

**Lernumgebung für objektorientiertes Modellieren im Informatikunterricht —
LEO
– Projektgruppenantrag –**

1. PG–Thema:

Lernumgebung für objektorientiertes Modellieren im Informatikunterricht — LEO

2. PG–Zeitraum:

Wintersemester 2001/2002 und Sommersemester 2002

3. PG–Umfang:

jeweils 8 SWS

4. PG–Veranstalter:

Torsten Brinda, Didaktik der Informatik, OH16, R. E03, Tel.: 6142

Ludger Humbert, Didaktik der Informatik, OH16, R. E02, Tel.: 6141

5. PG–Aufgabe:

Motivation:

Von der Didaktik der Informatik werden für das Schulfach Informatik neue Konzepte gefordert. Während früher Inhalte im Informatikunterricht oft auf dem Wege der imperativen Programmierung, z.B. mit Systemen wie „NIKI der Roboter“, erarbeitet wurden, wird heute zunehmend gefordert, „informatisches Modellieren“ zu betonen. Von besonderer Bedeutung ist dabei der objektorientierte Ansatz, im Folgenden objektorientiertes Modellieren (OOM) genannt. Bislang hat der konzeptionelle Wandel allerdings nur wenige Schulen erreicht. Einerseits fehlt es noch an praxiserprobten und evaluierten Unterrichtskonzepten, die die Modellierung betonen, andererseits aber auch an den Lehr-Lern-Prozess stützenden Systemen. Gegenwärtig werden zur Erstellung und Bearbeitung von informatischen Modellen neben Tafel und Kreide, Papier und Stift auch professionelle Software-Entwicklungsumgebungen zur Unterstützung eingesetzt. Problem dieser Umgebungen ist, dass sie für die Rationalisierung des Software-Entwicklungsprozesses in der Industrie erstellt worden sind, insofern eine oft unüberschaubare Vielzahl an Werkzeugen und Konfigurationsmöglichkeiten für jeden erdenklichen Anwendungsfall bieten. Damit sind sie allerdings für den Einsatz im Informatikunterricht überladen und der Weg bis zu ihrer nutzbringenden Anwendung erfordert eine intensive Einarbeitung und Orientierung von Lehrenden und Lernenden, Zeit, die für wichtige Unterrichtsinhalte verloren geht.

Ziel:

Das Ziel der Projektgruppe besteht darin, eine Lernumgebung für den Informatikunterricht zu entwerfen und zu realisieren, in der Lernende mit objektorientierten Modellen experimentieren. So werden sie schrittweise an objektorientierte Konzepte und Modellierungstechniken herangeführt. Mit dieser Umgebung sollen sowohl einzelne Modellierungsschritte als auch die zielgerichtete Konstruktion von Modellen eingeübt werden können. Ziel der Projektgruppe ist es nicht, eine weitere, dafür aber didaktisch wertvolle, Software-Entwicklungsumgebung zu erstellen.

Für die Realisierung sind insbesondere folgende Teilziele bedeutend:

(a) **Verschiedene Zugangsmöglichkeiten zu den Inhalten:**

Um eine flexibel einsetzbare Umgebung zu erhalten und um unterschiedlichen Vorkenntnissen bei Lernenden gerecht zu werden, sind verschiedene Zugangsmöglichkeiten

zu den Inhalten erforderlich. So kann, neben einem spielerischen Zugang für Anfänger, für Fortgeschrittene ein eher formaler Zugang angeboten werden. Für die formale Repräsentation von Modellen wird dabei eine didaktisch begründete Auswahl von UML-Darstellungsformen verwendet. Verschiedene Zugangsmöglichkeiten zur Modellierung erfordern verschiedene Sichten auf ein zu erstellendes, zu erkundendes oder zu analysierendes Modell. Im Folgenden werden einige **Modellsichten** skizziert, die für die gegebene Zielsetzung hilfreich sein können.

1. Animationssicht

In einer Animationssicht kann für einen vorgegebenen Realitätsausschnitt eine Nachbildung bzw. eine Veranschaulichung durch Simulation der darin enthaltenen Objekte, ihrer Eigenschaften und der mit ihnen verbundenen Verhaltensmöglichkeiten erfolgen. Die zentrale Idee dabei ist, dass der Umgang mit und die Manipulation von Objekten leichter zu verstehen sind, als deren Klassifizierung und der Aufbau von Klassenbeziehungen. Lernende sollen zuerst in Kontakt mit Objekten kommen, zunächst ohne etwas über Klassen und deren Beziehungen wissen zu müssen. Durch die Visualisierung von Objekten, ihrer Attribute, ihrer Beziehung und ihres Verhaltens kann der Lehr-Lern-Prozess entscheidend unterstützt werden. Dabei soll bewusst keine abstrakte Form der Darstellung gewählt werden (z. B. in Form von Objekt- oder Klassendiagrammen), sondern der modellierte Realitätsausschnitt soll möglichst anschaulich und prägnant aufbereitet werden (beispielsweise sollen in einer Modellwelt, in der es um Katzen und Mäuse geht, animierte Katzen und Mäuse zu sehen sein). Das zugrundeliegende Modell ist abgeschlossen und auf der Animationsebene nur insoweit erweiterungsfähig, wie es vom Ersteller der Animation vorgesehen wurde. Die Wahl einer geeigneten Modellwelt ist aus didaktischer Sicht eine zentrale Aufgabe, da motivierende Wirkung, Ausdrucksstärke und Handhabbarkeit in Einklang gebracht werden müssen.

2. Objektsicht

Die Objektsicht steht in direkter Verbindung mit der Animationssicht und stellt eine formale Darstellung der Zusammenhänge in Form von Objektdiagrammen dar. Diese Sicht kann aber auch unabhängig von der Animationssicht verwendet werden. Der Benutzer kann hier Objekte erzeugen, diese mit Attributen versehen und diesen Attributen Werte zuweisen, ferner kann er Beziehungen zu anderen Objekten herstellen. Diese Sicht ist deshalb wichtig, da sie bzgl. der Abstraktion einen Zwischenschritt bei der Konstruktion von Klassendiagrammen darstellt. Ferner soll der Benutzer dabei unterstützt werden, ein das Objektdiagramm realisierendes Klassendiagramm zu erstellen. Werden später in der Animations- oder der Interaktionssicht die Nachrichtenflüsse bestimmter Methoden ausgearbeitet, so können diese hier veranschaulicht werden, indem der zeitliche Ablauf der Kommunikation zwischen den Objekten simuliert und Veränderungen der Objektzustände dargestellt werden. Dazu kann jedes einzelne Objekt angewählt und beauftragt werden, eine Botschaft an ein bekanntes Objekt zu versenden.

3. Klassensicht

In der Klassensicht wird die Klassenstruktur des Modells dargestellt. Ein Klassendiagramm kann hier schrittweise konstruiert werden, mit Klassen, Attribu-

ten, Methoden und Relationen zwischen Klassen. Ferner kann eine Klassenstruktur halbautomatisch aus einem Objektdiagramm erzeugt werden oder es kann überprüft werden, ob ein Objektdiagramm eine mögliche Realisierung des Klassendiagramms darstellt. Ebenso kann halbautomatisch aus einem Klassendiagramm ein Objektdiagramm erzeugt werden.

4. Zusammen mit der **Interaktionssicht** kann zu einer Methode ein Interaktionsdiagramm erstellt werden, die Struktur des Algorithmus einer Methode kann in der **Struktogrammsicht** erstellt werden.

Wünschenswert ist ferner die Realisierung oder Anbindung eines Codegenerators, der zu den konstruierten Modellen Quelltext extrahiert erzeugt. Änderungen in einer Sicht sollen soweit wie möglich in allen anderen Sichten angepasst werden.

(b) **Konfigurierbarkeit durch den Lehrenden:**

Um eine sowohl flexible als auch zielgerichtete Einsetzbarkeit im Unterricht zu ermöglichen, ist die weitgehende Konfigurierbarkeit des Systems durch den Lehrenden zwingend erforderlich. Lernende unterschiedlichen Kenntnisstands erfordern unterschiedliche Handhabungsmöglichkeiten. Der Lehrende soll zu jedem Zeitpunkt zentral festlegen können, welche Möglichkeiten zur Ergänzung und zur Einflussnahme Lernende abhängig von ihrem jeweiligen Kenntnisstand haben. Die Einflussmöglichkeiten durch den Lehrenden umfassen dabei zwei **Teilbereiche**:

1. Auswahl der Modellsichten, mit denen ein Lerner arbeiten kann,
2. Festlegung der Komplexität jeder einzelnen Sicht.

(c) **Assistentenfunktion / Lehrkomponente**

Um die zielgerichtete Modellerstellung zu unterstützen, ist ferner eine Assistentenfunktion (Lehrkomponente) hilfreich, die den Lernenden bei der Modellerstellung führt, in dem Teilmodelle überprüft, Fragen gestellt und Anregungen gegeben werden.

(d) **Plattformunabhängigkeit**

Da das System nach seiner Erstellung zum breiten Einsatz im Informatikunterricht kommen soll, ist eine weitestgehend plattformunabhängige Realisierung (Realisierung in Java) anzustreben, da sich in Schulen jede erdenkliche Betriebssystemvariante finden lässt, von Windows- über Linux- bis hin zu Macintosh-Systemen.

Anwendungsszenario:

Vorstellbar ist ein Anwendungsszenario, in dem der Lehrende im Schulintranet von seinem Arbeitsplatz aus die Systeme an den Schülerarbeitsplätzen fernkonfigurieren kann, um zu bewirken, dass diesen vom System lediglich die Funktionalität angeboten wird, die zur Bearbeitung einer Aufgabe erforderlich ist. Damit kann das Augenmerk der Lernenden im System gezielt auf die interessierenden Aspekte gerichtet werden. Während Anfänger ein Modell nur in geringem Maße verändern können sollten (z.B. Attribute hinzufügen oder entfernen) stehen Fortgeschrittenen erheblich erweiterte Möglichkeiten offen (z. B. Methoden verändern oder hinzufügen). Damit besteht auch die Möglichkeit, mit Inhalten von OOM auf verschiedenen Abstraktionsebenen zu experimentieren. Im Anfangsunterricht könnten beispielsweise an einer Miniaufgabe alle erdenklichen Modellsichten vorgestellt werden, die Sichten für sich würden sich alle auf sehr wenige Teilaspekte beschränken,

z.B. würde in der Klassensicht keine Vererbung und Assoziation auftreten und in der Objektsicht gäbe es nur ein Objekt, das aus wenigen Teilobjekten bestünde. Modellierungstechniken zur Konstruktion komplexerer Modelle wären einfach nicht verfügbar.

6. PG-Teilnahmevoraussetzungen:

- Software-Technik (V)
- Simulation (M)
Graphische Systeme (M)
- Software-Ergonomie (W)

Legende: (V) Voraussetzung, (M) mindestens eins, (W) wünschenswert

7. Minimalziel :

- (a) Erstellung der Anforderungsspezifikation für die Lernumgebung
- (b) Entwurf, Realisierung, Dokumentation und Test der Lernumgebung

8. Literatur:

- BALZERT, H.: *Lehrbuch der Objektmodellierung*. Spektrum, Heidelberg, 1999.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.: *The Unified Software Development Process*. Addison-Wesley, Reading, Massachusetts, 1999.
- BRINDA, T.; SCHUBERT, S.: *Didaktisches System für objektorientiertes Modellieren*. Forschungsbericht Nr. 752, Universität Dortmund, 2001.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.: *Entwurfsmuster*. Addison-Wesley, Bonn, 1996.
- HUBWIESER, P.: *Didaktik der Informatik*. Springer, Berlin, 2000.
- LANDESINSTITUT FÜR SCHULE UND WEITERBILDUNG (Hrsg.): *Von Stiften und Mäusen*. Verlag für Schule und Weiterbildung, Bönen, 1999.
- MAGENHEIM, J.; SCHUBERT, S.: *Evaluation of teacher education in informatics*. 16th World Computer Congress 2000, In: BENZIE, D.; PASSEY, D. (eds.): *Proceedings of Conference on Educational Uses of Information and Communication Technologies*, China, Beijing, August 21st-25th, 2000, 181-184.
- RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W.: *Objektorientiertes Modellieren und Entwerfen*. Prentice-Hall, London, 1993.
- SCHUBERT, S.: *The impact of modelling in informatics education on collaborative learning with school Intranets*. In: HOGENBIRK, P.; TAYLOR, H. G. (eds): *The Bookmark to the School of the Future*. Kluwer Academic Publishers, Boston, 2000, 247-258.
- VAN WEERT, T.: *Integration of informatics into education*. In: TINSLEY, D.; WATSON, D. (eds.): *INTEGRATING INFORMATION TECHNOLOGY INTO EDUCATION*. Chapman & Hall, London 1995.