

Entwicklung von Komponenten eines Lehr-Lern-Konzeptes zum objektorientierten Modellieren

Torsten Brinda

Didaktik der Informatik und multimediale Lehr-Lern-Systeme
Universität Siegen
D-57068 Siegen

brinda@informatik.uni-siegen.de
URL: <http://www.didaktik-der-informatik.de/>

Hinweis zu diesem Dokument: Der hier vorliegende Text entspricht in großen Teilen der Publikation [Br01] des Autors. Er wurde um Folien aus einem Vortrag des Autors, der im Rahmen der Fachtagung Informatik und Schule 2001 (INFOS2001) in Paderborn gehalten wurde, damit verbundene Erläuterungen, sowie um Verweise und Kommentare das Thema betreffender multimedialer Lern- und Lehrmaterialien ergänzt.

Zusammenfassung

Im Rahmen dieses Beitrags werden zunächst Argumente dafür geliefert, dass objektorientiertes Modellieren (OOM) als Problemlösungsmethode dazu geeignet ist, Ziele allgemein bildenden Informatikunterrichts zu realisieren. Aufgrund des Mangels an Lehr-Lern-Materialien zum OOM in der Fachdidaktik, werden vorhandene Lehrstrategien und Lehr-Lern-Materialien in der Fachwissenschaft untersucht mit dem Ziel der Entwicklung von Strategien für die Gestaltung entsprechender Elemente für den Informatikunterricht. Kriterien und Vorgehensweisen zur Auswahl, Strukturierung und Repräsentation der kognitiven Beziehungen von Lerninhalten und Kompetenzen, sowie zur Konstruktion von Übungsaufgaben zu OOM aus Aufgabenklassen werden entwickelt, um die Planung und Ausgestaltung von Informatikunterricht zu OOM zu erleichtern, dessen Vergleichbarkeit zu erhöhen und Qualitätsstandards zu etablieren.

Abstract

In this contribution arguments are given, that object-oriented modelling (OOM) as a problem solving method is suitable to achieve the objectives of general secondary education in informatics. Because of a lack of learning and teaching material on OOM within didactics of informatics, available teaching strategies as well as learning and teaching material of the university subject are analyzed to develop strategies for the design of corresponding elements for secondary education in informatics. Criteria and methods for selecting, structuring and representing cognitive relationships of learning content and competences, as well as for constructing exercises on OOM by the use of exercise classes are developed in order to simplify the planning and arrangement of secondary informatics education on OOM and to raise its comparability and to establish quality standards.

1 Motivation

In der Arbeitsgruppe "Didaktik der Informatik" der Universität Dortmund (ab Ende 2002 an der Universität Siegen) wird seit 2000 ein Lehr-Lern-Konzept für objektorientiertes Modellieren (OOM) im Informatikunterricht entwickelt [Br00a], [Br00b]. Die Entwicklung ist eine Reaktion auf sich vom imperativen Problemlösen mit starker Betonung der Programmierung hin zum informatischen Modellieren verändernde Schwerpunkte bei den durch Informatikunterricht zu vermittelnden Kompetenzen (vgl. Ansatz Hubwieser [Hu00]), den hohen Bedarf für Konzepte zur Umsetzung didaktischer Theorien bei Informatiklehrenden und die Tatsache, dass objektorientierte Modellierung zu einem fundamentalen Gegenstandsbereich der Fachwissenschaft im Sinne der fundamentalen Ideen der Informatik nach Schwill [Sc93] geworden ist und damit intensiver als bislang im Informatikunterricht berücksichtigt werden sollte. Da OOM bislang nicht verbreitet im Informatikunterricht thematisiert wird, existiert noch kein Konsens darüber, welche objektorientierten Basiskonzepte, Modellierungstechniken und -strategien, sowie welche Grundzüge einer objektorientierten Programmiersprache von den Schülerinnen und Schülern erlernt werden sollen, damit sie die für die Realisierung der Ziele modernen, allgemein bildenden Informatikunterrichts [GI00], wie das Durchdringen von "Wirkprinzipien von Informatiksystemen" und das Erlangen von Gestaltungskompetenz im Sinne des "informatischen Modellierens", erforderlichen Fach- und Methodenkompetenzen erwerben können. Im World Wide Web und in Fachzeitschriften, wie LOG IN, publizierte Unterrichtsbeispiele zur Objektorientierung zeigen weiterhin eine starke Orientierung hin auf die Programmierung. Die Diskussion von Programmiersprachen in bezug auf ihre Unterrichtseignung dominiert dabei konzeptuelle Überlegungen. Um die Planung und Ausgestaltung von Informatikunterricht zu OOM zu erleichtern, dessen Vergleichbarkeit zu erhöhen und Qualitätsstandards zu etablieren, werden im Rahmen dieses Beitrags Kriterien und Vorgehensweisen zur Strukturierung und Repräsentation der Beziehungen von Kompetenzen und objektorientierten Konzepten, sowie zur Konstruktion von Übungsaufgaben zum objektorientierten Modellieren aus Aufgabenklassen entwickelt. Es handelt sich dabei um wesentliche Bestandteile eines didaktischen Systems zu OOM (vgl. [BS01]).

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Motivation

- Informatikunterricht wurde dominiert von der Programmierung (Programmiersprache und Softwareentwicklungsumgebung)
 - Wandel bei den Schwerpunkten der durch Informatikunterricht zu vermittelnden Kompetenzen vom imperativen Problemlösen hin zum informatischen Modellieren
 - OOM ist zu einem fundamentalen Gegenstandsbereich der Fachwissenschaft geworden und sollte daher stärker als bisher im Unterricht berücksichtigt werden
 - da OOM bislang nicht verbreitet im Informatikunterricht thematisiert wird, existiert noch kein Konsens darüber, welche objektorientierten Basiskonzepte, Modellierungstechniken und -strategien, Grundzüge von Sprachen erlernt werden sollten, um Ziele modernen, allgemein bildenden Informatikunterrichts zu erfüllen
- ⇒ Entwicklung eines Lehr-Lern-Konzeptes für objektorientiertes Modellieren im Informatikunterricht

2 Entwicklung von Kompetenzen im Informatikunterricht der Sekundarstufe II

Schülerinnen und Schülern werden in ihrem Alltag und späteren, beruflichen Werdegang auf vielfältige Weisen mit komplexen Informatiksystemen konfrontiert, sei es als Anwender, Betroffener, Entscheider, Planer, Entwickler oder als Administrator. Zur Vorbereitung auf die sich durch komplexe Informatiksysteme zunehmend verändernde Arbeits- und Lebensweise von Menschen heißt es in den "Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen" der Gesellschaft für Informatik [GI00]: "Der Umgang mit digital dargestellter Information und die Beherrschung von Informatiksystemen stellen folglich unverzichtbare Ergänzungen der traditionellen Kulturtechniken Lesen, Schreiben und Rechnen dar.". Die im Rahmen der Empfehlungen charakterisierte Bildung orientiert sich an vier Leitlinien, von denen hier die Linien "Wirkprinzipien von Informatiksystemen" und "informatische Modellierung" von besonderer Bedeutung sind. Um Informatiksysteme beherrschen und deren prinzipielle Möglichkeiten und Grenzen für einen gegebenen Zweck beurteilen zu können, ist ein Grundverständnis der Wirkprinzipien erforderlich, das nur durch Fach- und Methodenkompetenz zur Analyse und Modellierung von Systembausteinen im Informatikunterricht erworben werden kann und wofür deren Anwendung allein nicht ausreichend ist.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Entwicklung von Kompetenzen im Informatikunterricht

- Lernende werden in ihrem Alltag und späteren beruflichen Werdegang als Anwender, Betroffener, Entscheider, Planer, Entwickler, Administrator mit Informatiksystemen konfrontiert
- zur Vorbereitung auf die sich durch komplexe Informatiksysteme zunehmend verändernde Arbeits- und Lebensweise von Menschen:
„Der Umgang mit digital dargestellter Information und die Beherrschung von Informatiksystemen stellen folglich unverzichtbare Ergänzungen der traditionellen Kulturtechniken Lesen, Schreiben und Rechnen dar.“
(GI 2000)
- Leitlinien „Wirkprinzipien von Informatiksystemen“ und „informatische Modellierung“
- um Informatiksysteme beherrschen und deren prinzipielle Möglichkeiten und Grenzen beurteilen zu können, ist Grundverständnis dieser Bereiche erforderlich, Anwendung allein reicht dazu nicht aus

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 5 / 28

Neben der Tatsache, dass das objektorientierte Modellieren zu einem fundamentalen Gegenstandsbereich der Informatik geworden ist und damit in wissenschaftspropädeutischem Informatikunterricht der Sek. II thematisiert werden muss, besteht die begründete Vermutung, dass der objektorientierte Ansatz aufgrund zahlreicher Entwicklungen systematischer Vorgehensweisen und anschaulicher Darstellungsformen in der Fachwissenschaft gut dazu geeignet ist, die oben genannten Ziele modernen Informatikunterrichts zu realisieren. Die Arbeiten verschiedener Fachdidaktiker liefern hierzu eine Reihe von Indizien. Füller hat darauf hingewiesen, dass "ein objektorientierter Ansatz verwendet werden kann, um Anwendersysteme zu analysieren und neutral zu vergleichen" [Fü99, S. 190]. Mit der "Dekonstruktion von Informatiksystemen als Unterrichtsmethode" (Magenheim u. a. [MSH99]) liegt ein Zugang zu objekto-

orientierten Sichtweisen im Informatikunterricht vor, bei dem die Analyse objektorientiert gestalteter Informatiksysteme zum durchgängigen Prinzip wird. Hubwieser hat im Rahmen seines "informationszentrierten Ansatzes" allgemein bildenden Informatikunterricht mit Modellierung als inhaltlichem Kern begründet. Er merkt an [Hu00, S.85], dass unterrichtliche Betrachtungen des Modellierungsvorgangs oft zu rein philosophischen, wenig schülergemäßen Exkursen gerieten, da man bis vor wenigen Jahren nicht über geeignete geistige Techniken verfügte, diesen systematisch und in angemessener Tiefe im Unterricht umzusetzen. Er weist in diesem Zusammenhang auf die Arbeiten zu objektorientierten Modellierungstechniken und Vorgehensweisen von Rumbaugh, Booch und Jacobson (s. z. B. [BRJ98]) hin, von denen er einige für geeignet hält, diese methodische Lücke zu schließen.

Die genannten informatischen Bildungsziele können aber auch mit dem objektorientierten Ansatz nur dann erreicht werden, wenn dabei der konzeptionelle Wandel vom Programmieren zum informatischen Modellieren vollzogen wird.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Entwicklung von Kompetenzen im Informatikunterricht

- **begründete Vermutung, dass objektorientierter Ansatz aufgrund zahlreicher Entwicklungen in der Fachwissenschaft gut dazu geeignet ist, die Bildungsziele zu realisieren**
- **Indizien (Auswahl):**
 - „Ein objektorientierter Ansatz kann verwendet werden, um Anwendersysteme zu analysieren und neutral zu vergleichen.“ (Füller 1999)
 - „Dekonstruktion von Informatiksystemen als Unterrichtsmethode“ (Magenheim u. a. 1999) – Analyse objektorientiert gestalteter Informatiksysteme wird zum durchgängigen Prinzip
 - „informationszentrierter Ansatz“ (Hubwieser 1997) – Eignung ausgewählter objektorientierter Modellierungstechniken für Umsetzung des Modellierungsvorgangs im Unterricht
- **aber: informatische Bildungsziele auch mit dem objektorientiertem Ansatz nur erreichbar, wenn dabei der konzeptionelle Wandel vom Programmieren zum Modellieren vollzogen wird**

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 6 / 26

Zu starke Betonung der Programmierung bewirkt, dass Lernende zunächst sehr viel Faktenwissen zu einer Programmiersprache erwerben müssen, bevor sie die angestrebte Gestaltungskompetenz erlangen. Aufgrund des Mangels an Lernhilfen und Unterrichtsmaterialien zum Modellieren, lässt sich in der Schulpraxis weiterhin verbreitet die Betonung der Programmierung beobachten, allerdings zunehmend bei Verwendung objektorientierter Programmiersprachen. Zur Beseitigung dieses Mangels will die hier vorgestellte Arbeit beitragen.

Für den Bereich der Wirkprinzipien von Informatiksystemen wurde in der Fachgruppe "Didaktik der Informatik" an der Universität Dortmund bereits im Rahmen einer Diplomarbeit [St99] ein Konzept entwickelt (vgl. Abbildung 1) und exemplarisch erprobt, das einen explorativen Zugang zu Fachinhalten (im Beispiel aus dem Fachgebiet "Rechnernetze und verteilte Systeme") ermöglicht.

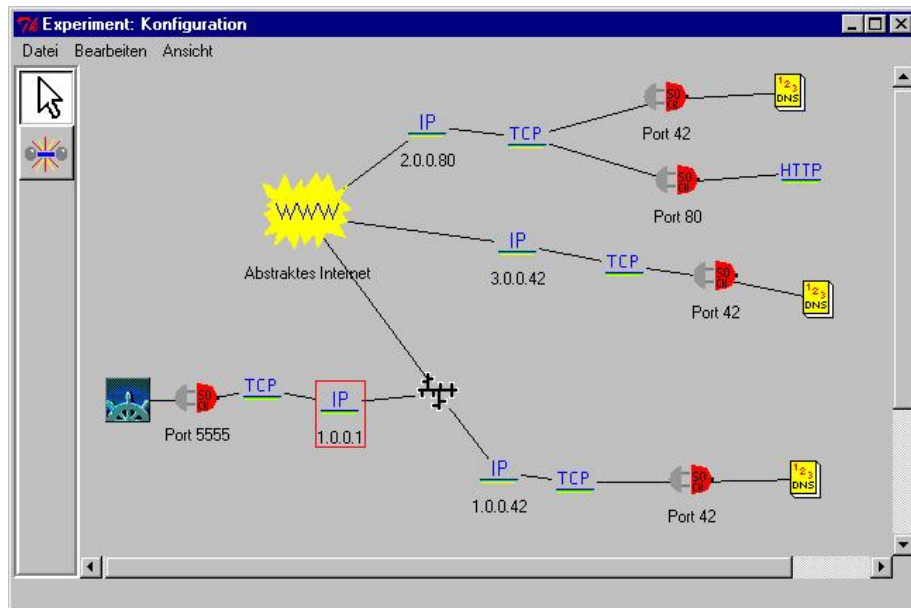


Abbildung 1: Erkundungsbaustein für das Anfordern einer Webseite

In Rahmen der Forschungsarbeit des Autors werden prozess- und ergebnisorientierte Lernhilfen für den Bereich des objektorientierten Modellierens konzipiert, entwickelt und in einem Einsatzkonzept verknüpft.

3 Fachwissenschaftliche Erkenntnisse zum objektorientierten Modellieren

Für die Suche nach Lehr-Lern-Materialien zum OOM in der Fachdidaktik steht ein großer Reichtum an Lehrbüchern, Lernhilfen, Übungsaufgaben und Werkzeugen in der Fachwissenschaft zur Verfügung. Obwohl sich diese Materialien an Informatikstudierende und Software-Entwickler und damit an völlig andere Zielgruppen als Lernende im Informatikunterricht richten, so verbindet alle dennoch, dass sie Anfänger beim objektorientierten Modellieren sind. Aus diesem Grund werden vorhandene Lehrstrategien und Lehr-Lern-Materialien in der Fachwissenschaft untersucht mit dem Ziel der Entwicklung von Strategien für die Gestaltung entsprechender Elemente für Informatikunterricht zu OOM.

3.1 Begriffsbildung

Objektorientierte Basiskonzepte, wie Objekt, Klasse, Assoziation oder Vererbung, sind erforderlich, um objektorientierte Modellierungstechniken und die Lösung von Problemen durch ein Geflecht kooperierender Objekte verstehen zu können. Deshalb stehen sie zumeist am Anfang entsprechender Ausbildungsabschnitte. In Lehrbüchern zu OOM (analysiert wurden hier Publikationen von Wirfs-Brock u. a. [WWW90, S. 17-28], Jacobson u. a. [Ja93, S. 44-68], Rumbaugh u. a. [Ru93, S. 27-59], Booch [Bo94, S. 109-186] und Meyer [Me97]) werden diese Basiskonzepte oft über Metaphern eingeführt, indem ein Bezug zu Begriffen der Lebenswelt hergestellt wird. Wirfs-Brock u. a. veranschaulichen den Klassenbegriff als Schablone für eine spezielle Art von Objekten und als Fabrik, die Objekte produziert [WWW90, S. 22]. Ein Objekt sehen sie als "Black Box" mit öffentlicher Schnittstelle und geheimem Inhalt [WWW90, S. 18]. Jacobson u. a. führen die Klasse als Bauplan für den internen Aufbau strukturgleicher Objekte ein [Ja93, S. 50]. Booch beschreibt eine Beziehung zwischen zwei Objekten als Pfad, über den Nachrichten versendet werden können [Bo94, S. 129] und Aggregatobjekte als Container [Bo94, S. 166].

Bei der Reihenfolge der Erarbeitung der Grundbegriffe lassen sich in der Fachliteratur folgende Varianten erkennen:

1. Objekt, Nachrichtenaustausch und Beziehungen zwischen Objekten, Klasse, Beziehungen zwischen Klassen (Wirfs-Brock u. a., Jacobson u. a., Booch),
2. Objekt, Klasse, Beziehungen zwischen Objekten und Klassen (Rumbaugh u. a.),
3. Klasse, Objekt, Beziehungen zwischen Objekten und Klassen (Meyer).

Während die ersten beiden Varianten keine Vorkenntnisse aus dem imperativen Paradigma erfordern und mit einem lebensweltlichen Objektbegriff beginnen, setzt die dritte Variante auf den Begriff des abstrakten Datentyps (ADT) auf und entwickelt diesen zum Klassenbegriff weiter. Die Grundbegriffe werden i. d. R. schrittweise und systematisch entwickelt. Zuerst eingeführte Begriffe werden verwendet, um nachfolgende daraus abzuleiten bzw. darauf aufzubauen. Teilweise werden Begriffe bereits verwendet, bevor sie formal eingeführt worden sind. Vorwissen aus anderen Paradigmen wird dazu verwendet, um Lerninhalte der Objektorientierung daran anzuknüpfen. Die Erarbeitung des Klassenbegriffs aus dem ADT ist ein Beispiel hierfür. Bei Meyer [Me97, S. 215] finden sich auch Ansatzpunkte für einen Bezug zum funktionalen Ansatz. Beim funktionalen Ansatz kann z. B. eine von zwei Variablen abhängige Funktion $f(x,y)$ überführt werden in eine Funktion höherer Ordnung $(g(x))(y)=f(x,y)$. Dies wird als "currying" bezeichnet. Beim objektorientierten Ansatz gibt es einen analogen Zusammenhang. So kann die Funktion f hier transformiert werden in $x.g(y)$ bzw. $y.g'(x)$.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der Ddi (INFOS2001, 18.09.01)

Fachwissenschaftliche Erkenntnisse zum OOM

- **Suche nach Lehr-Lern-Materialien in der Fachdidaktik vs. großer Reichtum an Lehrbüchern, Übungsaufgaben etc. in der Fachwissenschaft**
- **Begriffsbildung**
 - Beginn mit objektorientierten Basiskonzepten
 - Einführung oft über Metaphern (Bezug zur Lebenswelt), z.B. Schablone, Bauplan, Fabrik für Klasse (Wirfs-Brock 1990) oder Container für Aggregat (Booch 1994)
- **Erarbeitung der Grundbegriffe (Varianten der Reihenfolge)**
 - Objekt, Nachrichtenaustausch, Beziehungen zwischen Objekten, Klasse, Beziehungen zwischen Klassen (Wirfs-Brock, Jacobson, Booch)
 - Objekt, Klasse, Beziehungen zwischen Objekten und Klassen (Rumbaugh u. a.)
 - Klasse, Objekt, Beziehungen zwischen Objekten und Klassen (Meyer)
- **Anknüpfung an Vorwissen aus anderen Paradigmen**
 - imperativ: ADT führt zu Klasse
 - funktional: $f(x,y)=(g(x))(y)$ führt zu $x.g(y)$

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 8 / 26

Fachkonzepte werden hier also schrittweise und systematisch entwickelt. Dabei werden Wissen über die Lebenswelt und Vorkenntnisse zu anderen Problemlösungsmethoden benutzt, um die neuen Konzepte zu veranschaulichen.

3.2 Statische und dynamische Aspekte

Bei der objektorientierten Analyse und Konstruktion von Informatiksystemen lassen sich statische und dynamische Modelle unterscheiden. Während es das Ziel des statischen Modells ist, einen Überblick über den Aufbau bzw. die Architektur des Systems, z. B. in einem oder mehreren Klassendiagrammen, zu geben, zielt das dynamische Modell darauf ab, einen Überblick über zeitliche Systemabläufe, wie Objekterzeugung, Objektzustandsveränderung durch Nachrichtenaustausch mit anderen Objekten oder Objektzerstörung, z. B. in Interaktionsdiagrammen, zu geben. Beide Aspekte sind gleichermaßen bedeutend, untrennbar mit jedem nichttrivialen Informatiksystem verbunden und beeinflussen einander wechselseitig. Objektorientierte Modellierungstechniken stellen eine Reihe von Darstellungsmitteln zur Konstruktion solcher Modelle bereit. Objektorientierte Vorgehensweisen geben Hinweise zur systematischen Erstellung von statischem und dynamischem Modell (z. B. [Ba99, S.119ff]) und betonen, dass beide aufgrund der Abhängigkeit parallel entwickelt werden sollten. Die Einführung in die verschiedenen Techniken erfolgt in der Literatur, ähnlich wie bei den Basiskonzepten, streng systematisch. Praktische Übungsbeispiele zum OOM (s. z. B. [Ru93]) zeigen allerdings, wie objektorientierte Basiskonzepte, Modellierungstechniken und Vorgehensweisen erst einzeln, dann kombiniert eingeübt werden können.

In der Fachgruppe "Didaktik der Informatik" wurden 2001 interaktive Animationen zur Visualisierung objektorientierter Grundkonzepte (<http://www.didaktik-der-informatik.de/flash>, Hinweis: die Animationen können nur in Ihrem Webbrowser angezeigt werden, wenn Ihr Browser Flash-Animationen unterstützt oder Sie ein entsprechendes Plugin installiert haben!) mit Macromedia Flash entwickelt. Ziele waren es, die Eignung solcher Animationen für den Lernprozess, den Produktionsaufwand und die Darstellungs- und Interaktionsmöglichkeiten des Werkzeugs Flash zu erkunden. Der Einsatz der Animationen im Rahmen eines E-Learning-Praktikums der Schuppperuni des Fachbereichs Informatik für Oberstufenschülerinnen führte zu überwiegend positiver Resonanz.

Aus Sicht der Informatik ist es erforderlich, dass Anfänger nicht nur die prinzipielle, sondern aus ökonomischen Gründen die Konstruktion guter, objektorientierter Modelle erlernen in Sinne von Kriterien, wie z. B. Wiederverwendbarkeit und Wartbarkeit. Fowler weist darauf hin, dass bestimmte Modellierungstechniken diesbezügliche Mängel schnell ersichtlich machen. So z. B. kann in Interaktionsdiagrammen anhand des Nachrichtenaustauschs zwischen Objekten schnell erkannt werden, ob Aufgaben gleichmäßig zwischen Objekten verteilt sind oder ob der Entwurf zu stark zentralisiert ist [FS97, S. 8]. Die Verwendung graphischer Modellierungstechniken kann also dazu beitragen, dass bestimmte Fehlerklassen und strukturelle Mängel schneller entdeckt werden können, als dies z. B. auf Quelltextebene möglich ist. Der ursprünglich aus der Architektur stammende Ansatz der Entwurfsmuster hat seit 1995 im Rahmen der objektorientierten Software-Entwicklung weite Verbreitung gefunden (vgl. z. B. [Ga96]). Entwurfsmuster stellen Kombinationen von Objekten und Klassen als Lösungen für wiederkehrende abstrakte Entwurfsprobleme bereit und fördern somit das Lernen aus Beispielen. Durch die Verwendung erprobter Lösungen für Teilprobleme wird die Qualität der Modelle, die Muster verwenden, in der Regel verbessert.

4 Fachdidaktische Konzepte zum objektorientierten Modellieren

4.1 Wissen strukturieren

Da im Informatikunterricht die Entwicklung von Kompetenzen im Vordergrund steht, ist die sachlogische Struktur der Fachwissenschaft allein ungeeignet zur Strukturierung des Unterrichts. Aufgrund der Struktur der Fachkonzepte wird aber deutlich, welche Elemente als Vorkenntnisse für andere Elemente erforderlich oder hilfreich sind. Verschiedene Fachkonzepte lassen sich mit einer "ist-erforderlich-für-" bzw. einer "ist-hilfreich-für-Relation" verknüpfen. Da es alternative Möglichkeiten gibt, resultiert daraus eine Vielzahl von Varianten, Wissen und Können schrittweise zu entwickeln. Probleme können dabei z. B. auftreten, wenn es in der Struktur zu Sprüngen im Abstraktionsniveau kommt, wie z. B. bei der Konstruktion von Klassenhierarchien zu in Aufgabenstellungen beschriebenen Realitätsausschnitten zu beobachten.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Wissensstrukturen I

- sachlogische Struktur der Konzepte alleine ist nicht ausreichend für den Lehr-Lern-Prozess
- Analyse der Struktur der Fachkonzepte liefert Erkenntnisse darüber, welche Elemente erforderlich oder hilfreich als Vorkenntnisse für andere Elemente sind
- alternative Möglichkeiten führen zu einer Vielzahl von Varianten Wissen und Können schrittweise zu entwickeln
- Ziel: Visualisierung der vielfältigen Zusammenhänge für Lernende und Lehrende
 - bessere Orientierung in den Bildungsinhalten
 - Darstellung von Alternativen
 - Entwurf von sog. Explorationsumgebungen

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 10 / 26

Ein zentrales didaktisches Problem vieler hierzu publizierter Methoden und damit eine große, potentielle Fehlerquelle stellt der Übergang von der objektorientierten Sicht auf einen Realitätsausschnitt hin zur klassenorientierten Sicht des Klassendiagramms dar. In einem Realitätsausschnitt werden zumeist sofort potentielle Klassenkandidaten gesucht, anstatt zunächst problemrelevante Objekte zu identifizieren. Für Fortgeschrittene ist das eine leichte Aufgabe. Für Anfänger wird an dieser Stelle die Formalisierung eines Realitätsausschnitts durch Objekte und damit ein Abstraktionsschritt übersprungen. Um nicht zu frühzeitig zu formalisieren, werden im Informatikunterricht teilweise CRC-Karten [BC89] als Vorstufe zu Klassendiagrammen eingesetzt. Da eine einzelne Karte eine informell beschriebene Klasse repräsentiert, wird dadurch das oben genannte Problem nicht gelöst. Der Abstraktionsschritt von der Beschreibung eines Realitätsausschnitts hin zur Strukturierung von Klassen in einem Klassendiagramm kann vereinfacht werden, wenn Objektdiagramme (s. z. B. [Ru93, S. 29]) verwendet werden (s. Abb. 2).

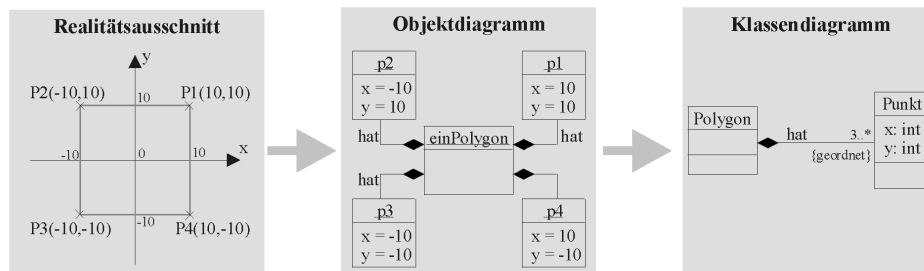


Abbildung 2: Vom Realitätsausschnitt zum Klassendiagramm

Diese stellen eine Momentaufnahme eines Objektgeflechts zu einem bestimmten Zeitpunkt mit den aktuellen Attributwerten und Beziehungen zwischen Objekten dar. Da im Objektdiagramm bereits Notationen verwendet werden, die auch im Klassendiagramm verwendet werden, stellen diese eine gute Vorbereitung dar. Für die Transformation eines Objektdiagramms in ein Klassendiagramm können später Regeln formuliert werden, die diesen Prozess unterstützen. Ein ähnliches Vorgehen wird von Balzert [Ba99, S. 132 u. S. 142] vorgeschlagen. Für jede typische Interaktion eines Benutzers mit einem System (Anwendungsfall) soll ein Objektdiagramm und ein lokales Klassendiagramm konstruiert werden. Die einzelnen Klassendiagramme werden anschließend zusammengeführt. Bei diesem Vorgehen wird ein großer Teil der gestalterischen Kreativität bereits bei der Konstruktion der Objektdiagramme gefordert. Um Brainstorming-Prozesse an dieser Stelle zu fördern, können die Objektdiagramme zunächst auf einer informellen Ebene entwickelt werden, etwa unter Verwendung eines Objekt-Analogons zu CRC-Karten. Iteriert man das beschriebene Vorgehen für alle Anwendungsfälle eines Informatiksystems, so erhält man ein statisches, objektorientiertes Analysemodell. Bei [Ba99, S. 170ff] finden sich weitere Hinweise, wie ausgehend von Anwendungsfällen und einem statischen Analysemodell über Szenarios eine Folge von Sequenzdiagrammen entwickelt und damit das dynamische Modell schrittweise und systematisch konstruiert werden kann.

Da OOM für Lehrende ein neues Basiskonzept ist, soll ein übersichtlicher und kompakter Überblick in grafischer Form darüber gegeben werden, in welchen Beziehungen Teile zu einander stehen, in welcher Reihenfolge sie sinnvoll angeeignet werden können und wo aufgrund des Vorwissens von Lernenden fortgesetzt werden kann, um so eine bessere Orientierung im Lernstoff zu ermöglichen. Lernenden kann die grafische Struktur für die Organisation von Selbststudienphasen bzw. zur Wiederholung und Nachbereitung von Unterricht dienlich sein.

Wissensstrukturen II

Arten von Wissensstrukturen im didaktischen System:

• für Lernende und Lehrende:

- Entwicklung von Kompetenzen
Definition von Etappen und wesentlichen Punkten im Bildungsprozess
- Entwicklung von Heuristiken zu den Bildungsinhalten
(z.B. Substantive in Problembeschreibungen sind Klassenkandidaten)

• für Lehrende:

- Strukturierung von Lehrheuristiken
(z.B. erfolgreiche Sequenzierung von Bildungsinhalten)

An Darstellungsformen für diesen Zweck ergeben sich eine Reihe von Anforderungen. Insbesondere müssen sie ausdrucksstark, selbsterklärend und leicht verständlich sein. Weiterhin müssen sie es von ihrer Topologie her ermöglichen, nicht nur einen festen Lehr-Lern-Pfad, sondern ein hohes Maß an Flexibilität für individuelle Lehr-Lern-Pfade zu eröffnen. Sequentielle Darstellungsformen, wie Listen, sind dazu ungeeignet. Baumbasierte Darstellungsformen mit beliebigem und variablem Knotenausgrad ermöglichen es, Fachkonzepte hierarchisch anzuordnen und so die Anforderungen an die Vorkenntnisse darzustellen, indem diese jeweils als Kinder eines Knotens dargestellt werden. Da alle Kinderknoten gleichberechtigt sind, bleibt die Reihenfolge der Aneignung der mit ihnen verbundenen Fachkonzepte offen. Reine Baumdarstellungen des Vorkenntnisgeflechts erweisen sich als nachteilig, wenn Fachkonzepte dieselben Vorkenntnisse benötigen. Der, die gemeinsamen Vorkenntnisse repräsentierende, Teilbaum wird dann so oft in den Gesamtbaum übernommen, wie es Elemente gibt, die diese Vorkenntnisse erfordern. Dadurch wird die Darstellung schnell unhandhabbar. Zyklenfreie, gerichtete Graphen vermeiden dieses Problem und bieten ferner den Vorteil, verschiedene Relationen zwischen Knoten darzustellen. Fachkonzepte können darin durch eine "ist-erforderlich-für-" oder eine "ist-hilfreich-für-Relation" strukturiert werden. Weiterhin muss die Darstellungsform hinreichend ausdrucksstark sein, um z. B. boolesche Verknüpfungen zwischen über Relationen verbundenen Knoten zu realisieren. Es muss sich ausdrücken lassen, dass verschiedene Elemente gemeinsam als Vorkenntnisse für andere benötigt werden oder dass aus einer Menge von Elementen wenigstens eines als Vorkenntnis benötigt wird. Bei alledem bietet die Verwendung von standardisierten Darstellungsformen in der Regel den Vorteil, dass bereits Editoren existieren, auf die zurückgegriffen werden kann.

Unter Berücksichtigung der gegebenen Anforderungen erweisen sich folgende, standardisierten Darstellungsformen für den gegebenen Zweck prinzipiell als geeignet: Und-Oder-Bäume, Begriffsnetze (concept maps) und semantische Netze. Begriffsnetze ermöglichen zwar beliebige Relationen, die boolesche Verknüpfung ist aber nicht darstellbar. Semantische Netze eröffnen den größten Darstellungsspielraum, allerdings stark zu Lasten der Übersichtlichkeit. Die Erweiterung von Und-Oder-Bäumen zu azyklischen Und-Oder-Graphen stellt den besten Kompromiss bzgl. der gegebenen Anforderungen dar (vgl. [Br00a], [BS01]). Lerninhalte von

OOM wurden mit dieser Darstellungsform erfolgreich vom Autor strukturiert (vgl. Beispiel in Abb. 3).

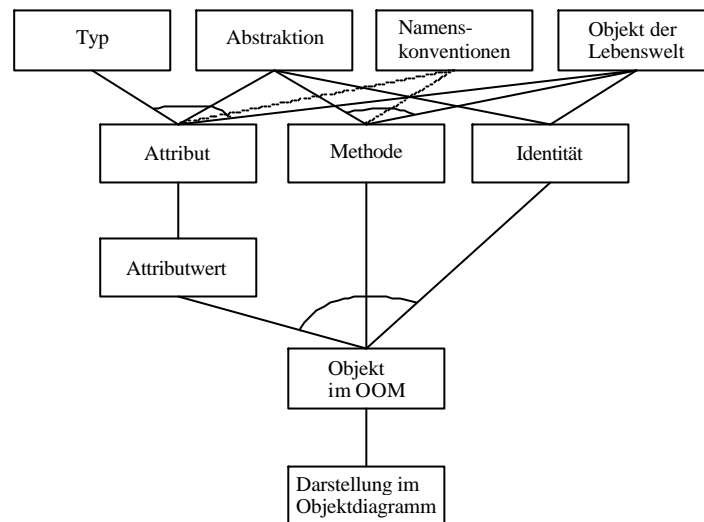


Abbildung 3: Strukturierung und Repräsentation von Fachkonzepten von OOM im Und-Oder-Graph



4.2 Aufgabenklassen

Die inhaltliche Schwerpunktverschiebung vom Programmieren hin zum Modellieren muss im gesamten Informatikunterricht umgesetzt werden. Damit werden neue Aufgabenklassen erforderlich, anhand derer sich die neuen Fachkonzepte erlernen lassen. In der Fachdidaktik findet sich ein großer Reichtum an publizierten Programmieraufgaben und -lösungen, Beispiele zur Modellierung sind noch selten. In Lehrbüchern zu OOM findet man einen reichen Vorrat an Aufgabenstellungen zur objektorientierten Modellierung, die wegen ihres einführenden Charakters auch für den Informatikunterricht herangezogen werden können (z. B.

[Ru93], [Ba99]). Da es sich bei den Adressaten dieser Lehrbücher aber nicht um Schülerinnen und Schüler handelt, werden Kriterien formuliert, anhand derer Aufgabenstellungen für den Informatikunterricht ausgewählt bzw. transformiert werden können (vgl. auch [Br00b]):

- **Fachkonzepte:** Es werden nur diejenigen Aufgaben ausgewählt, die sich auf die für den Informatikunterricht ausgewählten Fachkonzepte beziehen.
- **Betonung der Modellierung:** Da es hier um neue Aufgabenklassen zu OOM geht, ist die Betonung der Modellierung zentral.
- **Sprachenunabhängigkeit:** Die Formulierung der Aufgabe soll so gewählt sein, dass keine spezielle Modellierungs- oder Programmiersprache zur Bearbeitung erforderlich ist. Damit wird gewährleistet, dass die Aufgabe an die im Unterricht verwendeten Sprachen angepasst werden kann.
- **Komplexität:** Es werden Aufgaben sehr unterschiedlicher Komplexität benötigt, um sowohl einzelne Modellierungsschritte als auch die selbstständige Konstruktion komplexer Modelle erlernen zu können. Zu komplexe Aufgabenstellungen führen zur Überforderung und binden zu viel Unterrichtszeit. Dadurch verursachte Misserfolgserlebnisse führen meist zum Verlust der Motivation bei Lernenden.

Analysiert man Übungsaufgaben zu OOM, so lassen sich diese i. d. R. in verschiedene Arbeitsaufträge und einen Beispielkontext trennen. Klassen neuer Übungsaufgaben lassen sich dadurch identifizieren, dass die nach den zuvor genannten Kriterien ausgewählten Aufgabenstellungen von ihren jeweiligen Beispielkontexten getrennt und somit zu strukturierten "Aufgabengerüsten" reduziert werden.



Für jede Aufgabenklasse wird angegeben, welche Materialien zur Verfügung stehen bzw. welche Information gegeben ist und worin der Auftrag besteht bzw. welche Information gesucht ist, wie z. B. in folgender Aufgabenklasse:

Gegeben: Liste von Klassen-, Attribut- und Operationsnamen mit kurzer Beschreibung

Gesucht: Zuordnung von Attributen und Operationen zu Klassen

Weitere Beispiele:

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Beispiele für Aufgabenklassen

- **Merkmal zuordnen**
 - ! Liste von Klassen-, Attribut- und Operationsnamen mit kurzer Beschreibung
 - ? Zuordnen von Attributen und Operationen zu Klassen
- **Merkmal analysieren**
 - ! sehr große Objektgruppe, Anwendungszweck
 - ? identifizierende(s) Attribut(e) für gegebenen Anwendungszweck angeben
- **Modell beschreiben**
 - ! Objekt- oder Klassendiagramm
 - ? Beschreibung mit eigenen Worten
- **Modell analysieren**
 - ! Liste von Klassennamen aus der Lebenswelt
 - ? diskutieren, was Begriffe gemeinsam haben; Liste um weitere Klassennamen ergänzen

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 19 / 26

Eine Sequenz n unabhängiger Teilaufgaben zum selben Beispielkontext führt zu n verschiedenen Aufgabenklassen. Durch n aufeinander aufbauende Teilaufgaben zum selben Beispielkontext wird der Lösungsweg einer komplexeren Aufgabenstellung vorstrukturiert. Jede dieser Teilaufgaben kann als eigene Aufgabenklasse aufgefasst werden, für die dann das zu ihrer Bearbeitung erforderliche Wissen entweder direkt im Aufgabentext bereitgestellt oder durch vorgelagerte Aufgabenstellungen erarbeitet werden muss. Durch Kombination dieser elementaren Aufgabenklassen lassen sich komplexere Aufgabenklassen konstruieren, bspw. dadurch, dass für den Lösungsweg erforderliche Zwischenergebnisse nicht in eigenen Teilaufgaben erarbeitet werden, sondern durch den Lernenden selbst gefunden werden müssen.

Die Aufgabenklassen werden so strukturiert und in einer Baumstruktur angeordnet, dass Aufgabenklassen, die sich auf eine spezielle Modellierungstechnik oder ein spezielles Modelldetail beziehen, die Blätter bilden. Innere Knoten bilden Aufgabenklassen, die verschiedene Aspekte kombinieren. Diese haben zugleich auch Sicherungscharakter für in der Hierarchie tiefer befindliche Knoten. Die selbst- und vollständige Modellierung eines Informatiksystems bildet in dieser Struktur den Wurzelknoten. In [Br00b] wurden Aufgabenstellungen zur Konstruktion eines statischen Systemmodells auf die beschriebene Weise analysiert und dokumentiert und dabei die in Abb. 4 dargestellte Hierarchie von Aufgabenklassen konstruiert. Ziel dieser Strukturierung ist es nicht, kreative Prozesse der Unterrichtsgestaltung durch schematische Darstellungen einzuengen. Vielmehr soll dazu beigetragen werden, geeignete Fachkonzepte fachdidaktisch leichter zugänglich zu machen, ihre Verbreitung zu fördern und damit Informatikunterricht zu verbessern.

Aufgabenklassen

Aufgabenklassen als abstrakte Aufgaben kombiniert mit Lösungsstrategien

Gewinn für Lernende:

- Aufgaben nach Kriterien klassifizieren, um die Lücke zwischen kognitiven Problem und einer Lösungsidee zu überbrücken
- passendes Strukturkonzept für eine Aufgabe auswählen
- Daten aus einer Problembeschreibung extrahieren und mit einem Strukturkonzept verknüpfen

Gewinn für Lehrende:

- Materialien und Methodik, um Übungsaufgaben für den Bildungsprozess zu konstruieren
- Methodik um eine Folge von Übungen mit wachsender Schwierigkeit und Komplexität zu konstruieren

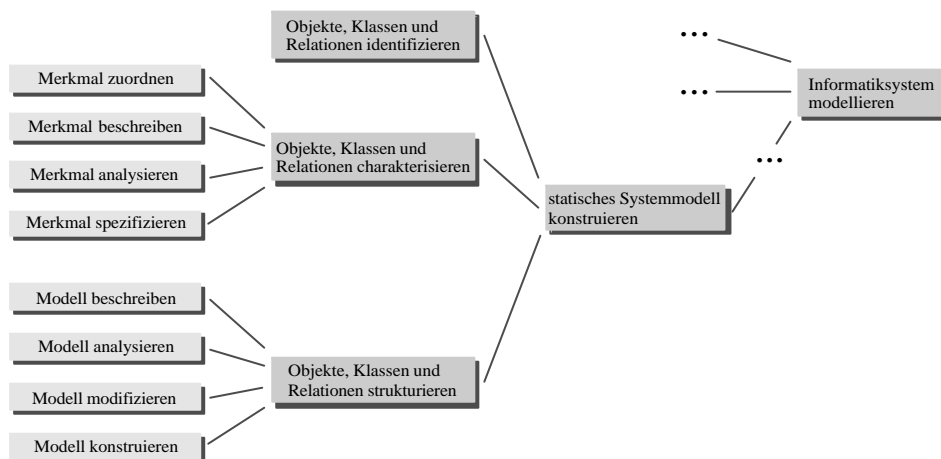


Abbildung 4: Aufgabenklassen bei der Konstruktion eines statischen Systemmodells

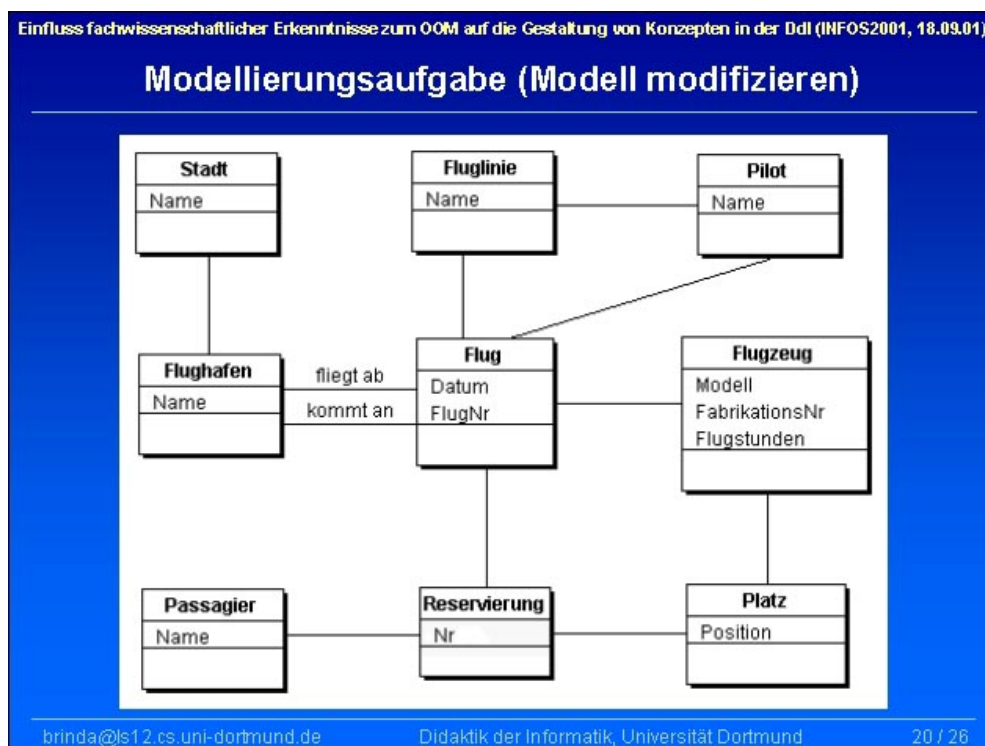
Um Übungsaufgaben für Informatikunterricht zu OOM zu konstruieren, sind neben abstrakten Aufgabenklassen auch konkrete und geeignete, motivierende Beispielkontexte erforderlich. Dieser Beispielkontext liefert dann den inhaltlichen Rahmen für die jeweilige Aufgabenstellung. Aufgrund der Analyse solcher Beispielkontexte wurden folgende Kriterien für die Auswahl von Beispielkontexten abgeleitet:

- **Eignung für OOM:** Nicht jeder Beispielkontext legt ein objektorientiertes Vorgehen gleichermaßen nahe (s. z. B. [Fü99]). Keinesfalls sollte dies durch Lehrende erzwungen werden. Eine Ausnahme hiervon kann lediglich ein Vergleich der Eignung verschiedener Programmierparadigmen zur Lösung derselben Problemstellung sein.
- **Lebensweltbezug:** Der Beispielkontext sollte aus der Lebenswelt der Lernenden stammen. Diese sollten die vielfältigen Zusammenhänge des Realitätsausschnittes aufgrund eigener Erfahrung kennen oder hinreichend viel Information darüber recherchieren können. Ein unbekannter Kontext erfordert zunächst eine zeitaufwendige Aus-

einandersetzung damit. Die Aufmerksamkeit kann dann nicht auf das Ziel, Modellierung zu erlernen, konzentriert werden.

- **Motivation:** Ein Beispielkontext muss motivierend sein, um die Aufmerksamkeit der Lernenden zu binden und ihr Interesse für weitere Beschäftigung mit den Inhalten zu wecken. Dies kann z. B. durch Kontexte geschehen, die den Lernenden aufgrund eigener Interessen Freude bereiten, oder die ihnen einen gewissen Nutzen für sich oder ihr Umfeld versprechen. Von besonderer Bedeutung sind Beispielkontexte zum Erlernen neuer Inhalte. In Verbindung mit den Aufgabengerüsten sollte ein solcher Beispielkontext Spannung aufbauen und aufrecht erhalten. Der Reiz des Neuen soll dazu genutzt werden, die Beschäftigung mit einer Problemstellung zu motivieren.
- **Leichte Änderbarkeit und Erweiterbarkeit:** Strukturierungstechniken, wie Klassenhierarchien, abstrakte Klassen, etc. zeigen ihre Qualität erst, wenn bestehende Strukturen verändert bzw. erweitert werden. Das setzt einen entsprechend offenen Beispielkontext voraus, in dem Erweiterungen und Modifikationen der Struktur möglich und sinnvoll sind. In diesem Zusammenhang kann zwischen größeren Projekten und einer Verkettung kleinerer Beispiele unterschieden werden. Ein größeres Projekt kann von vornherein so gewählt werden, dass die Anwendung der o. g. Strukturierungstechniken Vorteile bringt. Die Erstellung einer Gesamtlösung kann aber je nach Projektgröße sehr zeitintensiv sein und damit zum Motivationsverlust bei den Lernenden führen. Bei einer Verkettung aufeinander aufbauender kleinerer Beispiele kann das Ende flexibler gewählt werden. Ferner können bspw. verschiedene Klassenstrukturen erstellt und modifiziert werden und die Qualität der Techniken dadurch bewertet werden. Erweiterbare Kontexte, die auch Verknüpfungen fachlicher Zusammenhänge ermöglichen, fördern eine angemessene Binnendifferenzierung.

Einen Ausschnitt aus einer Modellierungsaufgabe zeigt diese Folie:



Das Szenario (auf dem hier gezeigten Niveau) ist den Lernenden i.d.R. bekannt. Mögliche Aufgaben sind hier, je nach Bildungsstand:

- Beschreibung des Diagramms mit eigenen Worten,
- Zuordnung von gegebenen Attributen und Methoden zu den Klassen (Unterscheidung nach einfacher und mehrfacher Zuordnung möglich),
- Benennung der Relationen und Spezifikation der Multiplizitäten,
- inhaltliche Modifikation / Ergänzung des Klassendiagramms,
- Restrukturierung des Klassendiagramms unter Verwendung von Vererbungsbeziehungen,
- Konstruktion eines Objektdiagramms zu einer textuell beschriebenen Situation,
- ...

4.3 Einsatzszenario

Die Repräsentation der Wissensstrukturen dient dazu, Lernenden und Lehrenden eine bessere Orientierung bei der Kompetenzentwicklung zu ermöglichen, indem gezeigt wird, wie einfache Wissens Elemente zur Kompetenz eines komplexeren Konzeptverständnisses verbunden werden. Es wird gezeigt, welche einfachen Konzepte von Lernenden verstanden werden müssen, welches Vorwissen sie haben müssen, um ein komplexeres Konzept zu durchdringen. Es handelt sich bei dieser Repräsentation um ein mögliches Modell des angestrebten fachlichen Lernfortschrittes, das schrittweise aufgrund von Unterrichtserfahrungen um Kompetenzbilder anzureichern ist. Dadurch gelingt es, den Lehr-Lern-Prozess besser am Vorwissen der Lerngruppe zu orientieren. Die Anwendung von höherwertigen Konzepten oder Methoden im Sinne vorwegnehmenden Lernens soll hierdurch keineswegs in Frage gestellt werden - zum Durchdringen des komplexeren Konzeptes ist aber das Vorwissen erforderlich, das für die Anwendung u. U. keine Rolle spielt.. Die Strukturen können zur Gewinnung von Kriterien für Lernerfolgskontrollen eingesetzt werden, da ersichtlich wird, welches die zentralen Bestandteile eines Konzeptes sind, die ein Lernender im Rahmen einer Prüfungssituation diskutieren und anwenden können muss. Sie dienen damit Lehrenden und Lernenden zur Evaluation des fachlichen Lernfortschrittes. Weiterhin können die Wissensstrukturen von Lehrenden dazu verwendet werden, Empfehlungen für die Sequenzierung von Kursreihen zu erhalten, indem die Auswahl von Übungsaufgaben dem in den Wissensstrukturen vorgeschlagenen Aufbau der Kompetenzen folgt. Für die Ausgestaltung kommunikativer Lehr-Lern-Situationen steht mit den vorgeschlagenen Aufgaben- und Kontextklassen eine fachdidaktische Vorgehensweise zur Verfügung, mit der sich vielfältige, unterschiedlich komplexe Problemstellungen für den Informatikunterricht zu OOM einfach durch Lehrende konstruieren lassen. Dabei ist es immer möglich, die vorgeschlagenen Aufgabenklassen durch Variantenbildung, Vereinfachung oder Kombination so zu modifizieren, dass sie für die gegebene Unterrichtssituation geeignet sind und dazu beitragen, vorhandene Lernbarrieren zu überwinden.

5 Software-Entwicklungswerkzeuge als Lernhilfen?

Informatikunterricht der Vergangenheit setzte auf die Entwicklung von kleinen Programmen, um daraus Erkenntnisse über Anwendungen in der Lebenswelt abzuleiten. Das funktionierte, solange beiden, den Unterrichtsbeispielen und den professionellen Lösungen, die gleichen Prinzipien und Methoden zugrunde lagen. Lernende handelten nach der gleichen Strategie wie Experten, allerdings mit Gegenständen von stark reduzierter Komplexität. Inzwischen sind Algorithmen, Datenstrukturen und Methoden der Software-Entwicklung, die großen Informationssysteme zugrunde liegen, prinzipiell andere, so dass der Transfer vom "Programmieren im Kleinen" auf das "Programmieren im Großen" [Ap98] nicht empfohlen wird.

Die Tätigkeiten der Lernenden im Informatikunterricht reduzieren sich häufig auf Anwenden von Standardsoftware und Problemlösen in einer Programmiersprache. Aufgrund veränderter

Bildungsziele werden objektorientierte Programmiersprachen erlernt. Lernende bevorzugen die Sprache Java wegen ihrer Bedeutung im professionellen Bereich. Diesem Wunsch geben Lehrende nach. Die Modellierungssprache "Unified Modeling Language (UML)" und die damit verbundene Werkzeuge für Analyse und Entwurf kommen im Informatikunterricht zum Einsatz. Diese Werkzeuge werden als wenig geeignet für das informatische Modellieren im Informatikunterricht bewertet, da die Komplexität ihrer Benutzungsoberflächen viel Unterrichtszeit bindet. Das ist aber lediglich ein Symptom der originären Zielsetzung von Software-Entwicklungswerkzeugen, nämlich professionelle Software-Entwickler bei der effizienten und möglichst fehlerfreien Erstellung von Software-Produkten zu unterstützen.

Software-Entwicklungswerkzeuge müssen Funktionalitäten bereitstellen, die sich im Erkenntnisprozess von Lernenden als Barrieren erweisen, z. B. Fehlerbegrenzung durch Handlungsverbot. Die für die Vermeidung bestimmter Fehlerklassen sehr sinnvolle Fehlerbegrenzung durch Handlungsverbot (z.B. Auswahl des Rückgabetyps einer Methode aus einer Liste von Datentypen) ist als ständig präsente Funktionalität aus fachdidaktischer Sicht kritisch zu bewerten, da Lernende auf diese Weise vom System vor bestimmten Fehlerklassen bewahrt werden und ein Lernen aus diesen Fehlern somit verhindert wird. Beim Arbeiten ohne ein Software-Entwicklungswerkzeug oder mit einem, das diese Funktionalität nicht bietet, wird der Anwender mit solchen Fehlern konfrontiert, ohne eine entsprechende Problemlösungsstrategie erlernt zu haben. Aus Gründen der Gestaltungsflexibilität bieten viele Systeme die Möglichkeit, vorbereitete Auswahlfelder um eigene Freitexteingaben zu ergänzen. Damit wird dieser Punkt zum Teil relativiert.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der Ddi (INFOS2001, 18.09.01)

Bedeutung von Software-Entwicklungsumgebungen

- Erkenntnisbarrieren**
 - Basiskonzepte werden vorausgesetzt
 - Benutzer wird zu bestimmten Eingaben gezwungen (Sicherung der Vollständigkeit)
 - Benutzer darf (oft) nur aus korrekten Alternativen auswählen (Fehlerbegrenzung durch Handlungsverbot)
- Erkenntnisfördernde Funktionalitäten**
 - Verletzungen der Syntax der Modellierungssprache identifizieren
 - logische Fehler identifizieren (z. B. Vererbungszyklen)
 - Vollständigkeit und Konsistenz von Modellen sichern
 - Hinweis auf fehlende Teile bei Modellelementen (z. B. Konstruktor bei Klasse)
 - Übersetzung eines Modells in Quellcode prüfen
 - Perspektivenwechsel (z. B. Diagramme vergrößern, Details ausblenden)
 - Exploration der Systembedienung wird teilweise unterstützt

brinda@le12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 22 / 26


Im Hinblick auf den Lehr-Lern-Prozess besteht das Hauptproblem darin, dass die fundamentalen Ideen des Arbeitsgegenstandes, hier also des Erstellens objektorientierter Modelle, vorausgesetzt werden. Diese Systeme richten sich an Anwender mit Vorwissen zur Gestaltung objektorientierter Problemlösungen und helfen diesen, alles Wesentliche bei der Modellierung zu berücksichtigen, bestimmte Routineaufgaben zu automatisieren und die Qualität der Lösung, wo möglich, zu erhöhen. Für die Aneignung von Modellierungskonzepten sind diese Systeme nicht konzipiert und nicht geeignet. Produktentwicklung und die Ausbildung von

Software-Entwicklern sind keine Ziele des Informatikunterrichts. Es ist ein anderes Ziel, Lernende im Erkenntnisprozess zu unterstützen. In der Fachgruppe "Didaktik der Informatik" entwickelt eine studentische Projektgruppe im WS 2001/02 und im SS 2002 unter Mitbetreuung des Autors eine Lernumgebung für objektorientiertes Modellieren im Informatikunterricht, die Lernenden einen handlungsorientierten, explorativen Zugang zum objektorientierten Modellieren eröffnen soll.

Einfluss fachwissenschaftlicher Erkenntnisse zum OOM auf die Gestaltung von Konzepten in der DdI (INFOS2001, 18.09.01)

Explorationsumgebung

- **Lerner helfen, die kognitive Lücke zwischen einer Sammlung von Diagrammen und den Prozessen in einem System zu überbrücken**
 - multiple, synchronisierte Sichten
 - Werkzeuge für Sichtentransfer (z.B. Objekt- und Klassendiagramm), Prozesse aufnehmen und analysieren
 - Lernen durch Entdecken und geführte Exploration
- **Entwicklung von Kompetenzen (z.B. Lernen aus Beispielen und aus Fehlern, Selbstreflexion)**
- **Konzentration auf das Wesentliche (OOM anstatt Systembedienung)**
- **Prototyp: „Vom Realitätsausschnitt zum Objektdiagramm“**

hier  starten

brinda@ls12.cs.uni-dortmund.de Didaktik der Informatik, Universität Dortmund 24 / 26

6 Schlussfolgerungen und Ausblick

Im Rahmen dieser Arbeit wurde begründet, dass die unterrichtliche Behandlung von OOM als Problemlösungsmethode dazu geeignet ist, Ziele allgemein bildenden Informatikunterrichts umzusetzen. Ferner wurde gezeigt, dass fachwissenschaftliche Erkenntnisse zum OOM dazu genutzt werden können, den Mangel an Lehr-Lern-Materialien in der Fachdidaktik zu beseitigen. Im Weiteren werden diese Arbeiten nun verfeinert. Schwerpunkte werden dabei Besonderheiten von Modellierungstechniken und Vorgehensweisen, sowie die bessere Berücksichtigung von Alternativen bei der Strukturierung der Fachkonzepte einerseits und die Erweiterung der Aufgabenklassen um dynamische Aspekte und deren Verzahnung mit den statischen andererseits sein. Die Entwicklungsstufen des Konzeptes werden prozessbegleitend erprobt und evaluiert ab 2001, um parallel dazu sowohl Konzept als auch Entwurfsmethodik verbessern zu können.

Literaturverzeichnis

- [Ap98] Appelrath, H.-J.; Boles, D.; Claus, V.; Wegener, I.: Starthilfe Informatik. B.G. Teubner, Stuttgart, 1998.
- [Ba99] Balzert, H.: Lehrbuch der Objektmodellierung. Spektrum Akademischer Verlag, Heidelberg, 1999.
- [BC89] Beck, K.; Cunningham, H.: A laboratory for teaching object-oriented thinking. In: Proceedings of OOPSLA 1989, SIGPLAN notices (ACM) vol. 24, New Orleans, 10/1989.
- [Bo94] Booch, G.: Objektorientierte Analyse und Design. Addison-Wesley, Bonn, 1994.
- [BRJ98] Booch, G.; Rumbaugh, J.; Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading, Massachusetts, 1998.
- [Br00a] Brinda, T.: Didaktische Systeme für objektorientiertes Modellieren (OOM) im Informatikunterricht. In (Gesellschaft für Informatik Hrsg.): Informatiktage 2000. Konradin, Leinfelden-Echterdingen, 2000, S. 282-285.
- [Br00b] Brinda, T.: Sammlung und Strukturierung von Übungsaufgaben zum objektorientierten Modellieren im Informatikunterricht. In: Log In 20 (2000) 5, S. 39-49.
- [Br01] Brinda, T.: Einfluss fachwissenschaftlicher Erkenntnisse zum objektorientierten Modellieren auf die Gestaltung von Konzepten in der Didaktik der Informatik. In: Magenheimer, J.; Keil-Slawik, R. (Hrsg.): Informatikunterricht und Medienbildung. Köllen, Bonn, 2001.
- [BS01] Brinda, T.; Schubert, S.: Didaktisches System für objektorientiertes Modellieren. Forschungsbericht Nr. 752, Fachbereich Informatik, Universität Dortmund, 2001.
- [FS97] Fowler, M.; Scott, K.: UML distilled. Applying the standard object modeling language. Addison Wesley Longman, Inc., 1997.
- [Fü99] Füller, K.: Objektorientiertes Programmieren in der Schulpraxis. In (Schwill, A. Hrsg.): Informatik und Schule. Fachspezifische und fachübergreifende didaktische Konzepte. Springer, Berlin, 1999, S. 190-201.
- [Ga96] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Entwurfsmuster. Addison-Wesley, Bonn, 1996.
- [GI00] Gesellschaft für Informatik e.V. (Hrsg.): Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. Beilage zu LOG IN 20 (2000) 2, S. I-VII.
- [Hu00] Hubwieser, P.: Didaktik der Informatik. Springer, Berlin, 2000.
- [Ja93] Jacobson, I.; Christerson, M.; Jonsson, P.; Övergaard, G.: Object-Oriented Software Engineering. A Use Case Driven Approach. Addison-Wesley Longman, New York, 1993.
- [MSH99] Magenheimer, J.; Schulte, C.; Hampel, T.: Dekonstruktion von Informatiksystemen als Unterrichtsmethode - Zugang zu objektorientierten Sichtweisen im Informatikunterricht. In (Schwill, A. Hrsg.): Informatik und Schule. Fachspezifische und fachübergreifende didaktische Konzepte. Springer, Berlin, 1999, S.

149-164.

- [Me97]** Meyer, B.: Object-oriented software construction. Prentice-Hall, New Jersey, 1997.
- [Ru93]** Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorensen, W.: Objektorientiertes Modellieren und Entwerfen. Hanser, München, 1993.
- [Sc93]** Schwill, A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25 (1993) 1, S. 20-31.
- [St99]** Steinkamp, D.: Informatik-Experimente im Schullabor. Diplomarbeit, Fachbereich Informatik, Universität Dortmund, 1999.
- [WWW90]** Wirfs-Brock, R.; Wilkerson, B.; Wiener, L.: Designing Object-Oriented Software. Prentice-Hall, New Jersey, 1990.