

Objektorientiertes Denken als didaktische Basis der Informatik

Cecile K. M. Crutzen
Open University - Technical Sciences
P.O.Box 2960
NL-6401 DL Heerlen
CCR@OUH.NL

Hans-Werner Hein
Verlässliche IT-Systeme
Neumarkt 35-37
D-50667 Köln
hwh@acm.org

Die Prädicate der Erscheinung können dem Objecte selbst beigelegt werden, in Verhältniß auf unseren Sinn, z.B. der Rose die rothe Farbe, oder der Geruch; ... [5]
Immanuel Kant (1787)

Wie wir uns räumliche Gegenstände überhaupt nicht außerhalb des Raumes, zeitliche nicht außerhalb der Zeit denken können, so können wir uns keinen Gegenstand außerhalb der Möglichkeiten seiner Verbindung mit anderen denken. [6]
Ludwig Wittgenstein (1918)

1 Einleitung

Es werden vier mit dem Objektorientierten Denken zusammenhängende didaktische Linien vorgestellt, die sich bei der Entwicklung eines neuen 100-Stunden-Fernkurses "Einführung in die Informatik" [1] für die Open University der Niederlande und bei der Durchführung von Informatikprojekten auf Schülerakademien des Bildung und Begabung e.V. bewährt haben. Anhand von ausgewählten Beispielen wird zuvor formal und methodisch in das objektorientierte Analysieren und Entwerfen eingeführt.

Objektorientiertes Denken ist nicht neu und auch kein Spezialgebiet der Informatik. Es war schon immer da und die obigen Zitate aus der Philosophie sind nur zwei von vielen möglichen. Überlegungen zur Formalisierung objektorientierten Denkens finden sich seit Anfang dieses Jahrhunderts in der Mathematik (Stichwörter: "Typenkalkül", "Abstrakter Datentyp"), waren in der Informatik von deren Beginn an bekannt, und ab ca. 1960 stand SIMULA als erste objektorientierte Programmiersprache zur Verfügung. Dennoch wurde das Selbstbild der Informatik bis Anfang der 80er Jahre weitgehend durch funktionsorientiertes und logikorientiertes Denken bestimmt. Eine ganze Reihe von Trends brachten dann das objektorientierte Denken wieder in den Blick der Kerninformatik, es seien nur zwei erwähnt:

- Die Versuche der Künstlichen Intelligenz, als Basis für verschiedene kognitive Leistungen größere Weltbeschreibungen im Computer objektorientiert anzulegen, zeigten deutliche praktische Vorteile gegenüber anderen Ansätzen.
- Die Anwendung objektorientierten Denkens auf das Thema der Benutzerschnittstellen von persönlichen Computern führte über die Erfindung des Prinzips "Fenster+Menü+Zeiger", zur Programmiersprache SMALLTALK und zum Betriebssystem des Apple Macintosh.

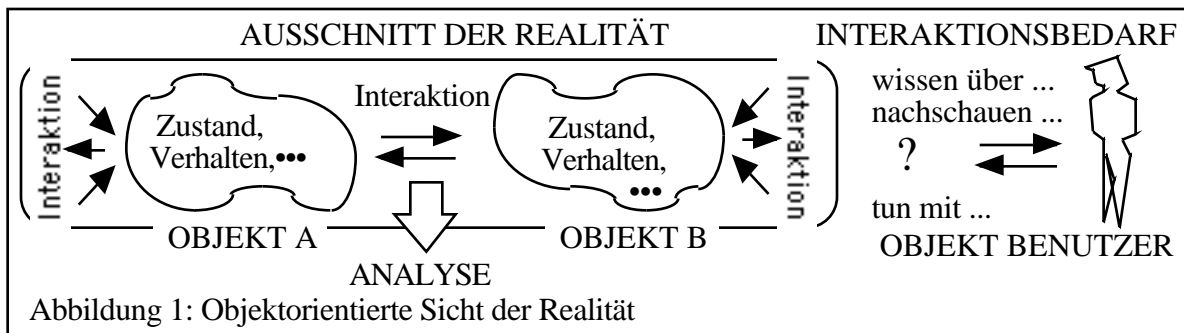
Inzwischen ist Objektorientiertsein in der Informatik ein Muß, wenn auch ein Blick in die Unzahl von Lehr- und Aufklärungsbüchern meist zeigt: "Die Autoren denken nicht objektorientiert und unter einem dünnen, objektorientiert tuenden Terminologie-Lack befindet sich das alte Blech".

Soll man also objektorientiertes Denken in Informatik unterrichten? Wir meinen: Ja, und zwar von Anfang an und zyklisch durch alle Klassenstufen immer wieder. Es braucht dazu keine besonderen Schüler und auch keine besonderen Programmiersprachen oder Computersysteme. Die allgemeine Strategie ist auch nicht exotisch: "Informell beginnen und stufenweise formalisieren". Eine Auswahl von realitätsnahen Beispielen ist einfach. Die Komplexität von

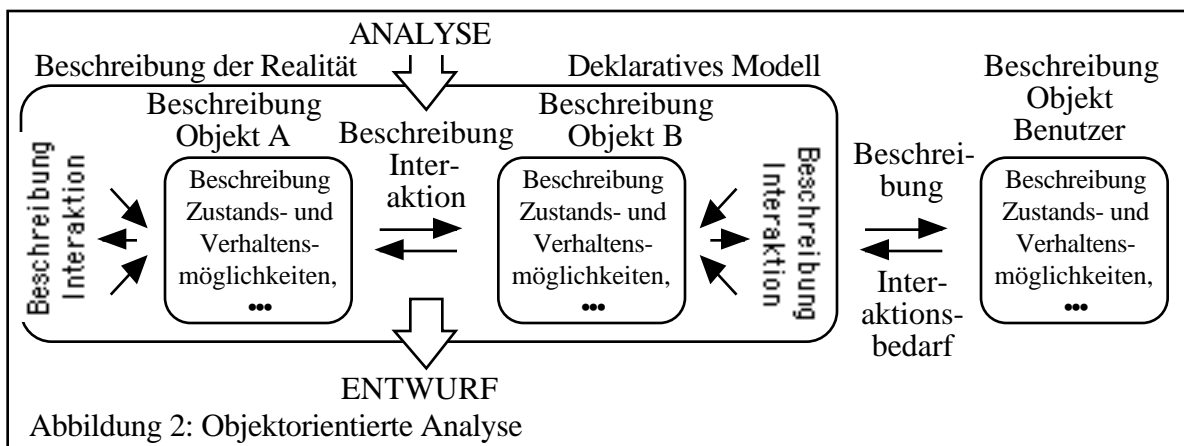
Beispielen und Übungen kann nach der Zahl und Größe der beteiligten Objekte zunehmen. Ein weiterer didaktischer Vorteil objektorientierten Denkens ist, daß das Analysieren, Beschreiben und Entwerfen zyklisch möglich ist. Man kann damit irgendwo anfangen. Behandelte Beispiele sind leicht zu kombinieren, weil man leicht jedes Objekt aus dem Zusammenhang nehmen und in jeden beliebigen anderen Zusammenhang bringen kann. Seitens der Informatik braucht man am Anfang nur wenig Begriffliches. Die allgemeinste Sicht (das Paradigma) ist: "Jedes Objekt der Realität, das Informationen aufnimmt, speichert, verarbeitet und abgibt, kann man als Informationsobjekt beschreiben. Objekte der Realität können untereinander interagieren, diese Interaktionen kann man als Kommunikation zwischen Informationsobjekten beschreiben. Ein Informationssystem ist die Realisierung einer Anzahl von Informationsobjekten - und damit ein Objekt der Realität, das man (hier schließt sich ein Kreis) wiederum als Informationsobjekt beschreiben kann."

2 Was ist "objektorientiert"

Betrachtet man die reale Welt mit der "Objektbrille", so kann man sie analysieren und in einem



Modell beschrieben, das aus Einheiten (Objekten) mit individuellen Zuständen, einem spezifischen Verhalten, sowie weiteren Eigenschaften besteht (Abb.1 und 2). Zwischen diesen Einheiten kann man Interaktionen beobachten, die zeitlich begrenzter oder permanenter Art sind. Aus der objektorientierten Sicht heraus kann man auch versuchen, Informationssysteme zu entwerfen (Abb.3) und anhand des fertigen Entwurfs zu implementieren (Abb.4). Dem Zweck eines geplanten Informationssystems entsprechend, braucht man dafür nur einen Ausschnitt der Realität zu analysieren. Dieser Ausschnitt muß genau die Objekte und Interaktionen enthalten, die direkt oder indirekt für den angestrebten Zweck relevant sind.



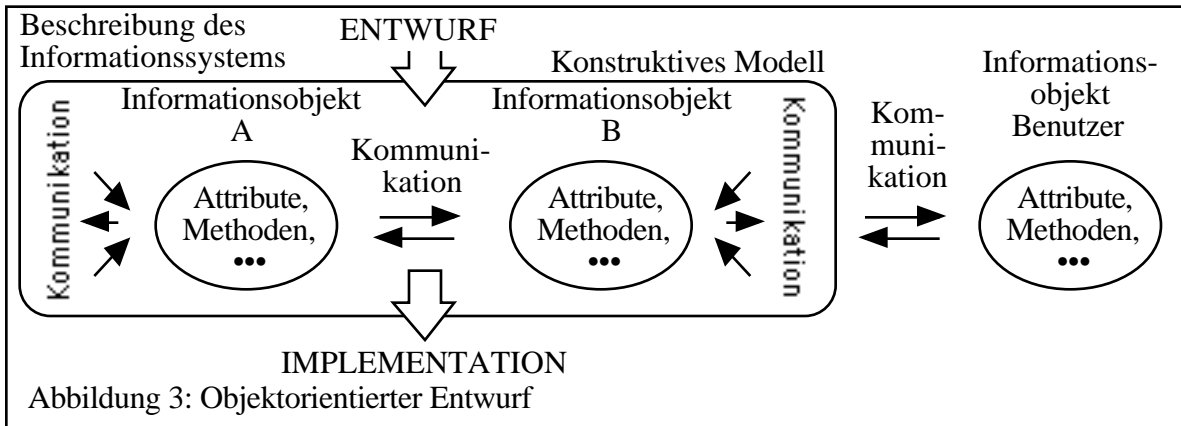


Abbildung 3: Objektorientierter Entwurf

Ein besonderes Objekt der Realität sind die zukünftigen Benutzer. Ihnen sollen ja mittels des geplanten Informationssystems neue synthetische Interaktionen mit Objekten der Realität ermöglicht werden. Daher ist die Analyse und Modellierung ihres Interaktionsbedarfs von Anfang an ein wesentlicher Bestandteil der Entwicklungsarbeit. Objektorientierte Analyse führt zu einer deklarativen Modellierung der Realität (Abb.2), die sich dadurch auszeichnet, daß für jedes reale Objekt und jede reale Interaktion genau ein separater Beschreibungs-Modul angelegt wird. Das steht im Gegensatz zu anderen Analysemethoden, deren Ergebnisse etwa funktional (z.B. Satz von Differentialgleichungen), logisch (z.B. Menge von logischen Aussagen), strukturell (z.B. hierarchisches Modell) usw. orientiert sind. Während der analytische Schritt nur rezeptiv ist und ein zweckentsprechend ausreichend genaues Eins-zu-Eins-Modell der Realität anstrebt, kommen beim Entwurf konstruktive und synthetische Überlegungen hinzu (Abb.3), die zu einer vollständigen Beschreibung des geplanten Informationssystems führen. Für jedes Objekt der Realität wird genau ein Informationsobjekt entworfen, sein Zustand in Attributen und sein Verhalten in Methoden modelliert. Aus konzeptionellen oder technischen Gründen können noch weitere Informationsobjekte hinzukommen, die im analytischen Modell keine Entsprechung haben (siehe z.B. Abb.6). Jede analysierte Interaktionsbeziehung zwischen Objekten wird jetzt in Kommunikation zwischen Informationsobjekten transformiert. Bei dieser Transformation ist mehreres zu beachten:

- Ein Interaktionspfeil zeigt ja die Richtung der Initiative an und nicht unbedingt die Richtung eines Informationsflusses. Daher kann aus einer analysierten einseitigen Interaktion schon einmal eine zweiseitige Kommunikation (ein Dialog) werden (siehe z.B. Abb.5). Das erfordert konstruktives Denken.
- Interaktionen in der Realität ereignen sich in allen sinnlichen Bereichen. Die Kommunikation zwischen Informationsobjekten muß aber rein verbal (symbolisch) entworfen werden (siehe

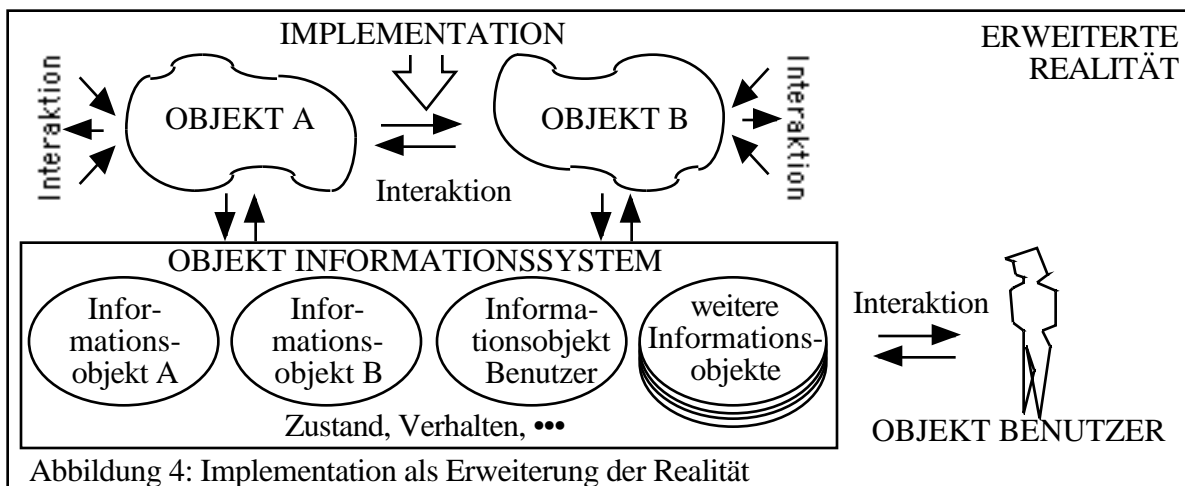


Abbildung 4: Implementation als Erweiterung der Realität

z.B. Abb.5). Das erfordert abstrahierendes Denken.

- Es ist außerdem durchaus möglich, daß aus einer "eigentlich" direkten Interaktion realer Objekte aus technischen Restriktionen heraus eine indirekte Kommunikation der zugehörigen Informationsobjekte wird (siehe z.B. Abb.6). Das erfordert problemlösendes Denken.

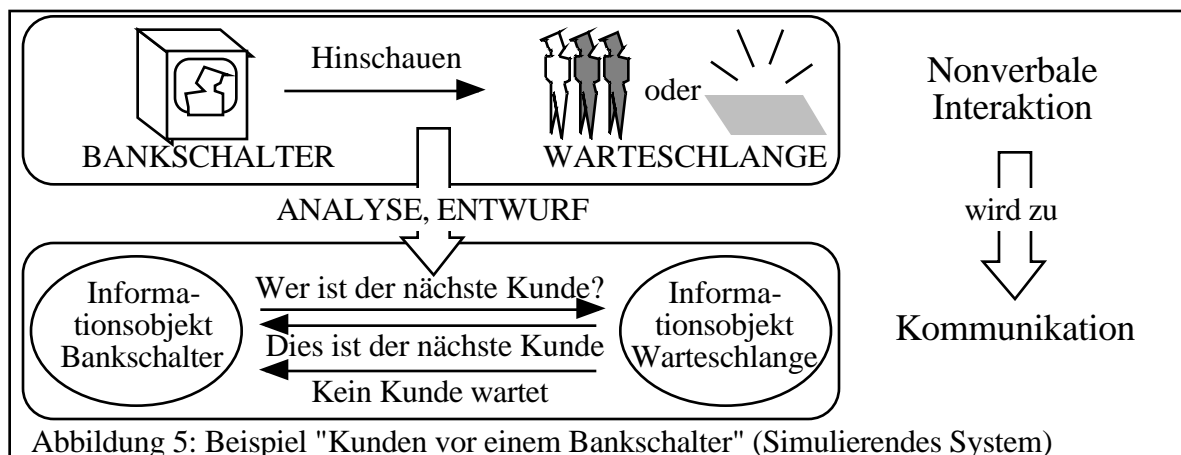
Ist ein Informationssystem fertig entworfen, kann es durch Implementation im wahren Sinne des Wortes "realisiert" werden (Abb.4). Aus der objektorientierten Sicht wird nämlich durch Implementation die bisherige Realität um ein neues Objekt "Informationssystem" und neue Interaktionen zwischen ihm und anderen Objekten erweitert. Diese erweiterte Realität kann man nun wieder analysieren (z.B. um den bisherigen Entwurf des Informationssystems zu verbessern) - objektorientiertes Denken ist grundsätzlich zyklisch angelegt. Bei einer professionellen Informationssystem-Entwicklung gibt es ohnehin noch zyklische Beziehungen zwischen den hier skizzierten Entwicklungsphasen Realitätsausschnitt-Analyse-Entwurf-Implementation. So treten fast zwangsläufig während der Evaluation späterer Phasen Defizite und Unklarheiten in den Modellen zutage, welche die Überarbeitung einer früheren Phase nahelegen.

Eine inkonsequente Betrachtung des Begriffs "Vererbung" im Zusammenhang mit objektorientiertem Denken hat in der Vergangenheit zu Mißverständnissen geführt. Selbst noch im Entwurf (im konstruktiven Modell des geplanten Informationssystems) müssen die Informationsobjekte völlig unabhängig voneinander beschrieben bleiben. Eine Einführung von hierarchischen Abhängigkeiten wäre hier verfrüht, sinnlos und würde nur die Wiederverwendbarkeit in anderen Kontexten gefährden. Erst bei der Implementierung dürfen aus Effizienz-Überlegungen heraus Entscheidungen getroffen werden, ob gleichartige Teilstrukturen in verschiedenen Informationsobjekten in einem gemeinsamen "Ober-Informationsobjekt" realisiert werden, oder nicht. Und analog dazu, ob gleichartige Methoden in einer gemeinsamen "Ober-Prozedur" realisiert werden, oder nicht. Um einen objektorientierten Entwurf zu implementieren, braucht es also nicht notwendigerweise eine Programmiersprache, die datenstrukturelle Vererbung zur Verfügung stellt - wie auch das Hinschreibenkönnen einer Objekthierarchie nicht zwangsläufig objektorientiertes Denken beweist. Für eine Einführung kann man jedenfalls problemlos auf die Betrachtung von Vererbungs-Beispielen verzichten.

3 Informationssysteme Typ 1: Simulierende Systeme

Simulierende Systeme sind am einfachsten zu entwerfen; sie modellieren einen Ausschnitt der Realität ohne synthetische Zugaben. Außer mit ihren Benutzern haben sie keine Interaktionen mit Objekten der Realität. Daher ist es sinnvoll, Einführungen in objektorientiertes Denken mit Simulationsbeispielen zu beginnen. Unser Beispiel in [1] modelliert eine Bankschalterhalle:

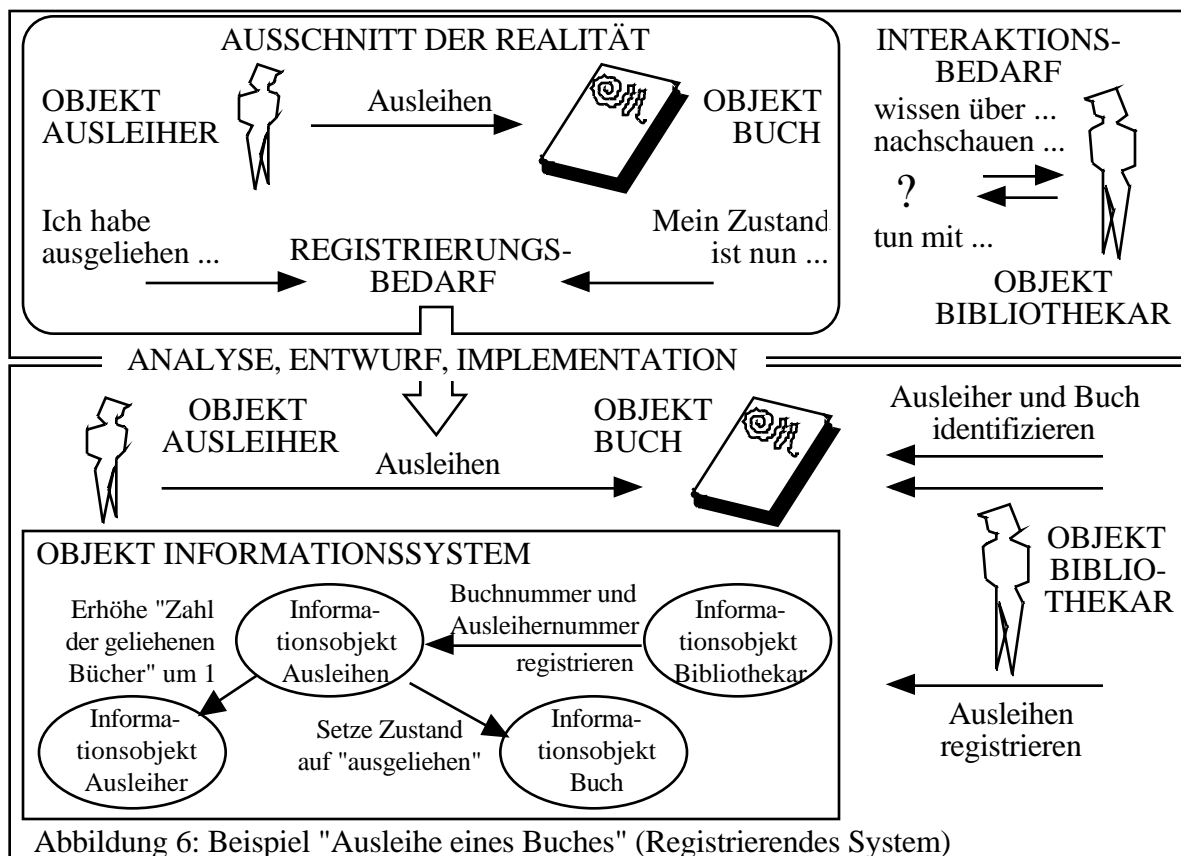
- mit Kunden, die zu verschiedenen Zeiten kommen und verschiedene Wünsche haben,



- mit Schaltern, die zu individuellen Zeiten öffnen und schließen, und für bestimmte Wünsche zuständig sind,
 - mit Warteschlangen vor den Schaltern,
 - mit Abhängigkeiten zwischen den Zuständigkeiten, so daß ein Kunde im Allgemeinen an mehreren Schaltern anstehen muß, bis sein eigentlicher Wunsch abschließend erfüllt wird.
- Zweck des simulierenden Informationssystems sollte es sein, anhand der zeitlichen Verteilung eintreffender Kunden und der statistischen Verteilung ihrer Wünsche, die Zusammenhänge zwischen den Zuständigkeiten der einzelnen Schalter und den maximalen und durchschnittlichen Kunden-Wartezeiten zu studieren. Die Implementation erfolgte zunächst in OBERON unter Macintosh 7.1 durch die Autoren, die Open University portierte das System dann nach TURBO PASCAL 6 unter WINDOWS. Abb.5 zeigt als besonders interessantes Detail dieses Beispiels die Interaktion des Schalterbeamten mit der Warteschlange vor seinem Schalter. Beim Entwurf wird aus der einseitigen nonverbalen Interaktion eine zweiseitige verbale Kommunikation mit drei möglichen Nachrichten (messages).

4 Informationssysteme Typ 2: Registrierende Systeme

Nach einer Betrachtung von Beispielen, die zu simulierenden Informationssystemen führen, kann sich als nächstschwierigeres Kapitel die Frage anschließen, welche Aspekte hinzukommen, wenn man eine Interaktion von anderen realen Objekten, als nur den Benutzern, mit dem geplanten Informationssystem realisieren will. Dies führt zunächst zu den registrierenden Informationssystemen. Als sehr übersichtliches Beispiel kann man eines vom Typ "Eine Person leiht ein Buch in einer Bibliothek aus" verwenden (das Beispiel wurde für [2] ausgearbeitet). Der dabei auftretende Registrationsbedarf besteht darin, daß der jeweilige aktuelle Zustand eines Buchs ("vorhanden", "ausgeliehen von ..") und der Zustand jedes Ausleihers ("hat Bücher ..")



gespeichert und einem Bibliothekar verfügbar sein sollen. Aus der Analyse ergibt sich, daß die Implementation einer direkten Interaktion "BUCH an INFORMATIONSSYSTEM: Mein Zustand ist nun.." ohne eine technische Aufrüstung der Bibliothek ("automatische Buchidentifikation", "diebstahlsichere Selbstbedienungsschleuse") unmöglich wäre. Unser Entwurf geht entlang des konventionellen Wegs über den Bibliothekar als vermittelndes Medium. Der obere Teil von Abb.6 beschreibt den Realitätsausschnitt mit den Objekten AUSLEIHER, BUCH und BIBLIOTHEKAR, sowie einer Interaktion AUSLEIHEN. Es besteht ein Registrierungsbedarf für AUSLEIHER und BUCH mit je einer Interaktion, und ein Interaktionsbedarf des BIBLIOTHEKARs mit der Realisierung dieses Registrierungsbedarfs. Der untere Teil von Abb.6 beschreibt eine mögliche Realisierung als neues Objekt INFORMATIONSSYSTEM mit drei neuen Interaktionen für den BIBLIOTHEKAR. Es liegt auch nahe, die Realisierung der Interaktion "Ausleihen" als ein synthetisches Informationsobjekt zu entwerfen, welches die Kommunikation der an einer Ausleihe beteiligten Informationsobjekte synchronisiert. Aus dem alltäglichen Leben kennen wir diese objektfizierten Interaktionen gut. Die deutsche Sprache erlaubt z.B. generell die Substantivierung eines Verbs ("handeln" -> "Handlung"). In der Verwaltungssprache kennt man wichtige, genau definierte Interaktionen als "Vorgänge" und "Sachen". In der Informatik treten sie sowieso auf, z.B. in der Datenbanktechnik als "Transaktionen". Es ist ein grundsätzlicher Unterschied, daß bei simulierenden Systemen keine Interaktionen als Informationsobjekte modelliert werden müssen, bei registrierenden System aber regelmäßig. Eine schöne Übung ist es, das Beispiel, nachdem es vollständig modelliert und implementiert ist, um die Interaktion ZURÜCKGEBEN zu erweitern. Die Schüler erfahren, wie natürlich und modular sich ein objektorientiertes Modell erweitern läßt und wie problemlos sich eine Implementation an neuen Bedarf anpassen läßt - falls sie sich vorher präzise an das Modell gehalten hat.

5 Informationssysteme Typ 3: Regelnde Systeme

Während registrierende Systeme keine eigene Interaktion mit den Objekten der Realität haben, ist dies für regelnde Systeme unerlässlich. Sie wirken auf die Umwelt, die Umwelt wirkt auf sie, es entstehen Regelkreise, die sehr bewußt analysiert und modellieren werden müssen. Dies ist gegenüber den registrierenden Systemen ein zusätzlicher Aspekt, er kann schon bei relativ wenigen Objekten, die aber alle stark interagieren, eine Bereitschaft zum Denken in Wirkungsnetzen (Laterales Denken) erfordern [4]. Daher ist es sinnvoll, sie erst nach den simulierenden und registrierenden Systemen zu behandeln.

Unser Beispielvorschlag einer automatischen Lagerverwaltung (Abb.7) enthält bei nur drei Objekten bereits zwei Regelkreise. Die Werkstatt kann beim Informationssystem Material

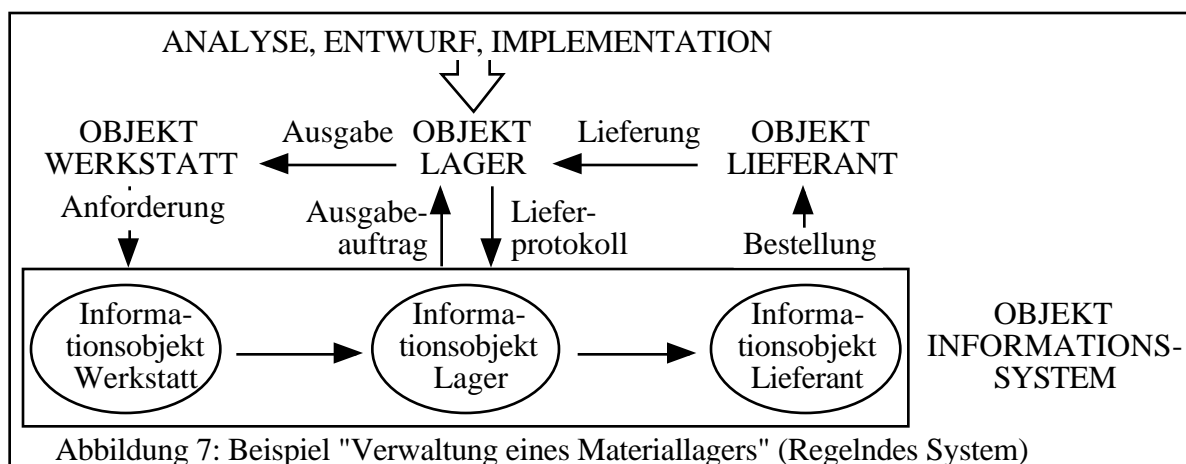


Abbildung 7: Beispiel "Verwaltung eines Materiallagers" (Regelndes System)

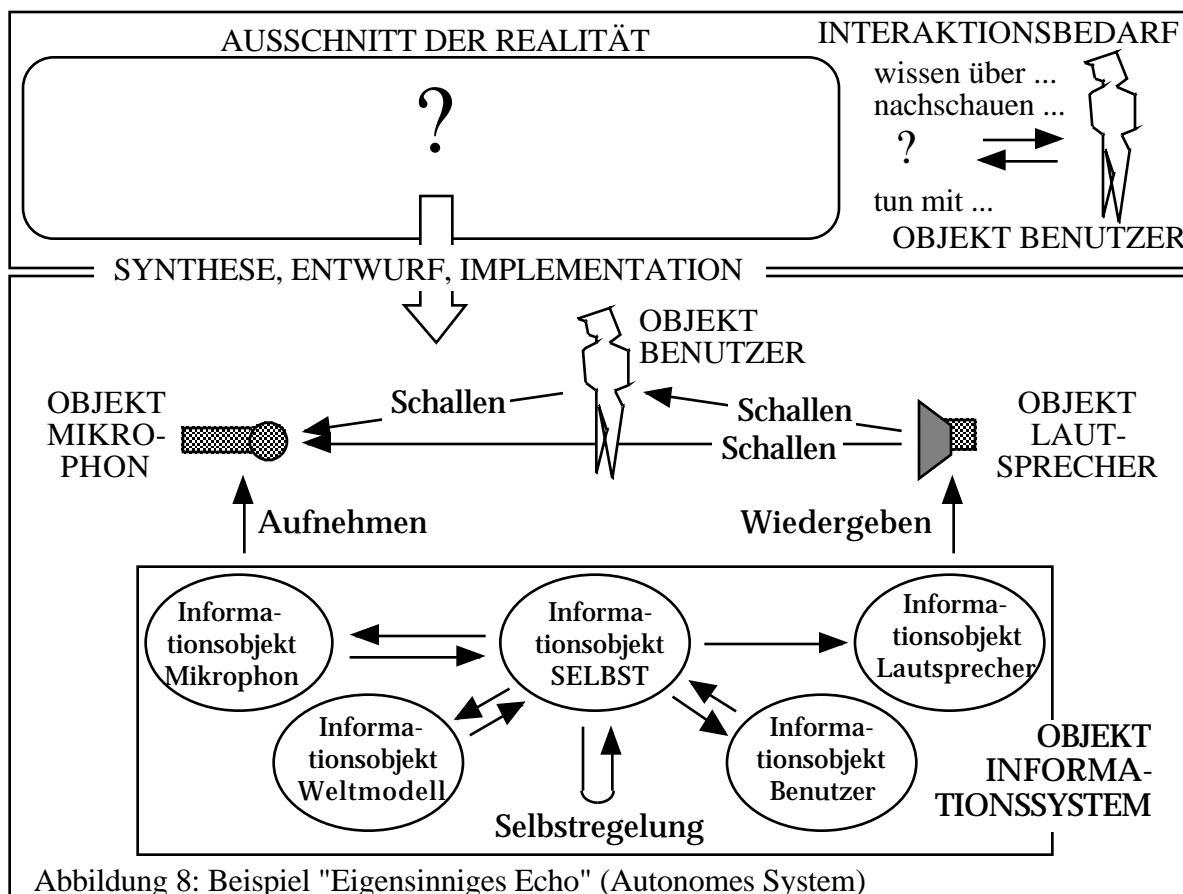
anfordern; ist es vorhanden, bekommt das Lager einen Auftrag und gibt Material an die Werkstatt aus; die verbraucht es und fordert wieder an, usw. (Kreis 1). Sinken die Lagervorräte unter ein verabredetes Minimum, dann erhält der Lieferant vom Informationssystem parallel zum Kreis 1 eine Bestellung und liefert irgendwann neues Material an das Lager; das Lager berichtet dem Informationssystem davon, dieses prüft, ob das Minimum noch unterschritten ist, usw. (Kreis 2).

6 Informationssysteme Typ 4: Autonome Systeme

Autonome Systeme können erst dann betrachtet werden, wenn das Prinzip des regelnden Systems verstanden wurde. Als neue Aspekte kommen hinzu:

- Die Autonomieeigenschaft ist vollkommen synthetisch, daher entfällt hierfür der Analyse-schritt. Ihre Modellierung beginnt mit einem kreativen Entwurf eines Informationsobjekts SELBST (Selbstbeschreibung), sowie einer Kommunikation mit sich SELBST (Selbstregelung).
- Ein autonomes System verfügt über direkte Interaktionen mit Objekten der Realität (diese heißen gelegentlich auch "Sensoren" und "Aktoren" [3]), aber nun haben die Objekte der Realität (auch die BENUTZER) keine eigenen Interaktionen mit dem System mehr. Trotzdem gibt es natürlich im Allgemeinen Regelkreise zwischen dem System und seiner Realität, weil (vergleiche Beispiel zu Abb.5) im Entwurf Kommunikation auch gegen die Richtung einer Interaktion laufen kann.

Unser Beispielsvorschlag ist das "Eigensinnige Echo". Ein Informationssystem ist mit Mikrofon (Sensor) und Lautsprecher (Aktor) verbunden. Gelegentlich nimmt es Schall aus



der Realität auf und speichert ihn, gelegentlich gibt es aufgenommenen Schall wieder ab, allerdings verzögert, möglicherweise verfremdet, vermischt und keineswegs immer chronologisch. Dieses Verhalten wird durch das Informationsobjekt SELBST gesteuert, welches zufällige und variable Eigenschaften enthalten kann. Anhand von Merkmalen des aufgenommenen Schalls können nach geheimen Kriterien diese Eigenschaften über die Selbstregelung laufend verändert werden. Für einen Benutzer zeigt ein solches System ein interessantes Verhalten: Man weiß zwar nie, ob es einem überhaupt gerade zuhört, aber das, was es ausgibt, ist auch nicht ganz beziehungslos zu dem, was man mal zu ihm gesagt hat.

7 Didaktische Linien beim Entwerfen von Informationssystemen

Die Linie "Sehen-Verstehen-Ändern-Selbertun":

Didaktische Linien beim Erwerb von Wissen und Fähigkeiten sollten so strukturiert werden, daß das Wissen und die Fähigkeiten in zyklischer Weise aufgebaut werden. In jedem Erweiterungszyklus sollten die Aktivitäten des Sehens, des Verstehens, des Änderns und des Selbertuns angeboten werden. Ein objektorientierter Ansatz für den Entwurf von Informationssystemen ist aus dieser Sicht besonders naheliegend, weil in der Realität vieles ohnehin als System interagierender Objekte zu sehen und zu verstehen ist. Anhand von Beispielen aus der Realität kann man lernen, was das Verhalten und die Zustände eines Objekt sind und wie die Objekte untereinander interagieren. Man kann sehen und verstehen, wie man in einer Analyse jedes Objekt als Einzelnes und das System als Ganzes beschreiben kann. Man kann sehen und verstehen, wie man in einem Entwurf die vielen sequentiellen Abläufe der einzelnen Objekte und das parallele Verhalten von mehreren Objekten miteinander in Attributen, Methoden und Interaktionen modelliert. Dann kann man versuchen, im Modell das Verhalten der Objekte zu ändern. Die Linie läßt sich dadurch abschließen, daß man ein kleines System von den Schülern anhand ihres eigenen Modells wieder spielen läßt. Ein einfaches Beispiel ist das Modell eines Ballspiels in einem Kreis von Kindern mit den Objekten "Kind", "Kreis", "Ball" und der Interaktion "Werfen".

Die Linie "Analyse-Entwurf-Implementation":

Ein objektorientierter Modellierungsprozeß ist in sich selber zyklisch und besteht aus den Phasen Analyse, Entwurf und Implementation. Diese Phasen zu erlernen, kann mit einer sehr geringen Anzahl von Objekten begonnen werden. In jeder Phase sollte man wieder der didaktischen Linie "Sehen-Verstehen-Ändern-Selbertun" folgen. Dabei können die Objektbeschreibungen bzw. Informationsobjekte verändert werden, nachdem man gesehen und verstanden hat, wie die realen Objekte funktionieren. Es können dem Realitätsausschnitt Objekte hinzugefügt werden, nachdem man durch Analysieren und Beschreiben gesehen und verstanden hat, wie er sich als Ganzes verhält. Man kann Informationsobjekte in mehrere trennen oder eines aus mehreren zusammensetzen. Art und Umfang der Änderungen können zyklisch gesteigert werden. Der Anfang könnte mit kleinen Änderungen an einzelnen Objekten gemacht werden. Dann kommen größere Änderungen, bei denen mehrere Objekte geändert werden müssen. Das Selbertun (Entwurf und eventuell Implementation) eines Informationssystems würde dann am Ende des Lernprozesses stehen. Das Selbertun kann man erleichtern, indem man Objekte, deren Analyse, Entwurf oder Implementation einen höheren Schwierigkeitsgrad aufweist, in drei Bibliotheken zur Verfügung stellt (Bibliothek der Objektbeschreibungen, Bibliothek der Informationsobjekte, Bibliothek der implementierten Objekte). Die Schüler brauchen von diesen Objekten nur das Verhalten zu sehen und zu verstehen, nicht die Art der Implementation. Sie brauchen nur zu wissen, was die Interaktions- bzw. Kommunikationsmöglichkeiten eines Objekts sind und welches Verhalten das Objekt danach jeweils hat.

Die Linie "Evaluation, Evaluation und nochmals Evaluation":

Evaluieren kann und soll man in jeder Phase des Entwicklungsprozesses eines Informationssystems. Evaluieren (nach der Modelltreue) bedeutet erstens Sehen und Verstehen, ob die Informationsobjekte des Modells in ihrem Verhalten und in ihrem Lebenslauf mit den Objekten aus der Realität übereinstimmen, und ob das ganze Modell auch dasselbe Verhalten hat, wie der betrachtete Realitätsausschnitt. Evaluation kann z.B. durch das Testen einer fertigen Implementation oder durch die Animation eines Modells (Rollenspiel der Schüler, Simulation auf dem Papier) geschehen. Wenn bei der Evaluation eines Modells relevante Abweichungen zur (existierenden oder gewünschten) Realität auftreten, führt das nach einer Kritik der Analyse zu Änderungen im Entwurf. Änderungen, die dann wieder in Änderungen der Informationsobjekte resultieren. Eine zweite Art von Evaluation (nach der Zweckmäßigkeit) geht der Frage nach, ob das entworfene oder implementierte Informationssystem in der erweiterten Realität den existierenden Interaktionsbedarf befriedigen würde bzw. befriedigt. Dies kann sowohl zu einer Änderung der Realität als auch wieder zu Änderungen von Informationsobjekten führen. Z.B. daß man nach der Analyse des Bibliotheksbeispiels aus dem Bedarf von Interaktionen "AUSLEIHER->INFORMATIONSSYSTEM" und "BUCH->INFORMATIONSSYSTEM" heraus tatsächlich eine Änderung der Realität beschließt (Magnetstreifen in den Büchern, computerlesbare Benutzerkarten, automatische Ausgangsschleusen, ...). Evaluieren ist also das Prüfen von Beziehungen zwischen Realität, Modellen und versuchten Modellrealisierungen. Diese Kompetenz zu lernen, ist außerordentlich wichtig, weil sie jeder Benutzer, Kunde, Arbeitnehmer, Staatsbürger, etc. braucht, wenn er Nutzen und Konsequenzen eines von wem auch immer eingeführten oder geplanten Informationssystems bewerten will. Dagegen ist Implementieren eine zweitrangige Spezialistenkompetenz.

Die Linie "simulierend-registrierend-regelnd-autonom":

Anhand der Beispiele von Kapitel 3 bis 6 kann man nachvollziehen, daß es prinzipielle Schwierigkeitsstufen des Analysierens und Entwerfens gibt, die sich in vier Typen von Informationssystemen darstellen lassen. Sie sind so charakterisiert:

- Simulierende Systeme sind 1-zu-1-Modelle der Objekte und Interaktionen der Realität. Sie haben keine Interaktion mit der Realität und diese nicht mit ihnen (Benutzer ausgenommen).
- Registrierende Systeme können synthetische Informationsobjekte enthalten, zu denen es keine Entsprechung in der Realität gibt. Objekte der Realität haben eine Interaktion zum registrierenden System. Der Antrieb des Systems ist extern.
- Regelnde Systeme haben Interaktionen mit Objekten der Realität und diese wieder mit ihnen, so daß sich Interaktionskreisläufe ergeben. Das System ist intern und extern angetrieben.
- Autonome Systeme haben auch Interaktionen mit Objekten der Realität, diese aber nicht mehr mit dem System. Der Antrieb des Systems verlagert sich ganz nach innen in eine Selbstbeschreibung und eine Selbstregelung.

Diese Gliederung macht es möglich, ein Beispiel Schritt für Schritt gezielt so zu verändern und zu erweitern, daß die Schüler bei ihrem zyklischen Sehen-Verstehen-Ändern-Selbertun jeweils die nächste Systemstufe erreichen.

7.1 Die Stellung objektorientierten Denkens in einem Informatik-Curriculum

Ist objektorientiertes Analysieren und Entwerfen eine Spezialfähigkeit und kann in die Endphase des Informatikunterrichts plaziert werden, oder ist es etwas so Grundlegendes, daß es von Beginn an im Curriculum vorkommen sollte? Man kann argumentieren, objektorientiertes Denken so früh wie möglich in den Informatikunterricht zu integrieren. Der Erwerb von Kompetenz in Analyse und Entwurf ist ein Prozeß. Man lernt erst durch Erfahrung, relevante Objekte zu sehen. Zwar kann man zunächst alles als Objekt auffassen, was man will, und die

Realität beliebig ordnen und klassifizieren. Für kleine Simulationen ist das auch ausreichend, aber auf die Dauer ist es nicht praktisch. Modelle sollten auf eine gewisse Weise zweckmäßig sein, und gut zu modellieren, ist Erfahrungssache. Wenn ein Schüler zu spät mit dem Modellieren beginnt, kann er diese Erfahrung nicht mehr aufbauen. Im Gegenteil wird seine Objektsicht dann sehr stark durch die Denkweisen des vorangegangenen Unterrichts und die ihm bereits geläufigen Werkzeuge gelenkt - es fließt Implementationsdenken in die Analyse und den Entwurf ein. Die Modelle werden unnatürlich. Das steht nicht im Widerspruch dazu, daß zu den aus der Analyse der Realität heraus gewonnenen Informationsobjekten im Entwurf oft noch virtuelle Informationsobjekte hinzukommen müssen (z.B. "objektifizierte Interaktionen"). Diese zu sehen und richtig einzusetzen ist ebenfalls Erfahrungssache, hat aber nichts mit Implementation zu tun.

8 Schluß

Man kann objektorientiertes Denken besonders gut lernen, wenn man entsprechend den vorgestellten didaktischen Linien die Gelegenheit hat, ein selbst oder von anderen implementiertes informationelles Modell interaktiv zu ändern, zu verfeinern, zu erweitern und mit anderen implementierten Modellen zu kombinieren. Auch wenn man damit ganz klein und ohne viele Voraussetzungen beginnen kann, ist objektorientiertes Modellieren ein weittragendes Denkinstrument. Es erlaubt unter anderem auf direkte Weise:

- die Darstellung asynchroner Parallelität (Nebenläufigkeit) von Prozessen.
- die Beschreibung der Kausalität des Verhaltens interagierender Objekte.
- eine beliebige Differenzierung der Arten von Interaktionen zwischen Objekten.
- eine dynamisch-objektive Wahl der Verteilung und Einteilung des eigenen Weltbildes auf die Objekte.
- eine syntaktisch korrespondierende Beschreibungen der Realität (als Ergebnis der Analyse) und eines diese erweiternden Informationssystems (als Ergebnis eines Entwurfs).

Objektorientiertes Denken verlagert die Schwerpunkte des Unterrichts von der Programmierkompetenz hin zu einer Modellierungs- und Evaluationskompetenz. Das Produkt des Schülers ist weniger wichtig, als seine Bereitschaft, dieses Produkt in gute Beziehungen zur Realität zu bringen, und es zyklisch zu verbessern und zu erweitern. Er lernt zudem, daß man die Realität in sehr unterschiedlichen Modellen beschreiben kann, die alle gleichwertig sind. Das Ziel, ein Informationssystem "aus"nutzen zu können, tritt hinter die Fähigkeit zurück, ein modelliertes oder implementiertes Informationssystem auf seine Realitätstreue und Zweckmäßigkeit hin zu betrachten, sowie konstruktive Erweiterungs- und Verbesserungsvorschläge machen zu können. Die Schüler sollen erfahren, daß interaktives softwaretechnisches Operieren in komplexen Informationssystemen möglich ist; daß man dazu nicht das gesamte Informationssystem, insbesondere nicht seine Implementation, verstanden haben muß, und daß man mit Komplexität auf kontrollierte Weise umgehen kann.

Objektorientiertes Denken unterstützt das Bemühen, ein lineares (produktorientiertes) Problemlösen durch ein zyklisches (prozeßorientiertes) Problemlösen zu ersetzen, der Erkenntnis folgend: "Ein Informationssystem ist niemals aktuell, niemals korrekt, niemals optimal, niemals objektiv. "

9 Literatur

- [1] Crutzen, C. K. M. et al.: "Oriëntatie op informatica: mens, machine en informatieverwerking" (Einführung in die Informatik: Mensch, Maschine und Informations-

verarbeitung). Fernkurs. Teil 1 (Information, Kommunikation), ISBN 90-358-1384-7; Teil 2 (Technik, Methodik, Gesellschaft), ISBN 90-358-1385-5; Open universiteit, Heerlen (Nederland), 1994.

- [2] Crutzen, C. K. M. et al.: "Object-georiënteerde analyse (Een aanpak voor het HBO-I)". HBO-I platform, Utrecht, Juni 1994.
- [3] Hein, H.-W.: "Agenda-Systeme". In: Rembold/Dillmann/Levi (Hrg): Autonome Systeme, Universität Karlsruhe, 1991, Seiten 215-225.
- [4] Hein, H.-W. et al.: "Programmieren im Team (Simulation einer Biosphäre)". In: Bericht über die Schülerakademie (St. Peter-Ording, 08-24. Juli 1993), Verein Bildung und Begabung e.V., Bonn, 1993.
- [5] Kant, I.: "Kritik der reinen Vernunft". (Zitat aus "Allgemeine Anmerkungen zur transzendentalen Aesthetik , §8 Abschnitt III"), J. F. Hartknoch, Riga, 1787.
- [6] Wittgenstein, L.: "Tractatus logico-philosophicus". (Zitat aus Satz 2.0121), Suhrkamp, Frankfurt am Main, 1977, Seite 12.