

Rezension

Schubert, S., Schwill, A.: Didaktik der Informatik. Spektrum Akademischer Verlag, Berlin, 2004.

1 Zielgruppe und Aufbau

Das Lehrbuch richtet sich vornehmlich an Informatiklehrer der Sekundarstufe II. Inhaltsverzeichnis und Vorwort findet man auf der Webseite des Buches [Sch03]. Der Text geht auf [Sch93, Sch98, Sch99] zurück.

2 Positionierung

In der Diskussion um den Beitrag der Schulinformatik zur Allgemeinbildung scheinen Schubert/Schwill

- die Bedeutung metakognitiver Strategien der Problemlösung verbunden mit
- Techniken der Wissensrepräsentationen (ikonische, symbolische und enaktive Modelle)

zu betonen (S. 108). Der Bildungswert der Informatik liege in der

- Förderung von Metawissen zur Beherrschung von Komplexität und der
- Bedeutung der informatischen Strukturierung als Erkenntnismethode (auch unabhängig von Informatiksystemen),

was an Veränderungen in verschiedenen anderen Wissenschaften unter dem Einfluss der Informatik zu beobachten sei (S. 134).

Nichts liege nun näher, als dass die Informatikdidaktik Wert und Nutzen der Methoden ihres eigenen Lehrgegenstandes für die Unterstützung der Lehre deutlich mache. Schubert/Schwill nennen eine entsprechende Kollektion aufeinander abgestimmter Lehr-Lern-Materialien „didaktisches System“. Als Komponenten eines solchen Systems sehen sie:

- eine Baumstruktur von Aufgabenklassen, die jeder Lerneinheit klassifizierte Aufgaben zuordnen,
- Explorationsmodule (Lernsoftware) für entdeckendes Lernen (siehe z.B. [IML, Kara, Karol, LEO, life3, MOOP, OLLI, SIMBA, Ste99]),
- Graphen als Beschreibungsstrukturen für das Fachwissen / Lerneinheiten und
- Entwurfsmuster für Lernprozesse, die gegenüber herkömmlichen Erfahrungsberichten Vergleichbarkeit und Qualitätskriterien für den Anwendungserfolg sicherstellen sollen.

Eine derartige Repräsentation fachdidaktischen Wissens soll die fachdidaktische Kommunikation fördern und die Umsetzung im Unterricht inhaltlich und methodisch erleichtern. Sie diene nicht zuletzt der Operationalisierung von Bildungsstandards (S. 134).

3 Begutachtung der Inhalte

Die Autoren machen im Vorwort ihren eigenen Anspruch deutlich und verweisen auf in den letzten Jahren entstandenen Konzepte der Informatikdidaktik, auf die sich die Schulinformatik gründen kann: die fundamentalen Ideen, den informationszentrierten Zugang, didaktische Systeme sowie die Dekonstruktion.

Leitlinien der Auswahl und Klassifikation von Unterrichtsinhalten dienen dazu, Unterrichtsinhalte zu vernetzen und den jeweiligen Beitrag zu den Bildungszielen transparent zu machen. Schubert/Schwill schlagen in Kapitel 2 die Leitlinien Pläne, Sprachen und Systeme vor und stellen ein auf diesen Leitlinien aufbauendes Gesamtkonzept der informatischen Bildung vor. Daneben wird auch das von der Gesellschaft für Informatik empfohlene Gesamtkonzept [GI00] erwähnt, das den Leitlinien

- Interaktion mit Informatiksystemen,
- Wirkprinzipien von Informatiksystemen,
- Informatische Modellierung und
- Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft

folgt. Ein Abgleich der beiden Konzepte findet nicht statt. Ein Vorschlag zur Verteilung der Stoffinhalte auf die einzelnen Jahrgangsstufen der Sekundarstufen, wie man ihn von der Informatikdidaktik gerade im Hinblick auf die Erarbeitung von Bildungsstandards erwartet, wird nicht geleistet.

Die „Theoretische Fundierung der Schulinformatik“ (Kapitel 3, 31 Seiten) beschränkt sich auf das Konzept der „fundamentalen Ideen“. Fundamentale Ideen destillieren gewissermaßen diejenigen Prinzipien, Methoden und Denkweisen der Informatik, die für die Bildung von dauerhaftem Wert sind, heraus. Sie sollen den Stoff für den Schüler ordnen, Zusammenhänge erkennbar machen und die Fähigkeit zum nicht-spezifischen Transfer fördern. Der „vollständige Katalog“ fundamentaler Ideen rankt sich um die drei „Masterideen“ Algorithmisierung (gemeint ist Algorithmenentwurf), strukturierte Zerlegung und Sprache [Sch93]. Wie kommt man gerade auf diese drei? Es wird zwar ein „Programm“, wie man fundamentale Ideen finden kann, angegeben - angewendet wird es nicht. Wie steht es beispielsweise um den Lebensweltbezug der Ideen „log-cost Maß“ oder „Nichtdeterminismus“? Schubert [Sch99] und Thomas [Tho02] fehlen andererseits Aspekte der technischen und angewandten Informatik. Als eine der Idee der strukturierten Zerlegung untergeordnete Idee wird „Orthogonalisierung“ genannt. Es ergibt sich mithin die spannende Frage, inwieweit die drei Ideen zueinander orthogonal sind. Modrow [Mod03] stellt eine Überlappung von Algorithmisierung und strukturierter Zerlegung fest.

Kapitel 5 (Didaktisches System, 17 Seiten) konstatiert, was der Praktiker von der Informatikdidaktik fordern darf: die Bereitstellung (oder zumindest Organisation) einer Kollektion aufeinander abgestimmter Lehr-Lern-Materialien, darunter Aufgabenklassen, Explorationsmodule und Entwurfsmuster für Lernprozesse.

Aufgabenklassen beispielsweise sollen die "Bildungserfahrung zu einem kleinen Bereich der Informatik" bündeln. Aus dem Buch "Starthilfe Informatik" [ABC02] wird exemplarisch eine (recht oberflächliche) Liste von Aufgabenklassen zu "Algorithmen und ihre Darstellung" zitiert. Leider unterbleibt die spannende Übung, diese zu einer (nicht bloß stofforientierten) Lernzieltaxonomie oder den fundamentalen Ideen in Bezug zu setzen - würde dies doch gerade den didaktischen Wert einer Aufgabe deutlich werden lassen. Woher erhält denn z.B. der aufgeführte Euklidische Algorithmus zur Berechnung des größten gemeinsamen Teilers - mal ganz abgesehen vom fehlenden Bezug zu Alltag und Lebenswelt der Schüler - seine Daseinsberechtigung im Informatikunterricht? Der intuitiv nahe- liegendere Lösungsansatz ist doch auch kognitiv einfacher!

Interessant allerdings: Schubert/Schwill leiten aus dieser formalen Repräsentation fachdidaktischen Wissens „fachliche und lehrmethodische Fehler existierender Lehrpläne“ her: Prozess, Struktur und Aufbau von Rechnern sowie Hardware-Software-Schichten fehlten besonders häufig oder würden zu knapp vorgestellt.

Die beiden anderen Konzepte der Informatikdidaktik, Dekonstruktion und informationszentrierter Zugang, sucht man im Stichwortverzeichnis vergeblich. Im Kapitel 10 (Anfangsunterricht) findet man immerhin den systemanalytischen Zugang (Abschnitt 10.2, 2 Seiten) wieder. Er wird skizziert anhand einer Unterrichtsidee von Lehmann für das Lernziel Prozeduren und Module am Beispiel des Spiels Mastermind.

Die nächsten Kapitel 6-9 folgen thematisch in etwa den Leitlinien Pläne – Sprachen – Systeme. (Kapitel 4 gehört wohl auch dazu.)

Kapitel 6 (Informatisches Modellieren und Konstruieren) gibt eine Einführung in die imperative, funktionale und prädikative Programmierung, Graphen, Automaten sowie (mit zwei hübschen Beispielen) Petri-Netze - zum Teil mit didaktischem Kommentar. Klassen-, Sequenz-, Aktivitäts-, Zustands- oder Kollaborationsdiagramm hingegen fehlen, wobei die ersteren beiden immerhin in den beiden folgenden Kapiteln behandelt werden. Schubert/Schwill (S. 212) wie schon Hubwieser [Hub03] fassen Modellierung und Implementierung unter dem Begriff „Modellierung“ zusammen. Ob das letztlich hilfreich ist, darf bezweifelt werden, ist doch Modellieren ein Sprechen über das System – im Projektteam und nicht zuletzt mit den Kunden - während Implementierung eher ein Sprechen mit dem System und dem Rechner ist.

Abschnitt 10.1 (3 Seiten) bewertet die verschiedenen programmiersprachlichen Zugänge (imperativ, prädikativ, funktional) für den Anfangsunterricht Informatik und scheint die Schlussfolgerung „Objektorientierung von Anfang an“ nahezulegen. Kapitel 7 (Objektorientierung) macht am Unterrichtsbeispiel „Entwicklung einer Bibliotheksverwaltung“ allerdings die kognitiven Voraussetzungen „objektorientierter Programmierung von Anfang an“ deutlich: etwa Variablenkonzept, strukturierte Programmierung, Modularisierung. Um den entsprechenden Lernschwierigkeiten zu begegnen, werden in einem problemorientierten Stufenkonzept die Syntax- und Semantik-Kenntnisse bereitgestellt, indem der Schüler sukzessive ergänzende Quellcodebeispiele analysiert. Selbständiges Programmieren schließt sich an.

Empfohlen wird, mindestens ein weiteres Programmierparadigma einzuführen. Tatsächlich wird vielerorts auf den Wert der prädikativen Programmierung und auf die Programmiersprache Prolog hingewiesen. Kapitel 4 „Problemlösen im Informatikunterricht“ breitet auf 18 Seiten eine entsprechende Unterrichtssequenz aus.

Kapitel 8 ist wohl der „Idee“ Interaktion (gemeint ist nach Abschnitt 8.1: Mensch-Computer-Interaktion) gewidmet und enthält eine bunte Sammlung von Themen: Informatiklabor (auch Thema von Anhang C), Unterrichtsmittel (darunter Lehrbücher, Explorationsmodule und Rollenspiele), Experimente (hier wird noch einmal - wie schon in Abschnitt 5.5 - das Explorationsmodul [Ste99] vorgestellt) sowie Lernen mit Informatiksystemen (hier werden weitere Lernumgebungen vorgestellt).

Kapitel 9 (Informatiksysteme) belegt anhand des Beispiels „Recherche und Kommunikation im Internet“, dass ein in andere Fächer integrierter Informatikunterricht kaum zum Verständnis der komplexen Wirkprinzipien von Informatiksystemen und damit zu ihrer Entmystifizierung beitragen kann.

Kapitel 11 gibt nützliche Tipps, wie die Unterrichtsmethode, für die sich der Informatikunterricht besonders anbietet, die Projektarbeit, umgesetzt und dabei eine hierarchische Organisation oder Spezialistentum vermieden werden kann.

4 Kritische Einordnung des Lehrbuchs

4.1 Informations- und kommunikationstechnische Grundbildung (IKG / ITG) und die Sekundarstufe I

Auf die verschiedenen Anforderungsniveaus des Computer-Anwenders einerseits und des Informatikers als Entwickler andererseits wird nicht abgehoben. Die Schulinformatik muss aber beides leisten. So liest man in Kapitel 1 (Was ist Informatik): „ohne sie [die Computer Literacy] ist eine geordnete Teilnahme am Leben nicht möglich“ oder von der „wachsenden Durchdringung aller Schulfächer mit Informatikmethoden und -denkweisen“ oder gar von der „integrierenden Wirkung“ der Informatik, während sich das Buch im Anschluss auf die (objektorientierte bzw. prädikative) Programmierung konzentriert. Auch Kapitel 10 (Anfangsunterricht) hilft hier leider nicht weiter. Der Abschnitt 10.5 „Exkurs: Kognitive Aspekte objektorientierter Programmierung“ (1 Seite) beleuchtet zwar, dass schon Kinder Objekte ihrer Erfahrungswelt eher über die mit ihnen möglichen Handlungen als über deren Attribute wahrnehmen. Das wichtige Moment, dass Objektorientierung beispielsweise in der Mittelstufe - dem Spiralprinzips entsprechend - durch das Objekt-Attribut-Wert-Schema anhand von Elementen von Anwenderprogrammen vorbereitet werden kann (siehe [CL03]), wird aber nicht aufgegriffen.

4.2 Der Computer als Medium zur Informationsdarbietung und Kommunikationsunterstützung

Der Orientierungsrahmen für eine Medienerziehung in der Schule der BLK [BLK95, KP98] sieht neben der Funktion des Computers als Problemlösungswerkzeug die eines Mediums zur Informationsdarbietung und Kommunikationsunterstützung. Zwar gehen Schubert/Schwill im Kapitel 9.2 „Informations- und Kommunikationssysteme als Unterrichtsgegenstand“ auf Suchverfahren im WWW und insbesondere auf die auf TCP/IP basierende Internet-Architektur ein, doch die Gestaltung, Veröffentlichung und Kommunikation von Information mit Hilfe von Textverarbeitungssystem, Tabellenkalkulationsprogramm,

Präsentationssoftware, HTML bzw. e-Mail wird nicht betrachtet. Datenbanken werden zwar erwähnt, auf eine didaktische Aufbereitung des Themas wird verzichtet.

4.3 Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft

Als vierte Leitlinie für den Informatikunterricht in der Schule nennt das von der Gesellschaft für Informatik empfohlene Gesamtkonzept [GI00] die Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft.

Schubert/Schwill mahnen zwar an, im Unterricht im Zusammenhang mit Informatiksystemen auf Aspekte wie

- Informationssicherheit und Datenschutz
- Sicherheitsrisiken wie Viren und Gegenmaßnahmen
- Gestaltung im Sinne der „Analyse und Bewertung fertiger Systeme“, Benutzerfreundlichkeit und Software-Ergonomie

einzugehen (S. 48, S. 59, S. 65), sie bereiten sie aber didaktisch nicht auf. Insbesondere geben Schubert/Schwill (selbst in Kapitel 9: Informatiksysteme) keine Hinweise dazu, wie das Informatiksystem aus der Perspektive eines sozio-technischen Handlungssystems mit Aspekten der

- prozessorientierten Software-Entwicklung unter Einbeziehung der Nutzer
- Gestaltung zugehöriger Handlungs- und Kommunikationsprozesse (Organisationsentwicklung)
- sozialverträglichen Technikgestaltung als interessen geleiteten Entscheidungsprozess

im Unterricht umgesetzt werden kann. Diese Aspekte von Informatiksystemen versucht der systemorientierte Didaktikansatz mit dem Konzept der Dekonstruktion zu thematisieren [Mag00]. Auch Abschnitt 10.2 „Der systemanalytische Zugang“ bleibt beim Aspekt der Bewältigung (im Sinne der Software-Entwicklung) komplexer Systeme stehen.

Symptomatisch für diese Sichtweise ist neben dem informatik-intrinsischen Konzept der fundamentalen Ideen auch das folgende Beispiel: Schubert/Schwill bezeichnen (wie übrigens auch [GI00]) Programmierung und Implementierung als die „Simulation von Modellen“ im Sinne einer Realisierung des Modells (S. 67, S. 99). An anderen Stellen wird sichtbar, dass Simulation jedoch gemeinhin auf eine fachliche Metaebene abhebt: Simulation (von meist dynamischen Systemen) im Sinne der Entwicklung eines Modells für einen Ausschnitt der realen Welt zu Forschungs-, Planungs- und Beurteilungszwecken (z.B. S. 182, 212) oder Simulation im Sinne von Emulation (S. 166).

4.4 Fundamentale Ideen, Bildungsstandards und das didaktische System

Wie vermittelt man fundamentale Ideen? Was wird bei einem Informatik-Curriculum, dass dieses Konzept umzusetzen versucht, anders sein? Schließlich sollen sich die Ideen in unterschiedlichen Unterrichtsgegenständen widerspiegeln - und jeder Gegenstand wiederum bedarf einer Vorbereitung, hat seine Entwicklung (didaktische Lernstufen). Und auch in der Vergangenheit haben Lehrer die strukturellen Analogien über verschiedene Themen hinweg deutlich gemacht. Schließlich: Soll Unterrichtsstoff in Zukunft ausgespart werden, wenn nicht hinreichend durch fundamentale Ideen legitimiert - weil damit wohl nicht robust genug gegenüber Paradigmenwechseln? Und wenn ja, welchen Unterrichtsstoff betreffe das bitte beispielsweise? Oder findet man fundamentale Ideen - weil sie so fundamental sind - nicht überall?

Diese für den Praktiker „fundamentalen Fragen“ bleiben leider unbeantwortet. Ein Zufall? Diese Diskussion zeigt meines Erachtens eher, dass fundamentale Ideen nicht für die Definition von Bildungsstandards taugen - auch wenn Schubert/Schwill damit die fachwissenschaftliche Auswahl meinen begründen zu können (S. 134). Auf der „subjektiven Mikroebene“ allerdings können sie dem Praktiker sehr wohl eine Richtschnur sein, z.B. bei

- der Wichtung von Unterrichtsinhalten,
- der Auswahl von Aufgabenbeispielen oder der Art und Weise, wie man eine Aufgabe angeht.

Das auf Seite 139 herausgehobene Beispiel des Euklidischen Algorithmus zeigt dies sehr schön: diese Aufgabe gehört gerade *nicht* in den Kontext „elementare Kontrollstrukturen“ (bei Schubert/Schwill: fundamentale Idee „Programmierkonzepte“), sondern eben in den Kontext der Idee „Evaluation – Komplexität“. Entsprechend könnte man die Schüler hier zunächst ihre eigene Umsetzung für die Problemstellung (vermutlich den einfachen direkten Algorithmus) entwickeln und anschließend beide Ansätze vergleichen lassen. Fundamentale Ideen gehören also ins didaktische System!

Die vernetzende Wirkung der fundamentalen Ideen deutlich machen und den Diskussionsprozess darüber, was als fundamental zu gelten hat, sicherlich versachlichen würde eine Übersicht, an welchen Stellen im Gesamtkonzept der informatischen Bildung sich eine fundamentale Idee im Sinne eines Spiralprinzips wiederfindet (wiederfinden soll) und in welcher Beziehung sie zu den Leitlinien und Aufgabenklassen steht.

5 Sonstiges

Die Implementierung der Unterrichtsbeispiele erfolgt in den Programmiersprachen Prolog, Python und in Java. Der im Anhang abgedruckte Quellcode ist auch online verfügbar [Sch04].

Ein grober Schnitzer: NP-vollständig wird fälschlicherweise als „zur Klasse NP gehörig“ definiert (S. 176), NP wird entsprechend als die Klasse der praktisch unlösbaren, da nur durch vollständiges Durchmustern lösbarer Aufgaben bezeichnet, falls $P \neq NP$ (S. 63).

Ärgerlich auch das Layout: man denkt schon, das Kapitel wäre zu Ende, aber es sind nur größere Grafiken, die etliche halbe Leerseiten erzeugen.

Das Komma vor dem erweiterten Infinitiv hätte die Lesbarkeit an etlichen Stellen sicherlich gesteigert. Eine ganze Reihe von Tippfehlern hätte eigentlich schon das Textverarbeitungsprogramm bemerken müssen.

Das Literaturverzeichnis ist nicht korrekt sortiert und enthält Einträge doppelt (S. 333: Schwill 1993). Manche Einträge fehlen ganz (S. 34: Humbert 1999 oder S. 51: Hubwieser 2000), manchmal fehlen nur Verlag und Ort (wie bei Price et al.), die Literaturverweise auf die Lehrbücher „Zeising: Praktische Informatik“ und „Metzler Informatik“ schließlich werden gar im Text gegeben (S. 23). Das Literaturverzeichnis folgt keiner einheitlichen Syntax und ist nicht vollständig ins Deutsche übersetzt (z.B. „pp.“ anstatt „S.“ oder „eds.“ anstatt „Hrsg.“).

Auch das Register enthält Einträge doppelt (das eine Mal groß-, das andere Mal kleingeschrieben oder einmal im Singular, einmal im Plural). Es fehlen Verweise auf die angesprochenen Lernumgebungen und Programmiersprachen.

6 Schluss

Trotz der z.T. wenig pointierten Darstellung und einer gewissen Laxheit in der Präsentation und bei der Verwendung von Begriffen ist dieses Lehrwerk von den „Ideen“ her (fundamentale Ideen, didaktisches System) interessant. Für den Praktiker werden didaktische Systeme meines Erachtens von enormem Wert sein. Der Informatikdidaktik wird hier eine Vision aufgezeigt. Erste Ansätze, die sich dieser Aufgabe annehmen, stimmen erwartungsvoll, siehe z.B. [Bri03, ETH, HyFIS, IML, life3]. Ein Praktiker wiederum, der den Hubwieser [Hub03] ausgelesen und sich auf den einschlägigen Portalen mit dem Angebot an Lernsoftware und Unterrichtsmaterialien vertraut gemacht hat, wird - je nach Geschmack - sicher von beispielsweise der Unterrichtssequenz zur prädikativen Programmierung bzw. den Ausführungen zum projektorientierten Unterricht profitieren.

Wenn man übrigens die Formalisierung didaktischen Wissens weitertriebe und z.B. (gewichtete) Aufgabenklassen (kapazitierten) Leitlinien, Lernzielen oder fundamentalen Ideen zuwies, führte uns dies auf ein Generalized Assignment Problem. Zum Glück wäre dies NP-schwer zu lösen, so dass es nicht wunder nähme, wenn man sich nicht auf einen Bildungsstandard einigen könnte...

7 Referenzen

- [ABC02] Appelrath, H.-J., Boles, D., Claus, V.: Starthilfe Informatik. Teubner, Stuttgart, 2. Auflage, 2002.
- [BLK95] Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung: Medienerziehung in der Schule – Orientierungsrahmen. BLK-Materialien zur Bildungsplanung und zur Forschungsförderung Heft 44, Bonn, 1995. Download: http://www.labi-berlin.nubb.dfn.de/bibliothek/positionspapiere/blk_95.htm
- [Bri03] Brinda, T.: Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II. Dissertation, Universität Siegen, 2003. Download: <http://www.die.informatik.uni-siegen.de/forschung/brinda.pdf>
- [CL03] Caba, H., Landerer, C.: Fundamentale Ideen im Informatikunterricht am Beispiel der objektorientierten Softwareentwicklung. In: Schulinformatik in Österreich, quo vadis? (Hrsg. Micheuz, P.), CD Austria - Das Multimedia-Magazin für Österreichs Schulen 10, 2003, 20-24. Download: <http://www.schule.at/gegenstand/informatik>
- [CS03] Claus, V., Schwill, A.: Duden Informatik. Bibliographisches Institut, Mannheim, 2003.
- [ETH] Informatik auf EducETH. <http://www.educeth.ch/informatik/>
[Redaktion: Reichert, R.]
- [GI00] Fachausschuss 7.3 "Informatische Bildung in Schulen" der Gesellschaft für Informatik e.V.: Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. Beilage zu LOG IN 20 (2), 2000. Download: <http://www.informatische-bildung.de/>
- [Hub03] Hubwieser, P.: Didaktik der Informatik. Springer, Berlin, 2003.
- [HyFIS] HyFISCH - HyperForum Informatik in der Schule. Informieren - Kommunizieren – Produzieren. <http://www.hyfisch.de/HyFISCH>
- [IML] Informatik-Module zur Lehrerbildung. <http://www.die.informatik.uni-siegen.de/iml> [Gesamtleitung: Schubert, S.]
- [Kara] Programmieren lernen mit Kara. <http://www.educeth.ch/informatik/karatojava/>
[Kontakt: Reichert, R., Departement Informatik, ETH Zürich]
- [Karol] Robot Karol. <http://www.schule.bayern.de/karol/>
[Kontakt: Freiburger, U.]
- [KP98] Koerber, B., Peters, I.-R.: Informatische Bildung in Deutschland – Die Wurzeln der Zukunft. In: Informatische Bildung in Deutschland - Perspektiven für das 21. Jahrhundert (Hrsg. Koerber, B., Peters, I.-R.), LOG IN Verlag, Berlin, 1998. Download: http://log-in.fachdid.fu-berlin.de/IBiD/Koerber_Peters/index.html
- [life3] life3 - Lernwerkzeuge für den Informatikunterricht: Einsetzen, Evaluieren und (Weiter)Entwickeln. <http://life.upb.de>
[Lernwerkstatt zum Thema 'Objektorientiertes Modellieren' mit Unterrichtsmaterialien und -werkzeugen, gefördert vom Universitätsverbund MultiMedia NRW, Ansprechpartner: Magenheim, J.]
- [LEO] LEO - Lernumgebung für objektorientiertes Modellieren im Informatikunterricht. <http://www.die.informatik.uni-siegen.de/pgleo/>
[Entwickelt von einer studentische Projektgruppe der Universität Dortmund (2001-2002) sowie der Fachgruppe "Didaktik der Informatik und E-Learning" der Universität Siegen (2002-2004), Leitung: Brinda, T.]
- [Mag00] Magenheim, J.S.: Informatiksystem und Dekonstruktion als didaktische Kategorien – Theoretische Aspekte und unterrichtspraktische Implikationen einer svstemorientierten Didaktik der Informatik. Taaunasbeitraa zur GI-Tauna

- "Informatik - Ausbildung und Beruf 2000", 2000.
Download: <http://ddi.uni-paderborn.de/didaktik/Veroeffentlichungen>
- [Mod03] Modrow, E.: Pragmatischer Konstruktivismus und fundamentale Ideen als Leitlinien der Curriculumentwicklung am Beispiel der theoretischen und technischen Informatik. Dissertation, Martin-Luther-Universität Halle-Wittenberg, 2003. Download: <http://sundoc.bibliothek.uni-halle.de/diss-online/03/03H066/>
- [MOOP] Mediothek Objektorientierte Programmierung. <http://www.learnline.nrw.de/angebote/oop/medio> [betreut durch Hildebrecht, H.]
- [OLLI] Oldenburger Lernprogramme zur Informatik (OLLI). Carl-von-Ossietzky Universität, Oldenburg. <http://olli.informatik.uni-oldenburg.de/> [Zuletzt geändert am 20.02.2004, Kontakt: Faltin, N.]
- [Sch93] Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik 1, 1993, 20-31.
Download: <http://www.informatikdidaktik.de/Forschung/Schriften/ZDM.pdf>
- [Sch98] Schwill, A.: Projektunterricht - Grundlagen und Beispiele. Vortrag in der Rahmenplanbegleitenden Fortbildung Informatik, Pädagogisches Landesinstitut Brandenburg (PLIB), Ludwigsfelde, 1997-1998. Download: <http://ddi.cs.unipotsdam.de/Forschung/VortragsfolienProjektunterricht.pdf>
- [Sch99] Schubert, S.: Einführung in die Didaktik der Informatik. Vorlesungsskript, Universität Dortmund, 1999. Download: http://eldorado.uni-dortmund.de:8080/FB4/13/Lehre/vorlesungsskripte/schubert/ddi_1.pdf
- [Sch03] Schwill, A.: Didaktik der Informatik. <http://www.informatikdidaktik.de/Forschung/Schriften/DidaktikLehrbuch2003> [Webseite zum Lehrbuch: Schubert, S., Schwill, A.: Didaktik der Informatik. Spektrum Akademischer Verlag, Berlin, 2004, zuletzt geändert am: 26.08.2003]
- [Sch04] Schubert, S.: Onlinematerialien zum Buch "Didaktik der Informatik". http://www.die.informatik.uni-siegen.de/DIE_BIB/Lehre/didaktikbuch [Letzte Änderung: 22.07.2004]
- [SIMBA] BMBF-Verbundprojekt SIMBA - Schlüsselkonzepte der Informatik in multimedialen Bausteinen unter besonderer Berücksichtigung der spezifischen Lerninteressen von Frauen.
<http://www.die.informatik.uni-siegen.de/simba/medienobjekte> [Projektmanagement und -evaluation: Schubert, S.]
- [Ste99] Steinkamp, D.: Informatik-Experimente im Schullabor. Diplomarbeit, Universität Dortmund, 1999.
<http://www.die.informatik.uni-siegen.de/lehre/diplom/steinkamp/> [Experimentierumgebung zu Kommunikationsprozessen beim Anfordern einer Webseite, letzte Änderung: 28.07.2003]
- [Tho02] Thomas, M.: Didaktik der Informatik II. Vorlesungsskript, Universität Dortmund, 2002.
Download: http://www.hyfisch.de/Personen/marco/vorl_ddi2_02/lit.pdf

*Dr. Thomas Emden-Weinert
Bad Homburg, im Dezember 2004*