Einfluss fachwissenschaftlicher Erkenntnisse zum objektorientierten Modellieren auf die Gestaltung von Konzepten in der Didaktik der Informatik

Torsten Brinda

Didaktik der Informatik
Universität Dortmund
D-44221 Dortmund
brinda@ls12.cs.uni-dortmund.de

Zusammenfassung: In 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.

1 Motivation

In der Arbeitsgruppe „Didaktik der Informatik“ der Universität Dortmund 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]).

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.
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 genannte 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 objektorientierten 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. 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.

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),
   Objekt, Klasse, Beziehungen zwischen Objekten und Klassen (Rumbaugh u. a.),
2. 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. $y.g'(x)$.
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.
Aus Sicht der Informatik ist es erforderlich, dass Anfänger nicht nur die prinzipielle, sondern aus ökonomischen Gründen die Konstruktion guter, objektorientierter Modelle erlernen im 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.
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[1] [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[2] (s. z. B. [Ru93, S. 29]) verwendet werden

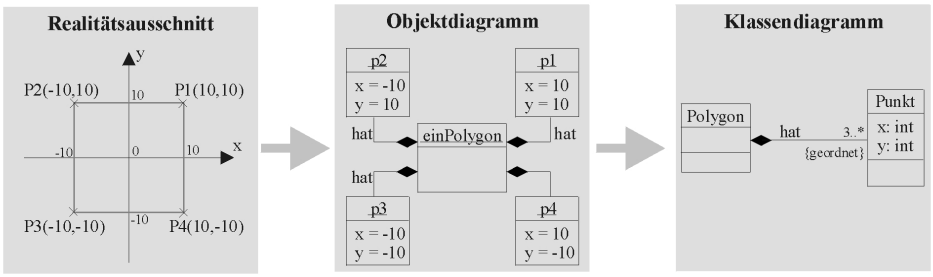

Abb. 1: Vom Realitätsausschnitt zum Klassendiagramm

(s. Abb. 1). 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

[1] Class-Responsibilities-Collaborators (deutsch: Klassen-Verantwortlichkeiten-Beteiligte)
[2] Bei Rumbaugh u. a. werden Objektdiagramme als Instanzendiagramme bezeichnet. Aktuellere Publikationen sprechen allerdings von Objektdiagrammen (s. z. B. [Ba99, S. 19]).

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. 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, standardisierte Darstellungs-formen für den gegebenen Zweck prinzipiell als geeignet: Und-Oder-Bäume, Begriffsnetze (concept maps) und semantische Netze. Begriffsnetze ermöglichen zwar beliebige Rela-

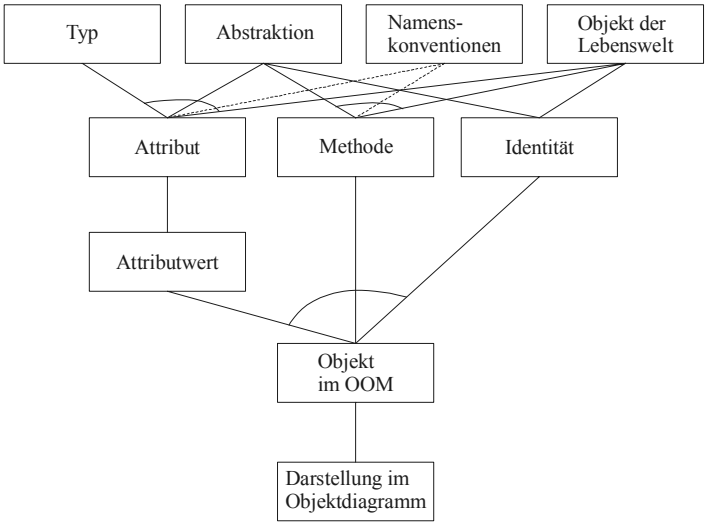

Abb. 2: Strukturierung und Repräsentation von Fachkonzepten von OOM im Und-Oder-Graph

tionen, 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. 2).

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

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. 3 dargestellte

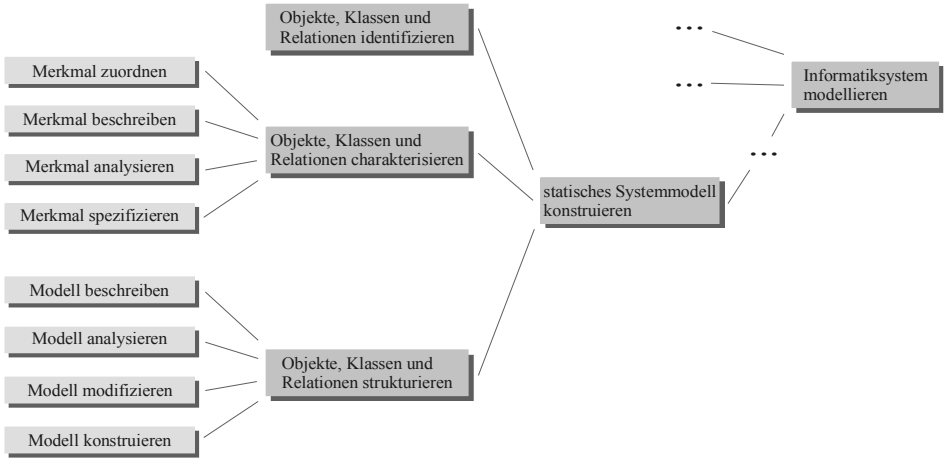

Abb. 3: Aufgabenklassen bei der Konstruktion eines statischen Systemmodells

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.
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 Auseinandersetzung 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 aufrechterhalten. 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.

4.3 Einsatzszenario

Die Informatiklehramtsstudierenden der Universität Dortmund führen immer im Wintersemester ihre schulpraktischen Übungen in der Sekundarstufe II durch. Vorgesehen ist, die Unterrichtsreihe zum objektorientierten Modellieren nach dem Konzept des hier beschriebenen didaktischen Systems zu planen. Die Wissensstrukturen geben den Lehrenden und Lernenden Orientierung für den Unterrichtsprozess. Ziel ist die individuelle Aneignung komplexer Kompetenzen durch Verknüpfung von elementaren Einsichten und Fähigkeiten. Für ein komplexes Konzept sind die erforderlichen Vorkenntnisse im Und-Oder-Graph einfach zu identifizieren, da es dessen Vorgängerknoten sind. Das vorwegnehmende Lernen in Form von Anwendung komplexer Konzepte und Methoden

soll nicht in Frage gestellt werden, denn damit gelingt die Verbindung von kognitionspsychologischen Anforderungen, wie Motivation durch Lebensweltbezug, und der unterrichtspraktischen Notwendigkeit zur Sequenzialisierung des Aneignungsprozesses. Jeder Lernende kann mit dieser Wissensrepräsentation seinen persönlichen Lernfortschritt überprüfen. Die neuen Aufgabenklassen fördern die Gestaltung von Übungen, in denen die Lernenden OOM als Fortsetzung der „Strukturierten Zerlegung“ durch diese neue Entwicklungsmethode kennen lernen. Diesem Ziel wird die enge Bindung an eine Programmiersprache nicht gerecht. Die flexible Verknüpfung von solchen Basiskompetenzen, wie Objekte, Klassen und Relationen identifizieren, charakterisieren und strukturieren, mit dem vielfältigen Unterrichtsthemen bereichert den Lernprozess.

5 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 Gestaltung von Software-Bausteinen zur Visualisierung komplizierter Zusammenhänge und zur Exploration (entdeckendes Lernen) sollen den handlungsorientierten Zugang zu OOM erweitern. Lernende können ihre eigenen Modelle dann mit Simulation überprüfen und sind nicht an den ausschließlichen Weg über die Programmiersprache gebunden. Die Programmierung als wichtige Methode der Informatik soll damit ergänzt, aber nicht ersetzt werden. Die Entwicklungsstufen des Konzeptes werden ab 2001 prozessbegleitend erprobt und evaluiert, um parallel dazu sowohl Konzept als auch Entwurfsmethodik verbessern zu können.

Literaturverzeichnis

[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.

[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] Magenheim, 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.

[WWW90] Wirfs-Brock, R.; Wilkerson, B.; Wiener, L.: Designing Object-Oriented Software. Prentice-Hall, New Jersey, 1990.