computerarbeitsplatzes . . . . .	213
F.5	Unterrichtsverlauf – objektorientierte Modellierung - Kontrollstrukturen . . . . .	215
F.6	Unterrichtsverlauf – Arbeit mit einem webbasierten CSCW zur Kursplanung . . . . .	216
F.7	Unterrichtsverlauf – Fachkonzept Keller – Realisierung als Klasse . . . . .	218
F.8	Unterrichtsverlauf – Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient . . . . .	221

# Abbildungsverzeichnis

2.1	Zeitleiste zu Innensichten der Informatik . . . . .	19
2.2	Entwicklung von Abstraktionsmechanismen in Programmiersprachen . . . . .	22
3.1	Datenschema Methode . . . . .	34
3.2	»Didaktisches Dreieck« . . . . .	35
3.3	Hierarchiemodell nach Wolfgang SCHULZ . . . . .	36
3.4	Methoden/Konzepte des Unterrichts . . . . .	41
3.5	Schema zum Problemlöseprozess . . . . .	43
3.6	Unterrichtsformen/Sozialformen des Unterrichts . . . . .	49
4.1	Neubewertung der Fachstruktur . . . . .	58
4.2	Zeitleiste zu didaktischen Orientierungen . . . . .	78
5.1	Strukturübersicht 1.–5. Kapitel . . . . .	83
5.2	Anzahl der Schulwechsel . . . . .	87
5.3	Expertinnen – Geburtsjahrgänge . . . . .	87
6.1	Strukturübersicht 1.–6. Kapitel . . . . .	103
6.2	Daten – Wissen – Information . . . . .	108
7.1	Strukturübersicht 6.–7. Kapitel . . . . .	115
7.2	Bild der Informatik bei Schülerinnen: Das Schulfach Informatik ... . . . . .	118
7.3	Bild der Informatik bei Schülerinnen: Informatik ist... . . . . .	118
7.4	Bild der Informatik bei Schülerinnen: Position zu Behauptungen . . . . .	120
7.5	Schülerinnen im Längsschnitt: Das Schulfach Informatik ... . . . . .	122
7.6	Schülerinnen im Längsschnitt: Informatik ist ... . . . . .	122
7.7	Schülerinnen im Längsschnitt: Position zu Behauptungen... . . . . .	123
7.8	Änderung des Bildes der Informatik bei Schülerinnen (schematisch) . . . . .	125
8.1	Strukturübersicht 6.–8. Kapitel . . . . .	127
9.1	Strukturübersicht 6.–9. Kapitel . . . . .	137

---

9.2	Einflussfaktoren Kapitel 9 . . . . .	138
10.1	Überblick Forschungsgang . . . . .	149
10.2	Übersicht zur Entwicklung und Evaluation des Modulkonzepts . . . . .	150
10.3	Schlussfolgerungen aus dem ersten Evaluationszyklus des Modulkonzepts . . . . .	151
11.1	Fachdidaktik, Schulinformatik und Hochschulinformatik . . . . .	153
11.2	Schülerin und Bezüge zum Informatikunterricht . . . . .	155
C.1	Fragebogen – Leitbilder der Informatik . . . . .	172
F.1	MuesliAutomat – Ausgabefunktion als Struktogramm . . . . .	201
F.2	MuesliAutomat – Zustandsübergänge als Struktogramm . . . . .	201
F.3	Folie – Arbeit mit einem webbasierten CSCW zur Kursplanung . . . . .	217
F.4	Tafel – Fachkonzept Keller – Realisierung als Klasse . . . . .	218
F.5	Folie/Tafel – Nebenläufigkeit zur Realisierung der „Parallelität“ von Server und Klient . . . . .	221

# Liste der Algorithmen

1	Schnittstelle AutomatIO . . . . .	199
2	Prozeduren AusgabeFunktion und Uebergang . . . . .	202
3	Verbale Beschreibung: MuesliriegelAutomat . . . . .	202
4	Modul MuesliriegelAutomat . . . . .	203
5	Modul GetraenkeAutomat2 . . . . .	205
6	Böswilliger Safe . . . . .	206
7	Passwortschnittstelle . . . . .	206
8	Schnittstelle KlasAutomatIO . . . . .	206
9	Modul ReelleZahlenAkzept . . . . .	207
10	Schnittstelle Stapel . . . . .	208
11	Klasse Keller (implementiert als Liste in Python) . . . . .	219
12	UDFsocket.py – didaktisch gestaltete Schnittstelle . . . . .	222
13	UDFserver_klient.py – „gleichzeitig“ Server und Klient mit einem Skript . . . . .	223
14	UDFserver_klient_plus_GUI.py – Server und Klient mit einer Benutzungsoberfläche . . . . .	224



# Namens- und Organisationsindex

- ACM – Association for Computing Machinery, 6, 18, 73  
Adam, Adolf (\*1918), 16  
AEDS – Association for Educational Data Systems, 6  
Aiken, Howard (1900-1973), 10  
Anders, Günther (1902-1992), 83  
Anderson, John R., 37  
Arbeitsgruppe zur Fachdidaktik Informatik, 66  
Arlt, Wolfgang (\*1934), 56, 61  
ATEE – Association for Teacher Education in Europe, 6
- Babbage, Charles (1791-1871), 12  
Backus, John Warner (\*1924), 12  
Balzert, Heide, 27  
Balzert, Helmut, 27, 55, 61  
Bauer, Friedrich Ludwig (\*1924), 10, 17  
Baumann, Rüdiger, 64, 65, 72  
Baumert, Jürgen, 52, 53  
Berger, Peter, 15, 68, 72, 151  
Bittner, Peter, 16  
Blankertz, Herwig (1927-1983), 32  
BLK – Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung, 62  
Bloch, Ernst (1885-1977), 103  
Blonskij, Paval Petrovič (1884-1941), 33  
BMBF – Bundesministerium für Bildung und Forschung, 151, 152  
BMBF – Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie, 49  
Brauer, Ute, 58, 61, 119, 179  
Brauer, Wilfried (\*1937), 10, 17, 18, 58, 61, 119, 179  
Brinda, Torsten (\*1972), 70, 72  
Brooks, Frederick P., 57, 61  
Broy, Manfred (\*1949), 25, 67, 72  
Bruner, Jerome Seymour (\*1915), 3, 4, 38, 47, 112  
Brunnstein, Klaus, 142  
Bussmann, Hans, 27  
Büttemeyer, Wilhelm, 16
- Capurro, Rafael, 18
- Chomsky, Avram Noam (\*1928), 37, 38, 206, 208  
Church, Alonzo (1903-1995), 12, 208  
Claus, Volker (\*1944), 17, 56, 61  
Comenius, Johann Amos (1592-1670), 32  
Coy, Wolfgang, 18, 73
- Dahl, Ole-Johann (1931-2002), 23  
Denning, Peter J., 18  
Dewey, John (1859-1952), 44  
DFG – Deutsche Forschungsgemeinschaft, 15  
Dijkstra, Edsger Wybe (1930-2002), 19  
DIN – Deutsches Institut für Normung, 201
- Eberle, Franz, 73, 75  
EDK – Eidgenössische Erziehungsdirektorenkonferenz, 74  
Engbring, Dieter, 66, 72  
Epiktet (50–138), 268
- FBI – Fachbereich Informatik, 142  
Fischer, Roland (\*1945), 3  
Flores, Fernando, 208  
Floyd, Christiane (\*1943), 10, 18, 20, 59, 61  
Freire, Paulo (\*1921), 115  
Frey, Karl (\*1946), 45  
Friedrich, Steffen, 66, 72  
Frühwald, Wolfgang (\*1935), 15
- Gagné, Robert Mills (\*1916), 39  
Gallin, Peter, 48  
Gamm, Joachim, 31  
Gaudig, Hugo (1860-1923), 33  
Genrich, Hartmann J., 17, 119, 179  
GI – Gesellschaft für Informatik e. V., 6, 47, 55, 65, 138  
von Glasersfeld, Ernst (\*1917), 39  
Goldberg, Adele, 23, 25  
Goos, Gerhard, 56, 61  
Görlich, Christian F., 35  
Gorny, Peter, 6  
Gruska, Jozef, 14  
Gudjons, Herbert (\*1940), 46, 47  
Guzdial, Mark, 141
- Hartmann, Werner, 152

- Herbart, Johann Friedrich (1776-1841), 32  
 Herzig, Bardo, 70, 72  
 Hesse, Hermann (1877-1962), V  
 Heymann, Hans-Werner, 27  
 Hopcroft, John, 208  
 Hubwieser, Peter (\*1955), 67, 72  
 Humbert, Ludger (\*1955), 69, 70, 72
- IBM – International Business Machines Corporation, 46  
 IEA – International Association for the Evaluation of Educational Achievement, 49  
 IFIP – International Federation for Information Processing, 6, 73, 74, 76  
 Illich, Ivan (\*1926-2002), 137  
 INFOS – Informatik und Schule, 130
- Kaasbøll, Jens J., 153, 154  
 Kay, Alan Curtis (\*1940), 23  
 Kerschensteiner, Georg (1854-1932), 33  
 Kilpatrick, William Heard (1871-1965), 44  
 Klafki, Wolfgang (\*1927), 27, 31, 64  
 KM – Kultusministerium, 164  
 KMK – Kultusministerkonferenz, 62  
 Knoll, Michael, 44, 46  
 Knöß, Petra, 63, 72  
 Krämer, Sybille, 66, 72  
 Kuhn, Thomas S. (1922-1996), 5, 21, 32
- Lehmann, Eberhard, 80  
 Linkweiler, Ingo, 144  
 Linneweber-Lammerskitten, Helmut (\*1951), 14  
 Lockemann, Peter C., 57, 61  
 Lovelace, Ada Augusta (1815-1852), 12  
 Luft, Alfred L., 18
- Magenheim, Johannes, 69, 72  
 Marx, Karl (1818-1883), 33  
 Maturana, Humberto R. (\*1928), 39  
 McCarthy, John (\*1927), 12  
 Meyer, Hilbert (\*1941), 3, 39, 42, 47, 48  
 Meyer, Meinert A. (\*1941), 35  
 Montessori, Maria (1870-1952), 47  
 Mössenböck, Hans-Peter, 22  
 MSWF – Ministerium für Schule, Wissenschaft und Forschung, 130  
 Münker, Stefan, 14
- Nake, Frieder (\*1938), 13  
 Nassi, Isaac, 201  
 Naur, Peter (\*1928), 12  
 von Neumann, John (1903-1957), 10, 11  
 Nievergelt, Jürg, 58, 61, 152  
 Nygaard, Kristen (1926-2002), 17, 23
- Oberquelle, Horst, 143  
 OECD – Organisation for Economic Co-operation and Development (dt. Organisation für wirtschaftliche Zusammenarbeit und Entwicklung), 49, 52  
 Otto, Berthold (1859-1933), 32
- PARC – Palo Alto Research Center, 23  
 Parnas, David Lorge (\*1941), 56, 61  
 Penon, Johann (\*1948), 67, 72  
 Penrose, Roger, 208  
 Peschke, Rudolf, 63, 72  
 Pestalozzi, Johann Heinrich (1746-1827), 47  
 Petri, Carl Adam (\*1926), 10, 17, 119, 179  
 Piaget, Jean (1896-1980), 38–40  
 Pólya, George (1887-1985), 43, 79  
 Popper, Karl R. (1902-1994), 37
- Rechenberg, Peter, 18  
 Richards, Charles R., 44  
 Robson, David, 23  
 Roth, Heinrich (1906-1983), 42  
 Ruf, Urs, 48
- Schöning, Uwe (\*1955), 60, 61  
 Schubert, Sigrid (\*1950), 63, 66, 72  
 Schulmeister, Rolf, 40  
 Schulte, Carsten, 69, 70, 72  
 Schulz, Wolfgang (1929-1993), 35  
 Schwill, Andreas, 4, 64–66, 72  
 Sezgen, Beytullah, 144  
 Shannon, Claude Elwood (1916-2001), 10  
 Shaw, Mary, 59, 61  
 Shneiderman, Ben (\*1947), 201  
 Siefkes, Dirk, 13, 25  
 Skinner, Burrhus Frederic (1904-1990), 37  
 Snelting, Gregor, 15  
 Soloway, Elliot, 141  
 Stachowiak, Herbert, 70  
 Steinbuch, Karl (\*1917), 9  
 Suin de Boutemard, Bernhard, 45
- Thomas, Marco, 26, 70–72  
 Tolstoj, Leo N. (1828-1910), 55  
 Turing, Alan Mathison (1912-1954), 10, 208
- UNESCO – United Nations Educational, Scientific and Cultural Organization, 73, 74, 76
- Varela, Francisco J. (1946-2001), 11, 39  
 Vollmar, Roland, 14  
 Vygotsky, Lev Semjonowitsch (1896-1934), 39
- W3C – World Wide Web Consortium, 140

- Wagenschein, Martin (1896-1988), 41  
van Weert, Tom, 74  
Wegner, Peter, 24  
Weizenbaum, Joseph (\*1923), 57, 59, 61  
von Weizsäcker, Carl Friedrich (\*1912), 16  
Whitehead, Alfred North (1861-1947), 4  
Wiener, Norbert (1894-1964), 11  
Wilson, James W., 43  
Winograd, Terry, 208  
Wirth, Niklaus (\*1934), 22, 25, 59, 61  
Witten, Helmut (\*1945), 67, 72  
Wittgenstein, Ludwig (1889-1951), 37  
Wittmann, Erich, 38
- Zemanek, Heinz (\*1920), 16  
Zuse, Konrad (1910-1995), 10, 11



# Sachindex

- 4GL – 4th Generation Language, 21
- Abbilddidaktik, 3
- Abitur, 4, 62, 89, 111, 136, 141
  - Bedingungen, 62
- Ablauforganisation, 211
- Abstraktion, 24
  - Mechanismen, 20
- Accounting, 134, 210
- Adoleszenz, 32
- ADT – Abstrakter Datentyp, 209
  - Schlange, 209
- Akkomodation, 40
- Akquisition von Daten, 108
- Aktivierung
  - kognitiv, 53
- Akzeptor, 198, 204
- ALGOL, 12
  - Algorithmic Language, 12
  - Sprachfamilie, 12, 23
- Algorithmen, 12
- Algorithmen und Datenstrukturen, 20
- algorithmische Zeichen, 13
- Algorithmus
  - parametrisierter, 20
- Allgemeinbildung, 5
- allgemeine Bildung, 3
- Allokation, 31
- Alltagssituationen, 52
- Analyse
  - Top-Down, 43
- Analytical Engine, 12
- Anfangsunterricht mit Automatentheorie, 197
- Anforderungen
  - gesellschaftliche, 3
- Anpassungsleistung, 38
- anregungsarmer Unterricht, 53
- Anschlussfähigkeit, 52
- anspruchsvoller Unterricht, 53
- Anwendungsfallanalyse, 23
- Anwendungsfälle, 109
- Anwendungsmodell, 13
- API – Application Program(ming) Interface, 140
- Äquilibration, 40
- Arbeitsformen, 34
- Arbeitshypothese, 29, 54, 60, 78
  - Lerntheorie, 54
  - Wissenschaftstheorie und Informatik, 29
- Arbeitsschule, 33
  - bewegung, 47
  - historisch, 32
- Arbeitsweise
  - explorativ, 210
  - professionell, 46
- Architektur, 65
- Artikulation, 32
- Assimilation, 40
- Aufbauorganisation, 211
- Auffassungen von der Wissenschaft Informatik,
  - 16
- aufgabenorientierte Sicht, 22
- Aufwandsabschätzung – Komplexität, 56
- Ausbildung, 31
  - berufsbezogen, 43
- Ausgangsüberlegungen, 3–7**
- Auswertung von Ausdrücken, 21
- Automat
  - endlicher, 141, 198
- Automaten mit „Gedächtnis“, 208
- Automatengraph, 112, 198, 204
- Automatentabelle, 198
- Automatentafel, 200
- Automatentheorie, 209
- Automatisierung von Symboloperationen, 66
- autooperationale Form, 13, 29
- basale Kulturwerkzeuge, 52
- Basiskompetenz, 52
  - fächerübergreifend, 52
- Basiskonzepte, informatische, 19
- Bayern, 127
  - Pflichtfach Informatik, 128
- Beantwortung von Anfragen, 21
- Bedienwissen, 139
- Befragung
  - schriftlich, 116
- Befragung zur Prädikativen Modellierung, 167–*
  - 170
- Begabungsförderung, 135
- Behaviorismus, 37

- beibringen, 35  
 Beiträge zur Fachdidaktik, 85  
 Benchmarking  
   inhaltlich, 52  
 Berechenbarkeit, 208  
 Berechnungsagenten, 24  
 Berlin, 140  
 Berliner [Didaktik-]Modell, 35  
 Berliner Modell, 35, 37  
 berufliche Bildung, 43  
 Berufsausbildung, 44  
 berufsbezogenen Ausbildung, 43  
 Berufskolleg, 144  
 Betriebssystem, 106  
   -funktionen automatisieren, 106  
   Oberon, 141  
 Betriebssystemstreit, 4  
 Bezugsnorm, 51  
 Bezugswissenschaft, 3  
*Bild der Schülerinnen von der Informatik, 171–177*  
 Bildung, 31  
   allgemeine, 3  
   berufliche, 43  
   Informatische, 1, 55, 59, 67, 72, 78, 104, 118, 119, 141  
   anwendungsorientiert, 59  
   Leitlinien, 69  
   Neuorientierung, 63  
   Wissenschaftspropädeutik, 62  
 Bildungsbemühungen, 153  
 Bildungsgut, 26  
 Bildungsökonomie, 32  
 Bildungsplanungsmaßnahmen – Bundesrepublik Deutschland, 62  
 bildungstheoretisches Konzept, 52  
 Bionik, 16  
 Black Box, 108  
 Bloomsche Taxonomien, 37, 153  
 BNF – Backus-Naur-Form, 12  
 böswilliger Safe, 204, 206  
 Bottom-Up, 22  
 Bremen, 140  
 BSCW – Basic System Cooperative Workspace, 59, 68, 110  
 Buchschule, 33, 38  
  
 C & U – Computer & Unterricht (Zeitschrift), 164  
 CACM – Communications of the ACM, 229  
 CASE – Computer Aided Software Engineering, 163  
 CCC – Cross-Curricular Competencies, 52, 79  
 Checklisten, 84  
  
 Chomsky-Grammatik  
   Komplexität von Sprachen, 38  
   Typ 2, 208  
   Typ 3, 206  
 Choreographie, 32  
 Church'sche Hypothese, 14, 208  
 Church-Turing-These, 14  
 CLOS – Common LISP Object System, 21  
 Cluster, 53  
 Compiler, 25  
 Compilerbau, 209  
 Computer  
   Vertrautheit, 52  
 Computer Literacy, 52  
 computer science  
   Definition der ACM, 73  
 Computer-Weltbild von Informatiklehrerinnen, 68  
 Computerartefakt, 13  
 Computerfachraum, 4  
 Computerkabinett, 4  
 Computerraum, 4  
 CRC – Class-Responsibility-Collaboration, 23, 166, 226  
   Karten, 23, 146  
 Cross-Curricular Competencies, 52, 79  
 CSCCL – Computer supported cooperative learning, 106, 110, 211  
 CSCW – Computer supported collaborative work, 106, 211  
   im Informatikunterricht, 215  
 curriculare Validität, 52  
 Curriculum, 38  
   spiralförmig, 38  
   spiralförmig, 105  
 CUU – computerunterstützter Unterricht, 63  
  
 Daten  
   Akquisition, 108  
 Datenabstraktion, 20  
 Datenbankschnittstelle, 109  
 Datenschema, 34  
 Datenstruktur, 12  
   allgemeine, 20  
   Keller, 208  
   Schlange, 209  
 Datentyp, 11  
 Dekontextualisierung, 13  
 Delegation, 24  
 Denkzeug, 67, 78  
 Diagnostik, 53  
 Didaktik, 32  
   kritisch-konstruktive, 31  
   lehrtheoretische, 35  
   lerntheoretische, 35, 37

- lernzielorientiert, 37
- Spiralprinzip, 62
- Didaktik der Informatik
  - Begründung für das Schulfach, 56
  - dokumentierte Unterrichtserfahrungen, 63
  - Einbeziehung theoretischer Elemente in das Schulfach, 63
  - geeignete Systemumgebungen für die Schule, 63
  - Legitimation des Schulfachs, 65
  - Modulkonzept, 62
  - Nichtdeterminismus, 63
  - Synergieeffekte zwischen Unterrichtsfächern, 18
  - tragfähige Konzepte, 63
  - wissenschaftliche Studien, 55
- didaktische
  - Gestaltung, 62, 68
  - Grundhaltung, 75
  - Grundorientierung, 37
  - Innovation, 68
  - Konzepte, 69
    - Modellierung, 33
  - Strukturierung, 79
  - Zugänge, 69
- Didaktische Gestaltung, 62
- Didaktische Grundorientierung, 37
- didaktische Prinzipien, 32
- Didaktische Reduktion, 62
- didaktisches Dreieck, 35
- DIN 66 261, 201
- Diplomarbeit, 142
- direktiver Unterricht, 53
- Direktivität, 53
- Diskretisieren, 13
- Diskussion – international 1993-1997, 73
- Dokumente
  - Versand, 134
  - Verwaltung, 134
- Dokumentenbeschreibungssprache, 109, 211
- dokumentierte Schnittstellen, 140
- DTD – Document Type Definition, 144
- DVA – Datenverarbeitungsanlage, 80
  
- E-Mail, Empfangen und Senden, 210
- ECDL – Europäischer Computer-Führerschein (European Computer Driving Licence), 76, 139
- EDSAC – Electronic Delay Storage Automatic Calculator, 12
- EDVAC – Electronic Discrete Variable Calculator, 11
- Eingliederung, 31
- Einlogvorgang, 210
  
- Einschätzung der Informatik durch Lernende, 115–125**
- Einzelfallstudie
  - Bild der Informatik, 116
- elektronisches Publizieren, 210
- empirische Ergebnisse, 7
- empirische Sozialforschung, 128
- enaktiv, 38, 47, 112
- endliche Automaten, 20, 141
- England, 128
- ENIAC – Electronic Numerical Integrator and Calculator, 12
- Enkulturation, 26, 71
- entdeckendes Lernen, 47
- Entmystifizierung, 66
- Entscheidungsfindung, 20
- Entscheidungsproblem, 208
- Entwicklungsstufenmodell
  - Piaget, 38
- Entwirklichung, 47
- Entwurfsmuster, 20, 107
  - MVC, 107, 139
- EPA – Einheitliche Prüfungsanforderungen in der Abiturprüfung, 136
- ER – Entity-Relationship, 163
- ereignisgetriebene Systeme, 109
- Erfahrungsmöglichkeiten
  - aktiv, 47
- erfolgreiche Lernprozesse, 54
- Ergebnisse
  - empirische, 7
- ERM – Entity Relationship Model, 107
- Erstbegegnung, 38
- erste Programmiererin, 12
- Ertragsindikatoren, 52
- Erziehungstechnologie, 32
- Erziehungswissenschaft, 31, 128
  - Ausdifferenzierung, 128
  - Autonomisierung, 128
  - kritische, 128
- Europäisches Gymnasium, 128
- EVA – Eingabe Verarbeitung Ausgabe, 199
  - Prinzip, 198
- Evaluation, 33, 52, 105, 128, 143
  - Geschichte des Begriffs, 128
  - unterrichtsbegleitend, 143
- evolutionäre Systementwicklung, 59
- exemplarisch, 41
- Exemplarische Unterrichtseinheiten, 197–223*
- Expertinnen
  - befragung
    - Arbeitshypthesen, 84
  - interview
    - Ergebnisse, 87

- des Informatikunterrichts, 83
- Geburtsjahrgänge, 87
- Informatikdidaktische Aktivitäten, 88
- Konzeptionelle Orientierungen des Informatikunterrichts, 94
- Schulwechsel, 87
- Expertise, 7
- Expertise durch Interviews – Übersicht und Ergebnisse, 161–166*
- explorativer Untersuchungsansatz, 116
- Fachdidaktik Mathematik, 3
- Fachdidaktische Gespräche, Königstein, 85
- fachdidaktische Kenntnisse, 4
- fachdidaktisches Konzept, 130
- fächerkoordinierend, 34
- fächerübergreifend, 46, 79, 105, 108
  - er Unterricht, 34
  - Basiskompetenz, 52
  - Kompetenzen, 52
- fächerüberschreitend, 34
- fächerverbindendes Lernen, 34
- fächerverknüpfend, 34
  - er Unterricht, 57
- Fachgruppe für Informatiklehrerinnen, 85
- Fachinformatiker, 142
- Fachkonferenz Informatik, 111
- Fachkonzept, 130, 139
- fachliche Stärken, 105
- Fachsystematik, 62
- Fachtagung Informatik und Schule (INFOS), 63
- Fallstudien, 6
- Featuritis, 143
- Fenster zur Wirklichkeit, 13
- FK IF – Fachkonferenz Informatik, 164
- Form
  - autooperationale, 13
  - operationale, 13
- formale Sprachen, 208
- Formalstufen des Unterrichts, 32
- Formen fächerübergreifenden Unterrichts, 34
- Forschungsansätze, 6
- Forschungsfragen – Eberle, 76
- Forschungsgegenstand, 5
- Forschungsmethodik
  - Vorüberlegungen, 6
- Fortsetzbarkeit als Prinzip, 38
- fragend-entwickelnder Unterricht, 53
- Fragestellungen, 78
  - wissenschaftliche, 78
- fundamentale Idee, 4, 38, 41, 48, 51
  - Algorithmisierung, 64
  - Darstellung als Baumstruktur, 64
  - der Informatik, 4
    - in der Informatik, 57
- Informatik – Diskussion, 64
- Konkurrenz, 57
- Konsistenz, 57
- Kritik, 66
- Mathematikunterricht, 63
- Persistenz, 57
- Sprache, 64
  - strukturierte Zerlegung, 64
  - Teamarbeit, 64
- funktionale Modellierung, 93, 109
- Funktionen der Schule, 31
- G8 – Gymnasium in 8 Jahren, 105
- ganzheitlich, 47
- Gedächtnis, 39
- Geisteswissenschaften, 28
- Pädagogik, 32
- Pädagogik, 128
- gemeinsame Nutzung von Ressourcen, 24
- genetisches Prinzip, 41
- Gesamtschule, 117, 144
- Gesamtunterricht, 32
- geschlechtsspezifische Zugangsweisen, 155
- geschlossene Fragen – Expertinneninterview, 86
- geschützte Materialien, 106
- gesellschaftliche Anforderungen, 3
- gesellschaftliches Steuerungsinteresse, 31
- Gesetzgeber, 106
- GI
  - Fachgruppe für Informatiklehrerinnen, 85
  - Jahrestagung, 85
- GOF – Gang of Four, 20
- Grammatik, 12
  - deterministisch – kontextfrei, 208
  - regulär, 206
- graphische Benutzungsoberfläche – GUI, 109, 210
- Grundbildung
  - informationstechnische (ITG), 68, 79
  - mathematische, 52
  - naturwissenschaftliche, 52
- Grundformen
  - des Unterrichts, 32
- Grundkursanforderungen lt. KMK, 62
- Gruppenarbeit, 53, 124
- Gruppeninterview, 116, 120
- Gruppeninterview, 179–186*
- Gruppenlernen, 40
- gruppenorientierte Arbeitsformen, 117
- GUI – Graphical User Interface, 109, 210
- gymnasiale Oberstufe, 34, 62, 69, 77, 115
- Gymnasiallehrer, 50
- Gymnasiasten, 50
- Gymnasium, 77

- Mittelstufe, 141  
 Gütekriterien, 51
- Halteproblem, 208  
 Haltungen, 26  
 Hamburger [Didaktik-]Modell, 35  
 Handgriff zur Wirklichkeit, 13  
 Handlungsorientierung, 47  
 Handlungsroutinen, 127  
 Handlungsrückgrat, 104, 111, 125, 135, 140, 154  
 Hauptschullehrplan NRW, 136  
 Hausaufgabe, 68  
 HCI – Human Computer Interaction, 140  
 Hierarchiemodell  
   Wolfgang Schulz, 36  
 Hierarchiestufen, 35  
 High school curriculum – ACM 1993/1997, 73  
 HLL – High Level Language, 21  
 Hochschuldidaktik, 153  
 Hochschulinformatik, 154  
*Hospitationsprotokolle zur Evaluation, 187–196*  
 HTML – Hypertext Markup Language, 140, 210  
 Hybridisierung, 14, 20  
 Hybridobjekte, 13
- IBU – Informatik Betrifft Uns (Zeitschrift – Erscheinen eingestellt), 164  
 ICF – Informatics Curriculum Framework, 76  
 ICT – Information and Communication Technology, 76  
 IDE – Integrated Development Environment, 141  
 ikonisch, 38, 112  
   -e Aneignungsweise, 47  
 imperative Konstrukte, 11  
 Implementierung, 21  
 Indikatoren, 52  
   quantitative, 52
- Informatik  
   a formal and an engineering science, 18  
   Architektur, 18  
   Computer Literacy, 74  
   Computerwissenschaft, 118  
   Definition – 1957, 9  
   Definition – IFIP/UNESCO, 74  
   Einschätzung durch Lernende, 115  
   enge Kopplung an reale Arbeitsprozesse, 18  
   Fachkonferenz, 111  
   Gegenstände, 10  
   Geistes-Ingenieurwissenschaft, 17  
   Grundideen, 58  
   Grundkonzepte, 58  
   hermeneutische Disziplin, 18  
   Ingenieur-Geisteswissenschaft, 17  
   Ingenieurwissenschaft, 16  
   Inhalte sind maschinenunabhängig, 17  
   konstruktive Wissenschaft, 56  
   Konzepte, 20  
   Lehre von der Bedienung von Computern, 118  
   Methoden, 13  
   Methodenkritik, 57  
   Neubewertung der Fachstruktur, 58  
   Objektbereich, 16  
   Paradigmenwechsel, 5  
   pragmatische Charakterisierung, 13  
   praxisorientierte und abstrakte Wissenschaft, 17  
   problem solving skills, 74  
   Schulfach, 3  
   technische, 64  
   theoretische, 107, 135  
   Unterrichtsfach, 3, 127  
   Wissenschaft  
     praxisorientiert und abstrakt, 17  
   Wissenschaft für das Internet, 118  
   Wissenschaft vom streng geregelten Informationsfluss, 17  
   Wissenschaft zur Entwicklung künstlicher Intelligenzen, 118  
   Wissenschaft zur Rationalisierung der geistigen Arbeit, 118  
   Wissenschaftstheorie, 9  
   Zeitleiste zu Innensichten, 19
- Informatik als Gegenstand und Mittel des Lernens, 137–147**
- Informatikbildungsdokument  
 GI-Empfehlung 1976, 55, 57  
 GI-Empfehlung 1993, 65  
 GI-Gesamtkonzept 2000, 69
- Informatiklabor, 4  
 Informatiklehrerin, 6, 83, 85  
 Informatikprodukte, 59, 63  
 Informatikprojekt, 46  
 Informatikraum, 4, 130  
 Informatiksystem, 2, 3, 5, 11, 41, 64  
   Architektur, 65  
   Bedienung/Benutzung, 59  
   BSCW, 59, 68  
   Dekonstruktion, 69  
   Eigenschaften  
     interaktive, 5  
   Entmystifizierung, 55  
   Interaktion, 69  
   Komplexität, 108  
   Konstruktion, 69  
   kooperativ, 58  
   menschengerechte Gestaltung, 154  
   methodischer Entwurf – Problemlösen, 64

- praktischer Umgang, 118, 127
- Quelle, 9
- soziotechnischer Kontext, 65
- Struktur, Entwurf, 65
- verstehen und verantwortlich nutzen, 134
- Wechselwirkungen, 69
- Wirkprinzipien, 57, 65, 69
- Informatikunterricht**
  - Algorithmen und Datenstrukturen, 63
  - besondere Zieldimension, 81, 83
  - Bildungsanforderungen, 127
  - Eignung von Programmiersprachen, 142
  - evaluieren, 127
  - funktionale Modellierung, 84
  - Konzepte, 84, 92
  - konzeptioneller Rahmen, 58
  - Lehrexpertise, 83
  - Medien, 130
  - Modellierung, 84
  - Modellierung als Prinzip, 67
  - objektorientierte Modellierung, 84
  - Projektphase, 55
  - prozedurale Programmiersprachen, 63
  - Systemorientierter Ansatz, 64
  - Telekommunikation, 67
  - vernetzte Systeme, 67
  - Werkzeuge, 84
  - wissensbasierte Modellierung, 84
- Informatikunterricht evaluieren, 127–136**
- Information, 108**
  - automatische Verarbeitung, 134
  - Darstellung mit Hilfe von Dokumenten, 134
  - in vernetzten Umgebungen, 134
  - Konzepte, 65
- Informationsgesellschaft, 5**
- Informationstechnische Grundbildung (ITG), 68, 79**
- Informationsverarbeitung**
  - prinzipielle Grenzen, 65
- informationszentrierter Ansatz, 67**
- Informatique, 9**
- Informatische Bildung, 1, 5, 55, 59, 67, 72, 78, 104, 118, 119, 141**
  - anwendungsorientiert, 59
  - Modulkonzept, 7, 103
  - Neuorientierung, 63
- informatische Kompetenzen, 3**
- informatische Modellierung, 92**
  - prinzipielle Grenzen, 108
- informatischer Problemlösungsprozess, 56**
- informatisches Handeln, 13**
- informatisches Modellieren**
  - objektorientiert, 211
- informatisches Wissenschaftsverständnis, 5**
- Informatisierung, 13**
- INFOS – Informatik und Schule, 63, 85**
- Ingenieurwesen, 44**
- inhaltliches Benchmarking, 52**
- Innenstruktur Lerngegenstand, 62**
- innere Reform des Unterrichts, 54**
- institutionell-politische Vorgaben, 54**
- Instruktion, 35, 44, 51**
  - Annahmen, 35
  - Probleme, 36
- instrumentelles Handeln, 67**
- Integration, 31**
- intelligentes Verhalten, 39**
- Interaktion, 40, 109**
- Interaktionsmerkmale, 53**
- interaktive Eigenschaften von Informatiksystemen, 5**
- Internetdienste, 88**
- Interview, 85**
  - Auswahl der Expertinnen, 85
  - Auswahlkriterien für Expertinnen, 85
  - Forschungsmethode, 84
  - geschlossene Fragen, 86
  - Konkretisierung der Fragen, 86
  - Leitfaden, 86
  - offene Fragen, 86
  - Phasierung, 85
  - Untersuchungsgestaltung, 83
  - Ziele, 83
- Intranet, 144, 210**
- Intranet in der Schule, 210**
- ITG – informationstechnische Grundbildung, 68, 79**
- Jackson-Methode, 107**
- Java, 142**
- JSP – Jackson Structured Programming, 107**
- kalvinistische Tradition, 128**
- Kellerautomaten, 198, 208**
- Kenntnisse**
  - fachdidaktische, 4
- Kernideen, 48**
- Kerninformatik, 64**
- KI – Künstliche Intelligenz, 38**
- Klammerstrukturen, 198, 210**
- Klassenarbeit, 68**
- Klassenstrukturen, 23**
- Klassifizierung von Ausdrücken, 24**
- Kognition, 39**
- Kognitionstheorie, 40**
- kognitive Aktivierung, 53**
- kognitive Psychologie, 37**
- Kognitivismus, 37, 153**

- Kommunikation, 33, 52, 109  
 in vernetzten Umgebungen, 134  
 Konzepte, 65  
 Kommunikationsprozesse  
 Unterstützung durch Informatiksysteme, 105  
 Kommunikationsunterstützung, 124  
 kommunikatives Handeln, 67  
 Kompetenz  
 fächerübergreifend, 52  
 informatische, 3  
 Problemlösen, 75  
 komplexe Planungsprozesse, 46  
 komplexes Ausgangsproblem, 53  
 Komplexität, 57, 108  
 Komplexität von Problemlösungen, 60  
 Konditionieren  
 operantes, 37  
 Konfiguration, 20  
 Königstein, 85  
 Konstruktion  
 objektorientiert, 5  
 Konstruktivismus, 37, 39  
 pragmatischer, 153  
 Konstruktivismus im Vergleich, 40  
 kontinuierliches Weiterlernen, 52  
 Konzept  
 bildungstheoretisch, 52  
 fachdidaktisch, 130  
 Kooperation, 52  
 Kriterienkatalog, 137  
 kritisch-konstruktive Didaktik, 31  
 Kritische Erziehungswissenschaft, 128  
 Kritischer Rationalismus, 37  
 4. Kulturtechnik, 56  
 vierte Kulturtechnik, 56  
 Kulturwerkzeuge  
 basale, 52  
 Kybernetik, 11, 16  
 Informationsbegriff, 17  
 Kybernetische Pädagogik, 37  
 Längsschnitt, 121  
 Laptopklassen, 140  
 Lebenschancen, 31  
 Lebensweltbezug, 154  
 Legitimation, 31  
 Lehrerbildung, 32  
 Lehrerfortbildung Informatik, 88  
**Lehrexpertise zum Informatikunterricht, 83–101**  
 Lehrmaschinen, 37  
 Lehrplan, 111  
 heimlicher, 81  
 Lehrpläne, 4  
 Lehrplanentwicklung, 88  
 Lehrprogramme, 37  
 lehrtheoretische Didaktik, 35  
 Lehrtheorie, 32  
 Leistungsanforderungen, 54  
 Leitkonzepte, 68  
 LEO – Lernumgebung für objektorientiertes Modellieren im Informatikunterricht, 143  
 Lernabsicht, 42  
 Lernen  
 entdeckendes, 47  
 fächerverbindendes, 34  
 handlungsorientiert, 47  
 selbstreguliert, 52  
 vorwegnehmend, 38  
 Lernfortschritt, 35  
 lernförderliche Szenarios, 4  
 Lernhilfen, 39  
 Lernobjekte, 33, 137  
 Lernplattformen, 33  
 Lernprozess  
 erfolgreich, 54  
 technisch unterstützt, 33  
 Lernschritt, 37  
 lerntheoretische Didaktik, 35, 37  
**Lerntheoretische Grundlagen, 31–54**  
 lernzielorientierte Didaktik, 37  
 Lesekompetenz, 52  
 Leseverständnis, 52  
 Linguistik, 209  
 LISP – List Processing, 12  
 Literacy  
 Computer, 52  
 Mathematical, 52  
 Reading, 52  
 Scientific, 52  
 LOG IN – Fachzeitschrift für Informatikunterricht, 63  
 LOM – Learning Object Metadata, 33  
 Lösungsmuster  
 wiederkehrende, 20  
 Lösungsstrategie, 43  
 lycée, 77  
 Mailprogramm, 210  
 Manipulation von Objekten, 21  
 Mathematical Literacy, 52  
 Mathematik  
 Fachdidaktik, 3  
 Mathematiklastigkeit, 93  
 mathematische Grundbildung, 52  
 Medienkompetenz, 19  
 Mensch-Maschine Analogie, 11  
 Merkmalsausprägungen, 53

- Methoden des Unterrichts, 32  
Mittelbehörde, 88  
mittlere Direktivität, 53  
Modell  
    Berliner, 37  
    informatisches, 13  
Modellbegriff, 211  
Modellbildungsstrategien, 107  
Modellierung, 75  
    informatisch, 13, 107  
    objektorientiert, 69, 70, 101, 109, 129  
    prädikativ, 116  
    übergreifendes Konzept, 92  
Modellierungsaufgabe  
    Familienbeziehungen, 116  
    geschlossen, 116  
    Labyrinth, 116  
    offen, 116  
Modellierungsphase, 125  
    Analyse, 125  
    Implementierung, 125  
    Problemlösung, 125  
    Test, 125  
    Verschrottung, 125  
Modula-2, 12, 204  
Modula-3, 21  
modulares Konzept, 7  
Modulkonzept, 1, 7, 105, 127  
    Beispielübersicht, 110  
    kritische Analyse, 134  
**Modulkonzept, 103–114**  
MUE – Multimediale Evaluation in der Informatiklehrerbildung, 130  
MuesliAutomat, Automatentafel, 200  
MuesliriegelAutomat, verbale Beschreibung, 202  
MVC – Model View Control, 107, 139
- nationale Bildungssysteme, 52  
naturwissenschaftliche Grundbildung, 52  
Nebenläufigkeit  
    Server-Klient Modellierung, 220  
Nebenläufigkeit, 109, 220  
Netiquette, 106, 134, 210, 211  
Netzwerkfähigkeit, 140  
Niederlande, 128  
Normen, 26, 42, 51, 139, 211
- Oberon, 4, 141  
Oberstufe, gymnasiale, 115  
Oberstufenreform, 3  
Object-Pascal, 21  
Objekte, 24  
objektorientierte Konstruktion, 5  
objektorientierte Modellierung, 69, 70, 101, 109, 129  
objektorientierte Sicht, 22  
Objektorientierung, 23  
**Offene Fragen, 153–155**  
offene Fragen – Expertinneninterview, 86  
OOA – Object Oriented Analysis, 27  
OOM – Objektorientierte Modellierung  
    Datenstruktur Keller als Klasse, 217  
    Klassenstruktur, 213  
    Kontrollstrukturen, 214  
    Schülercomputerarbeitsplatz, 213  
OOM – objektorientierte Modellierung, 109, 110, 211  
operantes Konditionieren, 37  
operationale Form, 20  
Optimalklassen, 53  
Ordnerstrukturen, 210  
Organisation und Gestaltung von Arbeitsplätzen, 18  
orthogonale  
    Grundkonzepte, 24  
    Konzepte, 24
- Pädagogik, 31  
    geisteswissenschaftliche, 32, 128  
    kybernetische, 37  
    relativistische, 48  
    systematisch, 32  
Paderborn, 130  
PAP – Programmablaufplan, 28, 84, 163  
Paradigma, 5, 21, 22, 24  
    Definition, 5  
    Programmier-, 21  
    traditionell, 18  
    Turing Maschine, 14  
Paradigmen des Lehrens und Lernens, 32  
Paradigmenwechsel, 5, 9, 19, 40  
partizipative Konzepte, 28  
partizipative Softwareentwicklung, 4  
Pascal, 4, 142  
Passwortschnittstelle, 206  
PDA – Personal Digital Assistant, 140  
PDA – Push Down Automata, 198, 208  
PDF – Portable Document Format, 140  
Perspektivenwechsel, 66  
Perturbation, 40  
Pflichtschulzeit, 52  
PG – Projektgruppe, 47, 146, 153  
Phase, 33  
Phasenabfolge, 46  
Phasenmodell, 46  
Phasenschema, 32  
    Kritik, 32, 33

- Problemorientierung, 42
- Unterrichtsplanung, 33
- Phasenunterteilung, 32
- Philosophie, 108
- Phlogiston, 11
- PISA – Programme for International Student Assessment, 49, 52
- Plankalkül, 11
- Planungskompetenz, 52
- Planungsprozess
  - komplexer, 46
- plattformunabhängig, 140
- Plattformunabhängigkeit, 106
- Portierung, 139
- Prädikatenkalkül, 21
- Prädikatenlogik, 11
- prädikative Modellierung, 116
- Pragmatik, 108, 153
- Praktika, 153
- Präsentation, 33
- Primat der Instruktion, 35
- Prinzip
  - Fortsetzbarkeit, 38
  - genetisches, 41
- prinzipielle Grenzen, 108
- Prinzipien
  - didaktische, 32
- Problem
  - didaktische Charakterisierung, 42
- Problemgewinnung, 125
- Problemlösefähigkeit
  - Planungskompetenz, 52
- Problemlösen, 43, 52
  - als Prozess, 43
  - Schema zum, 43
- Problemlösestrategien, 43, 48
- Problemlösen, 79, 83
  - Kompetenz, 75
- Problemorientierung, 105
  - Unterricht, 42
- Problemsituation, 42
- Produkt, 68
- Produktorientierung, 47
- Professionalisierung, 31, 44
- professionelle Arbeitsweise, 46
- Programmablaufplan, 84
- Programmablaufpläne, 96
- Programme starten und beenden, 210
- Programmierparadigmen, 21
- Programmiersprache
  - Java, 142
  - Oberon, 141
  - Pascal, 142
  - Python, 142, 144
- Programmiersprachen
  - Abstraktionsmechanismen, 22
- programmierter Unterricht, 37
- Programmierung, 4
- Projekt, 68
- Projektbegriff, 43
  - informatisch, 80
  - pädagogisch, 44, 80
- Projektgedanke, 43
- Projektgruppe, 47, 142, 146, 153
- Projektmethode, 43
  - Geschichte, 44
- projektorientierte Unterrichtsformen, 54, 117
- projektorientierter Unterricht, 43, 55, 63, 67–69, 73, 78–80
- Projektphasen, 4, 44
- Projektunterricht, 44, 79
  - Merkmale, 45
- Projektwoche, 141
- Prolog, 4
- prozentuale Ergebnisse, 87
- Prozessabstraktion, 20
- PS – PostScript, 140
- Pseudocode, 12
- Psychologie
  - kognitive, 37
- Pubertät, 32
- Pulldownmenü, 206
- PyLZK, 144
- PyNassi, 144
- PyNetzwerkmonitor, 145
- PySuM, 144
- Python, 142, 144
- Qualifikation, 31
  - berufliche, 154
- qualitative Studien, 83
- Qualitätskriterien, 3
- quantitative Aussagen, 88
- quantitative Indikatoren, 52
- quelloffene Software, 140
- Rahmenkonzepte – BLK, 62
- Rationalismus
  - kritischer, 37
- Reading Literacy, 52
- Rechenschaft, 128
- Reflexionsfähigkeit, 116
- Reformpädagogik, 32, 33
- Rekontextualisierung, 13
- relativistische Pädagogik, 48
- Repräsentationsmodell, 38
- Repräsentationsstufen, 38
- repräsentative Studie, 83

- Repräsentation von Texten, 210  
 RFC – Request for comment, 111  
 RGU – Rechner-Gestützter Unterricht, 142  
 Rollenspiel, 135  
 Routing, 135  
 RUP – Rational Unified Process, 107  
 RvS – Rechnernetze und verteilte Systeme, 95,  
 105, 106, 109, 110, 113, 117, 220  
 Rückkopplungsprozess, 7  
  
 SADT – Structured Analysis Design Technique,  
 107  
 Scanner, 208  
 Schema, 43, 84  
     Problemlösung, 43  
*Schemata zum Problemlösen*, 225–226  
 Schematheorie, 40  
 Schichtenmodell, 220  
 Schlüsselbegriffe, 68  
 Schnittstelle AutomatiO, 199  
 Schnittstellen  
     dokumentierte, 140  
 Schnittstellenspezifikation, 24  
 Schreibtischmetapher, 210  
     erweitert (CSCW), 211  
 schriftliche Befragung, 116  
 schriftliches Zentralabitur, 4  
 schrittweise Verfeinerung, 22, 56  
 Schule  
     Funktionen, 31  
     Institution, 31  
 Schulen ans Netz, 88  
 Schüleraktivität, 130  
 Schülerfragen, 53  
 Schülergruppeninterview, 120  
 Schülerorientierung, 53, 54  
 Schulfach Informatik, 3  
 Schulfächer, 66, 75, 77  
 Schulgemeinde, 106  
 Schulintranet, 140  
 schulpraktische Studien, 116  
 Schulserver, 140  
 Schulstoff, 35  
 Schulträger, 106  
 Schulversuch  
     Pflichtfach Informatik, 128  
 schutzwürdige Belange, 106  
 Scientific Literacy, 52  
 SCORM – Sharable Content Object Reference  
 Model, 33  
 SDK – Software Development Kit, 146  
 SEC III – Open IFIP-GI-Conference on Social,  
 Ethical and Cognitive Issues of Infor-  
 matics and ICT, 143  
  
 Sekundarstufe I, 127, 141  
     vergleichbare Abschlussbedingungen, 62  
 Sekundarstufe II, 62, 127, 141  
     Anfangsunterricht, 105  
 selbstreguliertes Lernen, 52  
 Selbstständigkeit, 47  
 Selektion, 31  
 Semantik, 108, 153  
 semantische Lücke, 22  
 Semiotik, 16  
 Setting, 33  
 SGML – Standard Generalized Markup Langua-  
 ge, 210  
 Sicht  
     Algorithmische  
         Struktogramm, 57  
         aufgabenorientiert, 22  
         objektorientiert, 22  
 Sichtweisen, 28  
 Simula, 23  
 Simulation, 109  
 Skriptsprachen, 106  
 Smalltalk, 23  
 Softwareentwicklung  
     partizipativ, 4, 64  
 Softwareergonomie, 140  
 Softwarekrise, 59  
 Softwareprojekt, 80  
 Softwaretechnik, Basiskonzepte, 59  
 sokratisch, 41  
 Sozialchancen, 31  
 Sozialisation, 31  
 soziotechnische Systeme, 4, 28, 64  
 Spezifikation, 13  
 spielerisch, 47  
 spielerische Interaktion als dritte Modalität des  
     Handelns, 67  
 Spielzeug, 67, 78  
 spiralförmiges Curriculum, 38, 105  
 spiraliger Aufbau, 62, 105  
 Spiralprinzip, 38  
 Sprache  
     applikativ, 21  
     funktional, 21  
     imperativ, 21  
     logisch, 21  
     objektorientiert, 21  
     prozedural, 21  
     relational, 21  
 Sprachklasse, 21, 22, 208  
 Sprachübersetzung, 108  
 Sprites, 144  
 Sprungbefehl, 11  
**Stand der Schulinformatik, 55–82**

- Standardbrowser, 210
- Stoffverteilungsplan, 111
  - Informatikgrundkurs, 198
- Stoffverteilungsplan 11.1 - Automaten, 198
- Struktogramm, 57, 201
  - MuesliAutomat
    - Ausgabefunktion, 201
    - Zustandsübergänge, 201
- Struktur
  - en, operationale, 20
- Strukturieren von Dokumentennetzen durch Verweise, 211
- strukturierte Programmierung, 22, 56, 57
- Strukturwissenschaften, 16
- Studie
  - repräsentativ, 83
- Studien
  - qualitative, 83
  - schulpraktisch, 116
- Stufenschema
  - Problemorientierung (didaktisch), 42
- subjektorientiertes Verständnis, 41
- Suchmaschine, 108
- SVG – Scalable Vector Graphics, 140
- symbolisch, 38, 112
- Symbolische Maschine, 20
- Syntax, 108, 153
- systematische Pädagogik, 32
- Systembetreuung, 106
- Systeme
  - soziotechnische, 4
  - vernetzte, 5
  - verteilte, 5
- Systemisieren, 13
- Szenarios
  - lernförderliche, 4
- Taxonomien des Lernens, 37
- TCP – Transmission control protocol, 110
- Team, 68
- Technikgeneseforschung, 11
- technische Informatik, 64
- Texteditor, 210
- Thread, 220
- Thüringen, 4
- Tierexperiment, 37
- TIMSS – Third International Mathematics and Science Study, 49, 52
- Top-Down, 22
  - Analyse, 43
- tradierte Wissenschaften, 29, 104
- träges Wissen, 37
- Transduktor, 198, 204
- Transfer
  - unreflektiert, 4
- Transkriptausschnitte, 88
- Trivialisierung, 53
- Turing-Test, 208
- Typen, 24
  - Erweiterung, 25
- Typen operationaler Strukturen, 20
- UDP – User Datagram Protocol, 220
- UML – Unified Modelling Language, 84, 142, 163, 226
- Unterricht
  - anregungsarm, 53
  - anspruchsvoll, 53
  - direktiver, 53
  - enaktiv, 47
  - fächerverknüpfend, 57
  - fragend-entwickelnder, 53
  - ganzheitlich, 47
  - handlungsorientiert, 47
  - innere Reform, 54
  - Konzepte, 41
  - Methoden, 32
  - Phasenschemata, 32
  - Phasierung, 32
  - programmierter, 37
  - projektorientiert, 43
  - schülerorientiert, 32
  - spielerisch, 47
- Unterrichtsalltag, 32
- Unterrichtsbeobachtung, 52
- Unterrichtsbeobachtungsinstrumente, 52
- Unterrichtsfach Informatik, 3, 127
- Unterrichtsformen
  - projektorientierte, 54
- Unterrichtsgestaltung
  - Konzepte, 86
- Unterrichtskonzepte, 41
- Unterrichtsmittel, 130
- Unterrichtsmuster, 52, 53
- Unterrichtsphilosophie, 32
- Unterrichtsprüfung, 53
- Unterrichtssequenz, 199
  - Gestaltung, 127
- Unterrichtsskript, 53
- Unterrichtsstruktur, 32
- Untersuchungsgestaltung
  - Bild der Informatik bei Schülerinnen, 115
- URL – Uniform Resource Locator, 162, 211
- Use-Case, 23
- Variablenbegriff der wissensbasierten Modellierung, 108
- Veranstaltungsform, 153

- Vergleichsnormierung, 52
- vernetzte Systeme, 5
- verpflichtendes Curriculum, 73
- verteilte Systeme, 5
- Videodokumentation, 130
- Videomitschnitte, 52
- Visualisierung, 84
- von-Neumann-Sprache, 12
- Vorgehen
  - evolutionär, 20
- Vorgehensmodell, 137, 146
- vorwegnehmendes Lernen, 38
  
- Wahrnehmungspsychologie, 140
- Wasserfallmodell
  - Aufkündigung, 59
- Webangebot Schulinformatik, 88
- Weltausschnitt, 29
- Werte, 26
- White Box, 108
- Lösungsmuster, 20
- Wiener Kreis, 37
- Wirkprinzipien, 57, 65, 69, 107
- Wissen, 108
- wissensbasierte Modellierung, 108
- Wissenschaft
  - Ingenieurbezug, 59
  - technische Artefakte, 59
- wissenschaftliche Arbeitsweise, 29
- Wissenschaftspropädeutik, 34, 62
- Wissenschaftstheorie, 7, 37, 42, 60, 63, 66
- Wissenschaftstheorie und Informatik, 9–29**
- Wissensgesellschaft, 5
- Wissensnetz, 154
  
- XMI – XML Metadata Interchange, 141
- XML – eXtensible Markup Language, 140, 144, 146, 211
  
- Y2K – Year 2 Kilo (Jahr 2000), 12
  
- Z1, 12
- Z3, 12
- Zusammenfassung, 149–152**
- Zusammenhang zwischen Automaten und formalen Sprachen, 206
- Zustandsveränderungen, 40

# Abkürzungsverzeichnis

4GL	4th Generation Language, Seite 21
ACM	Association for Computing Machinery, Seite 6
ADT	Abstrakter Datentyp, Seite 209
AEDS	Association for Educational Data Systems, Seite 6
ALGOL	Algorithmic Language, Seite 12
API	Application Program(ming) Interface, Seite 140
ATEE	Association for Teacher Education in Europe, Seite 6
BLK	Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung, Seite 62
BMBF	Bundesministerium für Bildung und Forschung, Seite 151
BMBF	Bundesministerium für Bildung, Wissenschaft Forschung und Technologie, Seite 49
BNF	Backus Naur Form, Seite 12
BSCW	Basic System Cooperative Workspace, Seite 59
C & U	Computer & Unterricht, Seite 164
CACM	Communications of the ACM, Seite 229
CASE	Computer Aided Software Engineering, Seite 163
CASE	Computer Aided Software Engineering, Seite IX
CCC	Cross-Curricular Competencies, Seite 52
CLOS	Common LISP Object System, Seite 21
CRC	Class-Responsibility-Collaboration, Seite 23
CSCL	Computer supported cooperative learning, Seite 106
CSCW	Computer supported collaborative work, Seite 106
CUU	Computerunterstützter Unterricht, Seite 63
DFG	Deutsche Forschungsgemeinschaft, Seite 15
DIN	Deutsches Institut für Normung, Seite 201
DTD	Document Type Definition, Seite 144
DVA	Datenverarbeitungsanlage, Seite 80
ECDL	European Computer Driving Licence, Seite 76
EDK	Eidgenössische Erziehungsdirektorenkonferenz, Seite 74
EDSAC	Electronic Delay Storage Automatic Calculator, Seite 12
EDVAC	Electronic Discrete Variable Calculator, Seite 11
ENIAC	Electronic Numerical Integrator and Calculator, Seite 12
EPA	Einheitliche Prüfungsanforderungen in der Abiturprüfung, Seite 136
ER	Entity-Relationship, Seite 163
ERM	Entity Relationship Model, Seite 107
EVA	Eingabe Verarbeitung Ausgabe, Seite 198
FBI	Fachbereich Informatik, Seite 142
FhG	Fraunhofer Gesellschaft, Seite 68
G8	Gymnasium in 8 Jahren, Seite 105
GI	Gesellschaft für Informatik e. V., Seite 6
GMD	Gesellschaft für Mathematik und Datenverarbeitung mbH – 2001 vereinigt mit der FhG, Seite 68
GOF	Gang of Four, Seite 20

GUI .....	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface, Seite 210
HCI .....	<u>h</u> uman <u>C</u> omputer <u>I</u> nteraction, Seite 140
HLL .....	<u>H</u> igh <u>L</u> evel <u>L</u> anguage, Seite 21
HTML .....	<u>H</u> ypertext <u>M</u> arkup <u>L</u> anguage, Seite 140
IBM .....	<u>I</u> nternational <u>B</u> usiness <u>M</u> achines Corporation, Seite 46
IBU .....	<u>I</u> nformatik <u>B</u> etrifft <u>U</u> ns, Seite 164
ICF .....	<u>I</u> nformatics <u>C</u> urriculum <u>F</u> ramework, Seite 76
ICT .....	<u>I</u> nformation and <u>C</u> ommunication <u>T</u> echnology, Seite 76
IDE .....	<u>I</u> ntegrated <u>D</u> evelopment <u>E</u> nvironment, Seite 141
IEA .....	<u>I</u> nternational Association for the Evaluation of <u>E</u> ducational <u>A</u> chievement, Seite 49
IFIP .....	<u>I</u> nternational <u>F</u> ederation for <u>I</u> nformation <u>P</u> rocessing, Seite 6
INFOS .....	<u>I</u> nformatik und <u>S</u> chule, Seite 63
ITG .....	<u>I</u> nformationstechnische <u>G</u> rundbildung, Seite 68
JSP .....	<u>J</u> ackson <u>S</u> tructured <u>P</u> rogramming, Seite 107
KI .....	<u>K</u> ünstliche <u>I</u> ntelligenz, Seite 38
KM .....	<u>K</u> ultusministerium, Seite 164
KMK .....	<u>K</u> ultusministerkonferenz, Seite 62
LEO .....	<u>L</u> ernumgebung für objektorientiertes Modellieren im Informatikunterricht, Seite 143
LISP .....	<u>L</u> ISt <u>P</u> rocessing, Seite 12
LOM .....	<u>L</u> earning <u>O</u> bject <u>M</u> etaddata, Seite 33
MSWF .....	<u>M</u> inisterium für <u>S</u> chule, <u>W</u> issenschaft und <u>F</u> orschung, Seite 130
MUE .....	<u>M</u> ultimediale <u>E</u> valuation in der Informatiklehrerbildung, Seite 130
MVC .....	<u>M</u> odel <u>V</u> iew <u>C</u> ontrol, Seite 107
OECD .....	<u>O</u> rganisation for <u>E</u> conomic <u>C</u> o-operation and <u>D</u> evelopment, Seite 49
OOA .....	<u>O</u> bject <u>O</u> riented <u>A</u> nalysis, Seite 27
OOM .....	objektorientierte <u>M</u> odellierung, Seite 109
PAP .....	<u>P</u> rogrammablaufplan, Seite 28
PARC .....	<u>P</u> alo <u>A</u> lto <u>R</u> esearch <u>C</u> enter, Seite 23
PDA .....	<u>P</u> ush <u>D</u> own <u>A</u> utomata, Seite 198
PDA .....	<u>P</u> ersonal <u>D</u> igital <u>A</u> ssistant, Seite 140
PDF .....	<u>P</u> ortable <u>D</u> ocument <u>F</u> ormats, Seite 140
PG .....	<u>P</u> rojektgruppe, Seite 47
PG .....	<u>P</u> rojektgruppe, Seite 146
PISA .....	<u>P</u> rogramme for <u>I</u> nternational <u>S</u> tudent <u>A</u> ssessment, Seite 49
PS .....	<u>P</u> ostScript, Seite 140
RFC .....	<u>R</u> equest for comment, Seite 111
RGU .....	<u>R</u> echner-Gestützter <u>U</u> nterricht, Seite 142
RUP .....	<u>R</u> ational <u>U</u> nified <u>P</u> rocess, Seite 107
RvS .....	<u>R</u> echnernetze und <u>v</u> erteilte <u>S</u> ysteme, Seite 95
SADT .....	<u>S</u> tructured <u>A</u> nalysis <u>D</u> esign <u>T</u> echnique, Seite 107
SCORM .....	<u>S</u> harable <u>C</u> ontent <u>O</u> bject <u>R</u> eference <u>M</u> odel, Seite 33
SDK .....	<u>S</u> oftware <u>D</u> evelopment <u>K</u> it, Seite 146
SEC III .....	Open IFIP-GI-Conference on <u>S</u> ocial, <u>E</u> thical and <u>C</u> ognitive <u>I</u> ssues of <u>I</u> nformatics and <u>I</u> CT, Seite 143
SVG .....	<u>S</u> calable <u>V</u> ector <u>G</u> raphics, Seite 140
TCP .....	<u>T</u> ransmission control protocol, Seite 110
TIMSS .....	<u>T</u> hird <u>I</u> nternational <u>M</u> athematics and <u>S</u> cience <u>S</u> tudy, Seite 49
UDP .....	<u>U</u> ser <u>D</u> atagram <u>P</u> rotocol, Seite 220
UML .....	<u>U</u> nified <u>M</u> odelling <u>L</u> anguage, Seite 84
URL .....	<u>U</u> niform <u>R</u> esource <u>L</u> ocator, Seite 162
W3C .....	<u>W</u> orld <u>W</u> ide <u>W</u> eb <u>C</u> onsortium, Seite 140
XMI .....	<u>X</u> ML <u>M</u> etaddata <u>I</u> nterchange, Seite 141
XML .....	e <u>X</u> tensible <u>M</u> arkup <u>L</u> anguage, Seite 140
Y2K .....	<u>Y</u> ear <u>2</u> <u>K</u> ilo, Seite 12

Es sind nicht die Dinge selbst, sondern die  
Meinungen über die Dinge, welche die  
Menschen verwirren.

---

[Epiktet 120]