

# Von vernetzten fundamentalen Ideen zum Verstehen von Informatiksystemen – Eine Unterrichtserprobung in der Sekundarstufe II

Peer Stechert

Didaktik der Informatik und E-Learning  
Universität Siegen  
Hölderlinstraße 3  
57068 Siegen  
stechert@die.informatik.uni-siegen.de

**Abstract:** Die Förderung des Informatiksystemverständnisses ist eine vorrangige Aufgabe des Informatikunterrichts. Die vorliegende Arbeit stellt die exemplarische Umsetzung des Unterrichtsmodells für das Verstehen von Informatiksystemen mit Lernenden in der Sekundarstufe II vor und beschreibt den kognitiven Zugang zu vernetzten fundamentalen Ideen in Informatiksystemen. Es werden objektorientierte Entwurfsmuster als Wissensrepräsentation vernetzter fundamentaler Ideen der Informatik verwendet.

## 1 Motivation und Stand der Forschung

Die Förderung des Informatiksystemverständnisses ist eine vorrangige Aufgabe des Informatikunterrichts. Ein entsprechendes Unterrichtsmodell des Autors wurde auf nationaler und internationaler Ebene diskutiert. Ziel dieses Artikels ist es, Erkenntnisse aus der praktischen Erprobung des Unterrichtsmodells für das Verstehen von Informatiksystemen<sup>1</sup> vorzustellen und Schlussfolgerungen für das Unterrichtsmodell aufzuzeigen. Der Ansatz des Unterrichtsmodells [St06a], d. h. eines theoretischen Rahmens für unterschiedliche Unterrichtskonzepte und -reihen in der Sekundarstufe II, basiert auf der Vernetzung von fundamentalen Ideen der Informatik [Sc93] in ausgewählten Entwurfsmustern [Ga95]. Fundamentale Ideen der Informatik sind zurzeit Inhalt und Ausgangspunkt mehrerer fachdidaktischer Publikationen. Schneider begann in Anlehnung an das Fach Mathematik eine Gliederung der Informatik nach konzeptuellen Leitideen. Dafür ordnete er der Leitidee „funktionale Modellierung“ fundamentale Ideen zu, die in dem Themenbereich zu erlernen sind [Sc06].

Der vorliegende Artikel beschreibt, wie vernetzte fundamentale Ideen der Informatik für das Verstehen von Informatiksystemen im Informatikunterricht der Sekundarstufe II mit Vorkenntnissen in der objektorientierten Modellierung erlernt werden können. Dafür

---

<sup>1</sup> Es wird die Definition des Informatiksystems aus [CS03, S. 301] verwendet.

werden nach außen sichtbares Verhalten, innere Struktur und konkrete Realisierung von Informatiksystemen systematisch untersucht. Der Lehr-Lern-Prozess wird insbesondere durch Einsatz dedizierter Aufgaben, Lernsoftware als Unterrichtsmittel und die Durchführung von Informatikexperimenten strukturiert. Das Unterrichtsprojekt wurde im Schuljahr 2006/2007 an einem nordrhein-westfälischen Gymnasium durchgeführt.

## 2 Forschungsmodell

### 2.1 Forschungsmethodik

Die Forschungsmethodik ist inspiriert von praxisorientierter Fachdidaktikforschung wie sie von Hubwieser auf der INFOS 2005 vorgestellt wurde [Hu05]. Für solch eine intervenierende Fachdidaktik ist es kennzeichnend, dass neue Konzepte der Fachdidaktik von den Forschenden in Feldstudien umgesetzt und evaluiert werden. Die Rückkopplung mit der Praxis erlaubt eine kritische Reflexion der theoretischen Ergebnisse. Derartige Interventionsstudien sind stichprobenbasiert bei kleiner Stichprobengröße. Dies ist der Heterogenität des Informatikunterrichts geschuldet, die aufgrund breit angelegter und somit ebenso weit interpretierten Rahmenpläne vorherrscht.

Als zweite Säule der Forschungsmethodik und Voraussetzung der Intervention tritt das didaktische System [BS02]. Es ist eine Formalisierung des Lehr-Lern-Prozesses und besteht aus Wissensstrukturen, Aufgabenklassen und Lernsoftware. Stärke dieses Ansatzes ist, dass das didaktische System in unserer Forschergruppe seit 2001 auf die Bereiche Objektorientierung, Internetworking und Verstehen von Informatiksystemen angewendet und kontinuierlich weiterentwickelt wird (vgl. auch [FS06]).



Abbildung 1: Entwurfs-, Interventions- und Evaluationszyklus der Unterrichtsentwicklung.

Eine Übersicht der Forschungsmethodik zeigt Abbildung 1. Während der Analyse des Forschungsfelds wurde der Bedarf der Lernenden bezüglich des Verstehens von Informatiksystemen unter Bezugnahme auf nationale und internationale Curricula und Empfehlungen untersucht. Die Kombination von Erkenntnissen sowohl aus der Fachwissenschaft und aus der Fachdidaktik erlaubt die Formulierung von Forschungshypothesen für den Lehr-Lern-Prozess.

In [St06b] wurde begründet, dass Entwurfsmuster informatikspezifische, extern darstellbare Wissensrepräsentationen von vernetzten fundamentalen Ideen sind. Einige nach fachdidaktischen Kriterien ausgewählte Entwurfsmuster sind gleichzeitig Lernmittel und Lerngegenstand. Zur Erstellung von Unterrichtsmitteln wurde vom Autor eine studentische Projektgruppe betreut, welche die Lernsoftware „Pattern Park“ entwickelte [PP07]. Damit können attraktive Aufgaben gestellt werden, die fundamentale Ideen und Entwurfsmuster in verschiedenen Sichten verknüpfen. Nächster Schritt in der Forschungsvorgehensweise war die Entwicklung eines Unterrichtsprojekts. Dafür mussten weitere Unterrichtsmittel und Schülertätigkeiten erstellt bzw. definiert werden. Auch die Formulierung von Lernzielen nach der überarbeiteten Bloom'schen Lernzieltaxonomie [St06c] wurde vorgenommen.

Die besondere Situation, Forschungs- und Lehrperson gleichzeitig zu sein, wird in der quantitativen Forschung kritisch gesehen. Deshalb und aufgrund der geringen Stichprobengröße ist die vorliegende Untersuchung qualitativ als Hinweis auf Machbarkeit, Durchführbarkeit und Akzeptanz bei den Lernenden zu interpretieren. Dafür wird davon ausgegangen, „dass die Teilnahme im Feld Empathie und Identifikation mit den Untersuchungspersonen voraussetzt, da erst so die Interpretationsprozesse der Untersuchungspersonen erfasst und verstanden werden können“ [At06, S. 94]. Merkmal solcher quasi-experimenteller Felduntersuchung im Sinne von Bassey – „previously developed theory is used as a template with which to compare the empirical results of the case study“ [Ba99, S. 31] – ist, dass sie in natürlichen, d. h. nicht randomisierten, im Zuge des Forschungsprozesses kaum veränderten Umgebungen stattfinden. Jedoch lassen Ergebnisse von quasiexperimentellen Untersuchungen „mehr Erklärungsvarianten zu als die Ergebnisse reiner experimenteller Untersuchungen, d. h. sie haben eine geringere interne Validität“ [BD02, S. 527]. Die qualitativ-formativen Evaluationsaktivitäten, d. h., die durch den Einsatz von Interview und Akzeptanzbefragung um eine Überprüfung des Lehr-Lern-Szenarios angereicherten Ergebnisse, führen zu einem in empirischen, explorativen Phasen erprobten Unterrichtsmodell [At06, S. 31].

## 2.2 Strukturierung des Lerngegenstands

Das Unterrichtsprojekt hat drei Lernphasen  $S_i$ , im Sinne von Basiskompetenzen, die im Fokus des Unterrichtsmodells für das Verstehen von Informatiksystemen stehen (vgl. [CS03, S. 658]):

- $S_1$ : Das Verstehen wesentlicher Aspekte des nach außen sichtbaren Verhaltens.
- $S_2$ : Das Verstehen von Aspekten der inneren Struktur, die auf fundamentalen Ideen der Informatik basieren.
- $S_3$ : Das Verstehen von Implementierungsdetails zur Entwicklung einer konkreten Realisierung.

Zur Vorbereitung wurde mit dem Informatiklehrer, in dessen Klasse das Unterrichtsprojekt durchgeführt werden sollte, vier Monate vor der Durchführung ein erster Entwurf der Unterrichtskonzeption diskutiert, um Umsetzungsprobleme frühzeitig zu erkennen.

Mit Blick auf das Schulcurriculum wurden die Lernziele zusammengestellt. Die Ergebnisse dieser Diskussionen flossen auch in [St06a] ein. Darüber hinaus hat der Autor in dem Kurs zwei Wochen vor der Durchführung hospitiert, um an vorangegangenen Unterricht anzuknüpfen. Dabei erwies es sich als unkompliziert, den bisherigen Unterricht nun nahtlos in einem Unterrichtsprojekt weiterzuführen, das dem Unterrichtsmodell für das Verstehen von Informatiksystemen folgte. Dies ist neben der Tatsache, dass konkrete Programme Unterrichtsgegenstand waren, auch auf die objektorientierten Entwurfsmuster zurückzuführen. Eingeführt in das Unterrichtsmodell, um vernetzte fundamentale Ideen der Informatik zu repräsentieren, sind sie als Strukturierungsmuster und Heuristiken aus der Informatikpraxis häufig mit wichtigen Informatikkonzepten verbunden, die wiederum Unterrichtsthemen sind. Somit finden sich für das Unterrichtsmodell sehr viele Anknüpfungspunkte an den traditionellen (hier objektorientierten) Informatikunterricht für eine Unterrichtsreihe zum Thema Verstehen von Informatiksystemen. Dennoch ist Verstehen von Informatiksystemen nicht zwangsläufig mit der objektorientierten Modellierung der Sekundarstufe II verknüpft. Vielmehr können im Sinne eines Spiralcurriculums besonders Themen und Fragestellungen aus  $S_1$  bereits früher im Informatikunterricht behandelt werden.

In der durchgeführten Unterrichtserprobung lag der Schwerpunkt auf dem Zusammenspiel zwischen dem nach außen sichtbaren Verhalten und der inneren Struktur eines zum Teil selbst erstellten Programms ( $S_1$  und  $S_2$ ). Für die einzelnen unten angegebenen Schwerpunkte wurden Teillernziele durch Operatoren aus den Anforderungsbereichen I (z. B. Wiedergabe von Kenntnissen), II (z. B. Anwenden von Kenntnissen) und III (z. B. Problemlösen und Werten) der Einheitlichen Prüfungsanforderungen Informatik [E-PA04] sowie nach Blooms überarbeiteter Taxonomie eingesetzt [AK01]. Exemplarische Schwerpunkte beim Verstehen des nach außen sichtbaren Verhaltens waren:

- $S_{11}$ : Die Datenstruktur Schlange und der Schlangentraversierung verstehen sowie weitere Anforderungen an das Entwurfsmuster Iterator erklären können.
- $S_{12}$ : Zugriffskontrolle anhand des Entwurfsmusters Proxy verstehen.

Die konkrete Umsetzung wird in Abschnitt 3.3 beschrieben. Exemplarische Schwerpunkte beim Verstehen der inneren Struktur waren:

- $S_{21}$ : Das Konzept der Schnittstelle anhand der Kombination von Schlange und Iteratortentwurfsmuster verstehen bei phänomenologischer Vorwegnahme der Zugriffskontrolle.
- $S_{22}$ : Zugriffskontrolle anhand des Entwurfsmusters Proxy sowohl statisch, d. h. via Klassendiagramm, als auch dynamisch mittels Sequenzdiagramm beschreiben können.

In der dritten Lernphase wurden folgende Schwerpunkte angestrebt:

- $S_{31}$ : Das Verstehen unterschiedlicher Traversierungsalgorithmen durch Modifikation des Iteratormusters.

S<sub>32</sub>: Das Verstehen einer auf Vererbung beruhenden Zuweisung von Zugriffsrechten durch Modifikation von Implementierungsdetails im Proxyentwurfsmuster.

Die Phase S<sub>3</sub> war nicht Beobachtungsziel des Projekts. Nur vereinzelt wurde existierender Quellcode von den Schülern<sup>2</sup> erweitert, um an vorhandenes Vorwissen anzuknüpfen.

### 3 Unterrichtserfahrungen

#### 3.1 Lerngruppe und zeitlicher Rahmen

Die Erprobung des Unterrichtsmodells fand im Herbst 2006 im Rahmen eines Grundkurses Informatik in der Jahrgangsstufe 12 eines Siegener Gymnasiums statt. Der Kurs setzte sich zusammen aus vier Schülerinnen und 19 Schülern. Dadurch, dass es sich hierbei um einen regulären Grundkurs handelte, sind valide qualitative Auswertungen möglich, da der Kurs nicht nur aus hoch motivierten Schülern besteht, wie es im Rahmen von Wahlfächern oft der Fall ist. Um die Ergebnisse der Fallstudie weitergehend verallgemeinern zu können, sollten die Schüler Vorkenntnisse in objektorientierter Modellierung besitzen und möglichst keine Spezialisierung im technisch-naturwissenschaftlichen Bereich vorweisen. Die Schüler nutzten bereits seit etwa einem halben Schuljahr die objektorientierte Modellierung u. a. mit der Klassenbibliothek „Stifte und Mäuse“ [LL07]. Vererbung, erstellen von Klassen und erzeugen von Objekten sowie ereignisgesteuerte Programmierung waren bekannt ebenso wie grundlegende Programmierkonzepte, z. B. Schleifen und Variablen. Es zeigte sich während der Unterrichtsfolge, dass Vererbung eine Fehlerquelle darstellte (siehe Abschnitt 3.3).

Die Erprobung umfasste 12 Unterrichtsstunden zu drei Stunden pro Woche. Zusätzlich wurden ein Abschlusstest und eine Akzeptanzbefragung mit den Schülern durchgeführt. Der Unterricht wurde von der Forschungsperson gehalten und vom Informatiklehrer sowie einem weiteren Mitarbeiter des Instituts für Didaktik der Informatik und E-Learning der Universität Siegen gezielt beobachtet, um erste Erkenntnisse über typische Bearbeitungsstrategien und Fehler zu gewinnen. Es wurde angenommen, dass die Schüler keine Vorkenntnisse zur systematischen Erkundung des nach außen sichtbaren Verhaltens eines Informatiksystems aufweisen.

Die Unterrichtsstunden wurden sowohl mit dem Lehrer als auch mit weiteren Mitarbeitern des Instituts für Didaktik der Informatik und E-Learning vor- und nachbereitet. Der Informatiklehrer stand abschließend für ein Leitfaden-Interview zur Verfügung. Die Erprobung wurde unterbrochen durch eine Woche, in der der Informatiklehrer (aufgrund einer Vortragsreise des Autors) unabhängig vom Unterrichtsprojekt unterrichtete. Darüber hinaus gab es eine Verzögerung durch einen beweglichen Ferientag direkt im Anschluss, so dass insgesamt eine anderthalbwöchige Pause entstand.

---

<sup>2</sup> Es werden wenn möglich geschlechtsneutrale Formulierungen verwendet. Wenn die männliche Form gewählt wird, sind damit, im Sinne eines generischen Maskulinums, immer Frauen und Männer gleichermaßen gemeint. Frauen mögen sich nicht ausgeschlossen fühlen.

### 3.2 Unterrichtsmethodik und technischer Rahmen

Bei der Gestaltung des Unterrichtsprojekts wurde besonders darauf geachtet, dass die Schülertätigkeiten im Mittelpunkt standen. Somit wurde möglichst wenig in Lehrervorträgen umgesetzt, stattdessen war das Unterrichtsprojekt schülerzentriert angelegt. Lehrervorträge nahmen nur wenige Minuten pro Unterrichtsstunde ein. Partnerarbeit war die vorherrschende Arbeitsform, sowohl an den Rechnern als auch bei schriftlichen Arbeitsaufträgen.

Die räumliche Gestaltung des Informatiklabors unterstützte die Trennung von Theorie und Informatikexperimenten mit Rechnern durch den zentralen Kommunikationsbereich und die dezentral an den Seiten aufgestellten Rechner. Das Schulintranet besaß eine Verbindung zum Internet und eine von allen gemeinsam nutzbare Speicherpartition des Dateiservers ermöglichte den problemlosen Austausch größerer Dateien. Der vordere Demonstrationsbereich des Lehrers war mit Whiteboard, Rechner und Projektor zur Präsentation ausgestattet.

### 3.3 Lernphasen und Problemstellen im Unterrichtsprojekt

Ziel der Erprobung war es, den Zusammenhang zwischen dem nach außen sichtbaren Verhalten und der inneren Struktur von Informatiksystemen zu verdeutlichen. Um Fehlvorstellungen analysieren zu können, wurden sämtliche in den Unterrichtsstunden handschriftlich oder digital erstellten Schülerlösungen eingesammelt. Nach Abschluss des Unterrichtsprojekts wurde ein Test in Form einer schriftlichen Übung durchgeführt. Um dessen Wichtigkeit für die Schüler zu unterstreichen, sollten die Ergebnisse in die Gesamtnote einfließen. Aus diesem Grund wurde die schriftliche Übung nicht anonymisiert durchgeführt.

Roter Faden des Unterrichtsprojekts war das Wechselspiel von Erkundung des nach außen sichtbaren Verhaltens eines Programms und der Betrachtung und Erstellung von Modellen der inneren Struktur. Startpunkt war ein Programm für eine Arztpraxis mit einer Warteschlange, das bereits im vorhergehenden Informatikunterricht von den Schülern mittels Klassendiagramm modelliert und in der Programmiersprache Objekt-Pascal mit Delphi 6 geschrieben worden war. Es wurde erweitert um das Iteratormuster ( $S_{11}$ ).

Erst für das Thema Zugriffskontrolle ( $S_{12}$ ) wurde ein neues Programm untersucht, das die Vergabe von Zugriffsrechten realisierte. Dies geschah jedoch immer mit Rückbezug zum „Arztinformationssystem“. Spiralförmig wurden somit die Lernphasen wiederholt und auf die Themen Datenstruktur Schlange, Iteration und Zugriffskontrolle angewendet.

Die Hospitation in der Klasse kurz vor Beginn des Unterrichtsprojekts machte deutlich, dass Klassendiagramme als wichtiges Strukturbeschreibungsmittel eines Informatiksystems ohne weitere Vorbereitung eingesetzt werden konnten. Allgemein fiel während der Hospitation die Schwierigkeit der Schüler auf, zwischen Daten und Darstellung zu unterscheiden – ein Abgleich zwischen Datenhaltung und Darstellungsebene fand in den Schülerlösungen dementsprechend oft nicht statt. Dies sollte ein erster Ansatzpunkt für

die Erkundung des nach außen sichtbaren Verhaltens des Programms im Unterrichtsprojekt sein.

Datenstruktur Schlange und Iterator: Als erster Schwerpunkt zu Beginn des Unterrichtsprojekts wurde das nach außen sichtbare Verhalten des „Arztinformationssystem“ systematisch erkundet und anschließend schlossen die Schüler auf zugrunde liegende informatische Konzepte. Hierfür wurde ein Experimentiervorgang beschrieben, der aus sechs Schritten besteht und die sechs kognitiven Prozesse umfasst [St06c], die in der überarbeiteten Lernzieltaxonomie nach Bloom angegeben sind [AK01]. Er bestand aus dem Aufstellen von Hypothesen über das Systemverhalten, dem Beschreiben und Dokumentieren des Experimentierablaufs, den Rückschlüssen aus dem tatsächlichen Verhalten auf informatische Konzepte, dem Identifizieren von Sonderfällen wie der leeren Schlange, dem Überprüfen der Umsetzung der Sonderfälle im Informatiksystem sowie Auswirkungen des Einsatzes des Informatiksystems. Insbesondere das Erkennen von möglichen Konzepten fiel anfangs bei Schlange und Iterator schwer ( $S_{11}$ ), obwohl das Listenkonzept bekannt war. Bei der Auswertung der ersten Durchführung des Informatikexperimentes wurde nur bei etwas mehr als der Hälfte der abgegebenen Lösungen das Konzept der Schlange wieder erkannt. Der Sonderfall, dass die Schlange leer, bzw. die Frage, ob die Schlange voll sein kann, wurde im Plenum diskutiert.

Die Demonstration der Informatikexperimente, sei es durch Lehrperson oder Schüler, unterstützt das Unterrichtsprojekt zum Verstehen von Informatiksystemen stark, da die neuen systematischen Vorgehensweisen zur Analyse erläutert werden konnten. Als zweiter Schwerpunkt galt es, die innere Struktur des Programms anhand des Quellcodes und eines unvollständigen Klassendiagramms zu untersuchen ( $S_{21}$ ). Die Umsetzung der identifizierten informatischen Konzepte in eine Klassenstruktur war hier Schwerpunkt. Um dynamische Abläufe in der inneren Struktur beschreiben zu können, wurden während der ersten Woche in einer Partnerarbeit Objektdiagramme zu verschiedenen Zeitpunkten erstellt. Relativ überraschend war für die Schüler die Einsicht, dass der Iterator eine Variante der Zugriffskontrolle ist ( $S_{21}$ ). Das Verstehen des Systemverhaltens führte dazu, in der inneren Struktur Fehler schneller finden und beheben zu können ( $S_{31}$ ).

Zugriffskontrolle: Bei der Erkundung der Zugriffskontrolle ( $S_{12}$ ) war den Schülern das Vorgehen zur systematischen Erkundung des Systemverhaltens bekannt. Überraschend war für die Schüler bei der inneren Struktur der auf Vererbung basierende Ansatz zur Überprüfung der Zugriffsrechte (Abbildung 2). Da das Verstehen von dynamischen Aspekten essenziell für Informatiksystemverständnis ist, wurde zur Beschreibung der Abläufe bei der Zugriffskontrolle ( $S_{22}$ ) das Sequenzdiagramm genutzt. Einstiegsaufgabe war die Erstellung eines so genannten Interaktionsdiagramms, anschließend wurden einfache Sequenzdiagramme mit Elementen der Lernsoftware „Pattern Park“ erstellt.

Die Einführung der Zugriffskontrolle mittels Proxy ermöglichte den Schülern, das Prinzip der Zugriffsrechte zu verstehen und selbst für bestimmte Situationen zu gestalten. So konnten neben Administrator, Benutzer und Gast weitere Anwender (-gruppen) mit bestimmten Rechten versehen werden ( $S_{32}$ ).

Rolle der Lernsoftware: Durch Modularisierung bei der Entwicklung der Lernsoftware „Pattern Park“ konnten für die unterrichtliche Erprobung schon vorab Animationen zu Iteration (S<sub>11</sub>) und Zugriffskontrolle (S<sub>12</sub>) sowie Übungsmodule mit Sequenzdiagrammen (S<sub>22</sub>) eingesetzt werden. Letzteres geschah bei der Nutzung eines Proxys zur Realisierung von Zugriffsschutz. Insbesondere das Übungsmodule griff eine Lebensweltsituation handlungsorientiert auf und schuf so den Brückenschlag zwischen dem nach außen sichtbaren Verhalten und der inneren Struktur eines Informatiksystems einerseits und die Verbindung der statischen und dynamischen Strukturbeschreibung andererseits. Die Beobachtung der Bearbeitungsweisen hat ergeben, dass gerade diese Aufgaben, die das nach außen sichtbare Verhalten von Informatiksystemen und die Analyse der inneren Struktur verknüpfen, Schlüssel zum Verstehen waren, denn hier wurden unterschiedliche Sichten auf das Informatiksystem kombiniert.

Außerdem zeigte sich während des Unterrichtsprojekts, dass das frühzeitige Vorführen der Animation den Schülern nicht nur die Problemstellung klarer werden ließ, sondern gleichzeitig durch den Bezug zum Entwurfsmuster, einem „Lösungsmuster“, eine konkretere Vorstellung von Lösungsansätzen unterstützte. Abbildung 2 zeigt das Klassendiagramm des Proxy zur Realisierung der Zugriffskontrolle. Anhand des Klassendiagramms mussten die Schüler ein korrektes Sequenzdiagramm eines typischen Ablaufs in dem vorgegebenen Szenario kreieren.

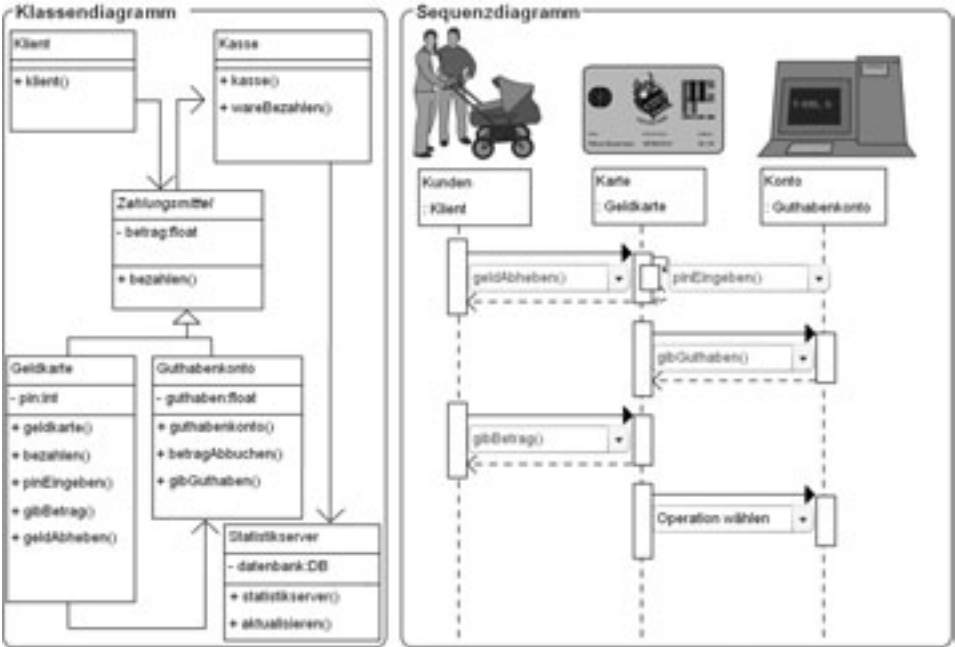


Abbildung 2. Klassen- und Sequenzdiagramm zum Zugriffsschutz aus der Lernsoftware „Pattern Park“



### 3.4 Auswertung des Abschlusstests

Für die schriftliche Übung wurden Multiple-Choice-Aufgaben und Lückentexte gewählt. Als Stimuli für die Testitems dienten u. a. Klassendiagramme. Inhaltlich waren die Schwerpunkte der schriftlichen Übung:

- nach außen sichtbares Verhalten von Informatiksystemen,
- innere Struktur von Informatiksystemen,
- Zusammenhang zwischen nach außen sichtbarem Verhalten und innerer Struktur,
- Zugriffskontrolle mittels Entwurfsmuster Proxy,
- Iteration mittels Entwurfsmuster Iterator.

Für die schriftliche Übung wurden zwei Varianten A und B eingesetzt, so dass ein Thema bei Bedarf in der einen Variante mit Lückentext, in der anderen mit Multiple-Choice-Aufgabe geprüft werden konnte. Die Bearbeitungszeit betrug 30 Minuten. Insgesamt wurden 21 Tests bearbeitet (Variante A: 11; Variante B: 10). Die Ergebnisse sind überdurchschnittlich gut ausgefallen (gerundete Werte): Minimalwert: 53%, Maximalwert 100%, Mittelwert: 75%, Median: 77%. Die häufigsten Werte (Modi) waren 63%, 67% und 77% mit je drei Ausprägungen. Der Mittelwert bei den Mädchen lag mit 72% leicht unter dem Gesamtdurchschnitt.

Bei den Fragen zum nach außen sichtbaren Verhalten hatten die Schüler Schwierigkeiten in Bezug auf eine systematische Erkundung. Hieraus könnte der Schluss gezogen werden, dass es kaum Vorkenntnisse bezüglich eines solchen Vorgehens gab. Im scheinbaren Widerspruch dazu steht die Aussage fast aller Schüler, dass das Erkunden der inneren Struktur schwieriger als das Erkunden des nach außen sichtbaren Verhaltens sei. Es haben alle Schüler angegeben, dass zum Verstehen des nach außen sichtbaren Verhaltens eines Informatiksystems ein systematisches Experimentieren notwendig sei und dass unerwartetes Verhalten eines Informatiksystems Rückschlüsse auf dessen innere Struktur liefere. Mit der Vorgehensweise zum Erkunden der inneren Struktur hingegen waren im Test weniger Schwierigkeiten verbunden. Dies mag an den Vorkenntnissen zur Erstellung von Klassendiagrammen und auch zu Delphi-Projekten liegen. Es zeigte sich eine teilweise Überforderung der Lernenden bei der Verbindung des nach außen sichtbaren Verhaltens mit der inneren Struktur. Angebracht gewesen wäre hier eine Aufteilung in kleinere Teilaufgaben.

Die Hauptschwierigkeiten lagen bei der Beschreibung der inneren Struktur der Datenstruktur Schlange und des Iterator, da oftmals falsche Assoziationen angegeben wurden. Des Weiteren ist einigen Schülern die Unterscheidung zwischen Schlange und Iterator nicht gelungen. Außerdem wurden bei dem Transfer des Proxy auf eine neue Situation Fehler gemacht, obwohl die zum Unterricht ähnliche Aufgabe zur Zugriffskontrolle gelöst wurde. In weiteren Unterrichtsprojekten sollten deshalb mehr Transferaufgaben hinsichtlich ähnlicher Situationen und Strukturen zu bearbeiten sein.

Die Ergebnisse aus der Analyse der Explizierung und Reflexion der jeweiligen Schüler-Vorgehensweisen in der Lerngruppe lassen sich jedoch nicht eindeutig Fehlvorstellungen beim Verstehen von Informatiksystemen zuordnen. Insbesondere ist zu vermuten, dass typische Fehlvorstellungen der Objektorientierung sehr dominant sind und Schüler-vorstellungen von Informatiksystemen überlagern. Zur Realisierung der Zugriffskontrolle wurden beispielsweise oft falsche Vererbungsbeziehungen verwendet.

### **3.5 Auswertung der Akzeptanzbefragung und des Interviews mit dem Informatiklehrer**

Im Nachgang des Unterrichtsprojekts wurden die Lernenden des Kurses schriftlich befragt. Die ausgefüllten Fragebögen wurden von den Lernenden anonymisiert. Die Fragebögen enthielten jeweils ca. 35 Aussagen aus den vier Bereichen „Informatikunterricht allgemein“, „Schwierigkeitsgrad und Lernstoff“, „Befragung zum Unterrichtsthema“ und „Einschätzung des Lernfortschritts in den einzelnen Lernbereichen“, zu denen die Lernenden durch Ankreuzen ihre Zustimmung oder Ablehnung bei (vorwiegend) vier vorgegebenen Skalenwerten äußern sollten zuzüglich der Möglichkeit sich zu enthalten. Ein Ziel war es, die Motivation der Lernenden zu analysieren. Gefragt wurde deshalb auch nach außerschulischer Beschäftigung mit dem Thema, weitergehendem Interesse sowie Einschätzung des persönlichen Nutzens. Weitere ausgewählte Einflussgrößen bzw. Einstellungen der Lernenden waren beispielsweise Geschlecht und Anzahl versäumter Unterrichtstermine, um die Antworten interpretieren zu können. In der Akzeptanzbefragung ergab sich, dass von 23 Schülern 16 (70%) den Schwierigkeitsgrad des Unterrichtsprojekts als angemessen und nur vier (17%) ihn als hoch empfanden. Der Stoffumfang wurde von 16 (70%) als angemessen und von sechs (26%) als viel wahrgenommen. Dieser Aspekt wird auch durch zehn schriftliche Anmerkungen gestützt, welche die geringe zur Verfügung stehende Zeit für das Bearbeiten einiger Aufgaben beanstanden. Zu dem eigenen Lernfortschritt bei den inhaltlichen Themen der Analyse, der inneren Abläufe und des Aufbaus von Informatiksystemen geben jeweils 16 (70%) oder mehr Schüler an, einiges bzw. viel dazugelernt zu haben. Die Konzentration auf die Aufgaben fiel leicht. Der Aussage, dass Hilfsmittel wie Arbeitsblätter und (Lern-) Software ausreichend und in guter Qualität vorhanden waren, wurde von 17 (74%) Schülern etwas, bzw. voll zugestimmt. Dies ist sicherlich auch auf den Einsatz ausgewählter Animationen und Übungen aus der Lernsoftware „Pattern Park“ zurückzuführen. Durchaus mit den auf den Basiskompetenzen liegenden Zielen des Unterrichtsmodell im Einklang steht, dass fünf (22%) Schüler anzeigten nichts und elf (47%) nicht viel zur Delphi-Programmierung dazugelernt zu haben. Dass elf (48%) Schüler angaben eher keinen bzw. drei (13%) bestätigten keinen Spaß daran zu haben, ihr Verständnis für das Thema zu vertiefen, offenbart ein Motivationsproblem, dem in folgenden Erprobungen durch stärkeren Bezug zu Alltagsproblemen und anderen Fächern begegnet werden muss. Sequenzdiagramme als Darstellungsform von Abläufen wurden von den Schülern positiv aufgenommen. Abschließend ist zu erwähnen, dass 16 (70%) Schüler angaben, für sich etwas dazugelernt zu haben, und dass die Mehrheit von 19 (82%) Schülern der Aussage, dass die Inhalte des Unterrichtsprojekts für den Umgang mit Informatiksystemen sehr nützlich sind, etwas bzw. voll zustimmen. In dem Interview, für das der Informatiklehrer abschließend zur Verfügung stand, wurde wiederum bestätigt, dass der Schwierigkeits-

grad angemessen sei. Auch die Konzeption mit dem Wechsel der unterschiedlichen Untersuchungsbereiche der Informatiksysteme wurde befürwortet.

## 4 Schlussfolgerungen und Ausblick

Die Problemstellen im Unterricht haben eine Überarbeitung des Unterrichtsmodells eingeleitet, das im WS 2007/2008 mit Lehramtsstudierenden erneut erprobt werden soll. Neben Zugriffskontrolle wird das Unterrichtsprojekt erweitert um Themen wie Zustände, Kompatibilität und Zuverlässigkeit von Informatiksystemen. Als Entwurfsmuster bietet sich neben dem Zustandsmuster insbesondere der Adapter an, dessen Potential auf der Vernetzung von Systemkomponenten liegt. Da sich das Identifizieren der informatischen Konzepte allein anhand des nach außen sichtbaren Verhaltens eines Informatiksystems anfangs als kognitive Hürde erwies, soll als weitere Vorgehensweise systematisches Testen verwendet werden: D. h. Aufgaben, die sowohl das Verändern des Systemverhaltens durch unterschiedliche Eingaben als auch durch Modifikation der inneren Struktur erfordern. Damit werden unterschiedliche Sichten auf ein Informatiksystem genutzt, die ein umfassenderes Verstehen ermöglichen. Die Lernsoftware „Pattern Park“ kann den Lehr-Lern-Prozess diesbezüglich unterstützen. Darüber hinaus soll bei Aufgaben im Unterrichtsprojekt eine Stärkung des Beobachtens vorgenommen werden.

Die gesellschaftlichen Auswirkungen des Einsatzes der Informatiksystemen sind mit Blick auf die aktuelle Diskussion um Bildungsstandards für die Informatik in das Unterrichtsmodell zu integrieren. Eine Schwierigkeit bei der Auswertung der Schülerlösungen war, dass gerade die aus der objektorientierten Modellierung bekannten Fehlvorstellungen, wie Unterscheidung zwischen Klasse und Objekt, auftraten und möglicherweise Fehlvorstellungen bezüglich Informatiksystemen überlagerten. Um diese dennoch identifizieren zu können, wird deshalb die kognitionspsychologische Methode des Laut-Denkens in einer Vorstudie von Lehramtsstudierenden an drei typischen Aufgaben zum Verstehen von Informatiksystemen evaluiert.

## Literaturverzeichnis

- [AK01] Anderson, L. W.; Krathwohl, D. R. (Hrsg.): A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives. Addison Wesley Longman, New York, 2001.
- [At06] Atteslander, P.: Methoden der empirischen Sozialforschung. 11. Auflage, Erich Schmidt Verlag, Berlin, 2006.
- [Ba99] Bassey, M.: Case study research in educational settings. Open University Press, UK, 1999.
- [BD02] Bortz, J; Döring, N.: Forschungsmethoden und Evaluation für Sozialwissenschaftler. 3. Auflage. Springer. Berlin, 2002.
- [BS02] Brinda, T; Schubert, S.: Didactic System for Object-oriented Modelling. In: Watson, D.; Andersen, J. (eds.): Networking the Learner. Computers in Education. Kluwer Academic Publisher, Boston, 2002; S. 473-482.
- [CS03] Claus, V.; Schwill, A.: Duden Informatik. Ein Fachlexikon für Studium und Praxis. Duden Verlag, Mannheim, 3. Auflage, 2003.

- [EPA04] KMK – Ständige Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland. Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik. URL: <http://www.kmk.org/doc/beschl/EPA-Informatik.pdf> (27.04.2007), 2004.
- [FS06] Freischlad, S.; Schubert, S.: Media Upheaval and Standards of Informatics. In: IFIP TC3 / WG 3.1, WG 3.3 & WG 3.5 Joint Conference “Imagining the future for ICT and Education”. Alesund, Norway, 2006.
- [Ga95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1995.
- [Hu05] Hubwieser, P.: Von der Funktion zum Objekt – Informatik für die Sekundarstufe I. 11. GI-Fachtagung Informatik und Schule INFOS, 2005; S. 27-41.
- [LL07] Bildungsserver Nordrhein-Westfalen: Objektorientierte Programmierung. URL: <http://www.learnline.de/angebote/oop/> (28.01.2007), 2007.
- [PP07] Franke, D.; Freischlad, S.; Friedrich, L.; Haug, F.; Klein, B.; Koslowski, R.; Stechert, P.; Ufer, J.: Projektgruppe Pattern Park, Universität Siegen, URL: <http://www.die.informatik.uni-siegen.de/pgpatternpark/> (24.04.2007), 2007.
- [Sc06] Schneider M.: Functional modelling, fundamental ideas and threads in the subject informatics; In Proceedings of Second International Conference on "Informatics in Secondary Schools. Evolution and Perspectives – ISSEP", Vilnius, Lithuania, 2006; S. 413-423.
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik 1, 1993; S. 20-31.
- [St06a] Stechert, P.: Unterrichtsmodellentwicklung zur Förderung des Informatiksystemverständnisses mit Entwurfsmustern. In Lecture Notes in Informatics: Schwill, A.; Schulte, C.; Thomas, M. (Hrsg.), 3. Workshop der GI-Fachgruppe Didaktik der Informatik, 2006; S. 89-98.
- [St06b] Stechert, P.: Informatics System Comprehension – A learner-centred cognitive approach to networked thinking. In: Education and Information Technologies 11 (2006) 3, Springer Netherlands, 2006.
- [St06c] Stechert, P.: Understanding of Informatics Systems – A theoretical framework implying levels of competence. In Proceedings of the 6th Baltic Sea Conference on Computing Education Research - Koli Calling, Koli, Finland, November 9-12, 2006; S. 128-131.