

Entwicklung von Software zur Exploration im Bildungskontext

Torsten Brinda, Andreas Hoffmann

Didaktik der Informatik und E-Learning
Universität Siegen
Hölderlinstr. 3
57068 Siegen
brinda@die.informatik.uni-siegen.de
hoffmann@die.informatik.uni-siegen.de

Abstract: Software zur Exploration im Bildungskontext ist insbesondere im nicht-technischen Bereich noch wenig verbreitet, wenngleich durch deren Anwendung lerneraktive Bildungsprozesse ermöglicht werden, die neben der Fachkompetenz auch die Lernkompetenz fördern. Auf der Basis von Literaturstudien zum explorativen Lernen mit Informatiksystemen werden grundlegende Anforderungen an Software zur Exploration im Bildungskontext präzisiert. Aufbauend auf diesen Anforderungen wird ein Architekturkonzept für Software zur Exploration entwickelt und zusammen mit der Gestaltung zweier Fallstudien aus der Digitaltechnik und der Anglistik exemplarisch überprüft.

1 Motivation

Exploratives Lernen gilt im Bildungskontext sowohl an Schulen als auch an Hochschulen als besonders erstrebenswert, da es sich dabei um ein aktives, forschend-entdeckendes, autonomes Lernen handelt [Ke01, 217], das oftmals mit großem Engagement und Begeisterung selbstreguliert und scheinbar mühelos erfolgt. Aus diesem Grunde wird im Bildungsbereich oftmals angestrebt, entsprechende Lernprozesse zielorientiert anzuregen, um Lernende damit insbesondere auf lebensbegleitendes Lernen (life-long learning – LLL) vorzubereiten. Es zeigt sich allerdings ein scheinbarer Widerspruch, der zwischen mühelosen, nicht angeordneten, explorativen, durch Lernende kontrollierten und damit pädagogisch besonders wertvollen Lernaktivitäten einerseits und planvollen bzw. durch Lehrende organisierten Lehr-Lern-Prozessen andererseits besteht. Dieser Widerspruch lässt sich jedoch auflösen, indem nicht die *Herstellung* explorativer Lernprozesse, sondern durch entsprechende Gestaltung des Bildungsprozesses und in ihm verwendeter Medien deren *Ermöglichung* bzw. *Anregung* im Mittelpunkt des „didaktischen Designs“ steht [ebd., 220].

Eine besondere Situation zeigt sich in der informatischen Bildung in der Sekundarstufe II, da dort Schüleraktivitäten und exploratives Lernen oftmals mit professionellen Pro-

grammiersprachen und Softwareentwicklungsumgebungen verknüpft werden. Um erfolgreich modellieren und eigene Lösungen gestalten und bewerten zu können, müssen die Lernenden viele programmiersprachliche Details erlernen und komplexe Software-Entwicklungsumgebungen anwenden. Diese Zeit geht verloren für so wichtige Themen, wie Wirkprinzipien von Informatiksystemen. Die gewünschte Bildungsqualität wird durch diese Art von Medieneinsatz nur auf Umwegen oder überhaupt nicht erreicht. Guter und für Lernende interessanter Informatikunterricht ist möglich ohne diese enge Bindung an eine spezielle Implementierungssprache, wenn im Unterricht Analyse und Entwurf von Lösungen stärker betont und durch lernergerichtete Informatiksysteme unterstützt werden. In [Br04] wurden solche Werkzeuge für den Bereich des objektorientierten Modellierens konzipiert und erprobt.

Die vorliegende Arbeit stammt von Betreuer und Verfasser einer Informatik-Diplomarbeit [Ho03], deren Ziel es war, eine domänenunabhängige Vorgehensweise zur Entwicklung von Software im Bildungskontext, hier als Explorationsmodule bezeichnet, zu entwickeln, zu begründen und in Fallstudien zu erproben. Ziel war dabei nicht die Gestaltung von E-Learning-Materialien in Form von multimedialen Präsentationen, durch die Lernende ziellos hindurch navigieren können, sondern die Entwicklung von Software zur Anregung des entdeckenden Lernens im jeweiligen Themengebiet, die vollständig vom Lernenden gesteuert wird. Erforderlich für die zielgerichtete Interaktion mit dem Lernangebot ist ein Explorationswunsch oder eine fachlich fundierte Hypothese zum Lernmaterial. Der Explorationswunsch oder ein Erkundungsziel führen zur Erforschung des Systems. Bei der Erkundung bilden die Lernenden Hypothesen über Zusammenhänge, die sie anhand von Experimenten mit der Software bestätigen oder widerlegen. Ungewünschte Zielzustände führen zur Modifikation von Hypothesen (vgl. Abb. 1).

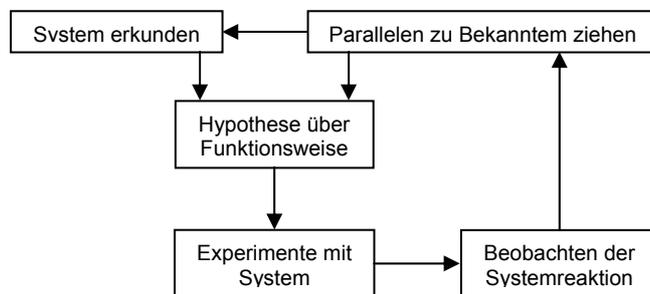


Abbildung 1: Exploratives Agieren (vgl. [Ho03, 10])

Keinesfalls ist es das Ziel, solche Handlungsszenarios zu programmieren. Zur Begleitung und zur Anregung der gewünschten fachlichen Entdeckungen benötigen Lernende aber geeignete Lernhilfen. Durch vielfältige Interaktionsmöglichkeiten [Ke01] und Visualisierung der Handlungskonsequenzen in verschiedenen Sichten [An01, 267] auf den Explorationsgegenstand werden sie zum aktiven Lernen angeregt. Im Sinne des konstruktivistischen Lernens [Vy78] sollen die Lernenden durch die Interaktion mit der Software dazu stimuliert werden, mentale Modelle über den jeweiligen Lerngegenstand zu konstruieren und schrittweise zu verfeinern [Ke01, 74ff].

2 Die Bedeutung von Software zur Exploration im Bildungskontext

Um den Stellenwert von Explorationsmodulen im Bildungskontext zu beurteilen, werden Vor- und Nachteile für wesentliche Akteure in diesem Kontext erörtert. Wesentliche Akteure sind:

- Lehrende und Lehramtsstudierende,
- Lernende und
- Entwicklerinnen und Entwickler von Lehr-Lern-Materialien.

Da Explorationsprozesse stark lernergesteuert sind, ergibt sich daraus für *Lehrende* (und *Lehramtsstudierende*) die Anforderung einer besonders intensiven Planung, wenn sie entsprechend flexibel und beweglich im jeweiligen Lehr-Lern-Prozess agieren möchten. Durch die Heranführung von Lernenden an die Prozesse explorativen Lernens in softwarebasierten, geschlossenen Explorationsumgebungen können Explorationsprozesse durch Auswahl und Kombination von Explorationsmodulen durch den Lehrenden beeinflusst und das Erkenntnisinteresse der Lernenden auf die für den jeweiligen Lernprozess relevanten Aspekte gelenkt werden. Hierin liegt Vor- und Nachteil zugleich, weil damit nur die Entdeckungen möglich sind, die vom Entwickler des jeweiligen Explorationsmoduls vorgesehen wurden. Wenn die Lernenden entsprechend lernkompetent sind, ihre individuellen Lernprozesse zielgerichtet auf unterrichtsrelevante Lernziele zu lenken, können geschlossene Explorationsumgebungen schrittweise geöffnet werden. Generell ist Unterricht, der auf explorativem Lernen basiert, deutlich weniger steuerbar als traditioneller Frontalunterricht. Für Lehrende ergibt sich die Aufgabe, Problemsituationen zu entwerfen, diese mit Medien für explorativen Unterricht zu verknüpfen und Lernenden Handlungsrahmen für ihre Explorationsprozesse zu vermitteln [SM81]. Durch Leitfragen können Lehrende Lernende dazu stimulieren, ihre individuellen Erkundungs- und Entdeckungsprozesse auf die relevanten Aspekte zu lenken [Ne88]. Ein potentiell Problem besteht darin, dass der Zeitaufwand im Unterricht deutlich höher ist als bei traditionellem Frontalunterricht [St95, 64].

Für *Lernende* liegt das Potenzial von Explorationsmodulen im Wesentlichen in den Bereichen

- Entdecken fachlicher Zusammenhänge,
- Entwicklung der individuellen Lernkompetenz.

Das Entdecken fachlicher Zusammenhänge fördert die Sach- und Methodenkompetenz der Lernenden. Durch Explorationsmodule und deren Einbettung in den Unterricht wird ein handlungsorientierter, lernerzentrierter Zugang zu theoretischen Fachkonzepten ermöglicht, der sonst vielfach und oft ausschließlich auf dem Wege des Frontalunterrichts erfolgte. Verschiedene Lehrervorträge werden dadurch überflüssig. Die Fachkonzepte und deren innere Zusammenhänge können so von Lernenden exploriert werden, ohne dass durch langwierige, vorgelagerte Theoriephasen, in denen sehr viel, oft sekundäres, Faktenwissen erworben werden muss, zu viel Unterrichtszeit gebunden wird. Durch Explorationsmodule kann der relevante Inhalt in den Mittelpunkt des Unterrichts gerückt werden, und somit durch besondere, zielgruppenspezifische Formen der Veranschaulichung und der Interaktion eine Konzentration auf das Wesentliche erfolgen [St99]. Lernende erhalten die Möglichkeit der aktiven Auseinandersetzung mit dem Stoff. Mit auf-

tretenden Problemen werden sie nicht allein gelassen. Diese werden zu einem wichtigen Element der Diskussion in der Lerngruppe [Ke01, 278ff]. In der Interaktion mit den Explorationsmodulen können sie logische Strukturen und Gedankenmodelle in Form von Hypothesen mit Experimenten auf Wirksamkeit prüfen. Dadurch wird ihre Hypothesenbildung gefördert. Durch die Kombination aus Veranschaulichungen und Rückmeldungen des Systems werden ferner das Lernen aus Beispielen und das Lernen aus Fehlern gefördert. Orientiert an Leitfragen explorieren die Lernenden fachliche Zusammenhänge. Damit werden auch ihre Fähig- und Fertigkeiten beim explorativen Lernen sowie bei der selbstbestimmten Organisation von Lernprozessen geschult [ebd., 217ff]. Das steigert ihre Lernkompetenz und stellt damit eine gute Vorbereitung auf lebensbegleitende Lernprozesse dar. Durch das aktive, lernerzentrierte Lernen kann der Grad an intrinsischer Motivation und des Interesses am Unterricht erhöht werden [ebd., 139f].

Für *Entwicklerinnen* und *Entwickler* von Lehr-Lern-Materialien stellen vorhandene Explorationsmodule Anregungen und Ausgangspunkte für weitere Entwicklungen dar. Insbesondere können über die Analyse von Gemeinsamkeiten Werkzeuge gestaltet werden, die den Entwicklungsprozess rationalisieren. Vorhandene Explorationsmodule können die Entwicklung von Unterricht insgesamt bereichern, weil dadurch die fachdidaktische Diskussion angestoßen und neue Impulse für zukünftige Gestaltungsdimensionen gegeben werden können.

Lernen mit Explorationsmodulen soll traditionellen Unterricht keinesfalls ersetzen. Es stellt eine interessante Zugangsalternative zu fachlichen Inhalten in geeigneten Phasen des Lehr-Lern-Prozesses dar.

3 Ein Konzept für Explorationssoftware im Bildungskontext

3.1 Grundlegende Anforderungen an Explorationssoftware

Mit dem Ziel der Entwicklung eines Gestaltungskonzeptes für Explorationsmodule wurden zunächst ausgewählte informatische und lerntheoretische Grundlagen für Software zur Exploration im Bildungskontext untersucht [Ho03, 5ff; Br04, 113ff]. Nachfolgend werden wesentliche der identifizierten funktionalen, technischen und softwareergonomischen Anforderungen dargestellt und begründet. Um perzeptive Exploration zu fördern, müssen den explorierenden Sinnen Strukturen angeboten werden, die interaktiv entdeckt werden können. Für die Entdeckung von Unbekanntem sollte es dennoch bekannte Bereiche geben, die als Grundlage für das neue Wissen dienen. Dies korrespondiert mit der Tatsache, dass exploratives Lernen nur dann zuverlässig zu Lernerfolgen führt, wenn ein gewisses Maß an Fachwissen als Grundlage vorhanden ist. Verschiedene, synchronisierte, einzeln ein- und ausblendbare *Sichten* auf den Lerngegenstand ermöglichen es der lernenden Person die Auswirkungen ihres Handelns auf den Lerngegenstand auf verschiedenen Ebenen zu betrachten und die damit verbundenen Wissensbereiche bestmöglich zu vernetzen. Beim objektorientierten Modellieren können bspw. Etappen des Gestaltungsprozesses (Realitätsausschnitt – Modell – Produkt) verknüpft mit Betrachtungen von Statik und Dynamik die Auswahl von Sichten leiten [Br04]. Damit lassen sich auch verschiedene *Abstraktionsniveaus* und damit verschiedene Lernwege bestmöglich unterstützen. Um manipulative Explorationsprozesse anzuregen und zu

stützen. Um manipulative Explorationsprozesse anzuregen und zu unterstützen müssen der explorierenden Person Strukturen angeboten werden, die von ihr durch *Modellieren* und *Konstruieren* geändert und bewertet werden können [St99, No02]. Zentral ist, dass Änderungen am Explorationsgegenstand zu einer unmittelbaren und für den Lernenden beobachtbaren Reaktion des Systems führen. Zu beachten ist hierbei, dass Freiheitsgrade bei der Konstruktion und Anschaulichkeit der Darstellung in den Sichten konkurrierende Zielsetzungen sind. In einer Explorationsumgebung für objektorientiertes Modellieren führen bspw. Darstellungen, die die Semantik eines objektorientierten Modells realitätsnah visualisieren zwangsläufig zu einer Einschränkung des freien Gestaltens, da ansonsten in der „Realsicht“ nichtdefinierte Zustände auftreten können.

Das Zusammenwirken der zuvor genannten Anforderungsbereiche muss durch geeignete *Simulations-* und *Experimentiermöglichkeiten* unterstützt werden. Simulationsmöglichkeiten eignen sich insbesondere für technisch orientierte Lerngegenstände (z.B. Mechanik, [No02]), bei denen sich ein formales, berechenbares, zugrunde liegendes Modell identifizieren lässt, das nach einer Parametrisierung des Lernenden dann in seiner Dynamik schrittweise analysiert werden kann. Experimentierprozesse erfordern neben Möglichkeiten, Experimentszenarios zu gestalten und zu beobachten insbesondere auch die Möglichkeit zur *Stornierung* (und Wiederholung) von Aktionen, falls eine Handlung eines Lernenden zu einem ungewünschten Zustand führte. Um in einem solchen Fall zu einem gesicherten Zustand zurückkehren zu können, sind sogenannte Wiederaufsetzpunkte z.B. in ihrer Implementierung als „freezing points“ [Pa94] hilfreich. Dadurch erhält der Lernende zusätzliche Sicherheit für weitere Handlungen. In einem *Log-Mechanismus* werden alle relevanten Aktionen des Lernenden protokolliert und stehen diesem für die eigene Orientierung (z.B. Wiederholung einer zuvor erfolgreich ausgeführten Aktionsfolge) und Lehrenden z.B. für die Analyse von Lehr-Lern-Prozessen zur Verfügung. Abb. 2 zeigt die Zusammenhänge der funktionalen Anforderungen.

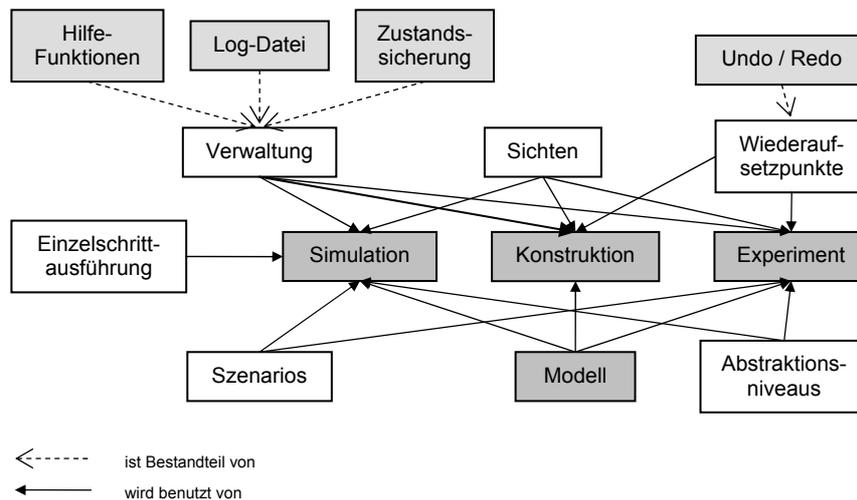


Abbildung 2: Zusammenhänge zwischen den funktionalen Anforderungen (vgl. [Ho03, 24])

Aus technischer Sicht sollte Explorationssoftware insbesondere *plattformunabhängig* sein und möglichst niedrige Anforderungen an die *Leistungsfähigkeit* des Rechners stellen, auf dem sie ausgeführt wird, da in öffentlichen Bildungsinstitutionen, in denen diese vorwiegend zum Einsatz kommen soll, in der Regel eher unterdurchschnittlich leistungsstarke Geräte vorzufinden sind.

Die softwareergonomischen Kriterien orientierten sich an der EN ISO-Norm 9241 (insb. Teil 10: „Grundsätze der Dialoggestaltung“). In diesem Zusammenhang sind besonders die Einschränkungen im Hinblick auf die *Selbstbeschreibungsfähigkeit* und die *Fehlertoleranz* von Explorationssoftware bedeutsam, die von Paul beschrieben wurden [Pa94]. Explorieren schließt eigenständiges Erkunden mit ein, womit sogar ein Mangel an Selbstbeschreibung gefordert wird. Weiterhin soll auch Explorationssoftware fehlertolerant sein, jedoch sollen Fehler beim Explorieren nicht primär vermieden werden, sondern die lernende Person soll diese erkennen und aus ihnen lernen. Die Explorationssoftware hat dabei die Aufgabe, die lernende Person bei der Fehlerbewältigung zu unterstützen.

Der so entstandene Anforderungskatalog [Ho03, 42] wurde überprüft und verfeinert, indem drei bestehende Systeme mit Explorationsanspruch (*Experimentierumgebung* aus [St99], *LEO*¹ und *mechADO*²) anhand dieser Kriterien analysiert wurden.

3.2 Entwicklung einer Architektur für Explorationssoftware

Ausgehend von den identifizierten Anforderungen an Explorationssoftware und den Ergebnissen der Analyse ausgewählter Bildungssoftware mit Explorationsanspruch (vgl. 3.1), wurden die Aktivitäten eines Lernenden innerhalb einer Explorationsumgebung präzisiert und mit Komponenten der Benutzungsschnittstelle in Relation gesetzt, um auf der Basis dieser Anwendungsfälle zu einem ersten Architekturentwurf für Explorationssoftware zu gelangen, der unabhängig von einem bestimmten Lerngegenstand ist (vgl. Tab. 1). Den Ausgangspunkt bildet das Erkunden der Benutzungsoberfläche und des Lerngegenstandes. Diese Erkundung führt zu Hypothesen über den Explorationsgegenstand, die in Experimenten mit dem System münden und darüber bestätigt oder widerlegt werden. Experimente bestehen bspw. in der Konstruktion und bzw. oder Konfiguration von Modellen und des Vergleichs der Systemantworten mit den zuvor aufgestellten Hypothesen. Die Sichten stehen dabei in Abhängigkeit zum Explorationsmodell. Ändert sich der Zustand des Explorationsmodells, dann ändern sich auch die Zustände der einzelnen Sichten. Lernende können so vielfältige Perspektiven im Sinne konstruktivistischen Lernens einnehmen. Dies wird weiterhin dadurch gefördert, dass auch die einzelnen Sichten hinsichtlich ihres Darstellungsumfangs konfigurierbar sind. Das für einen spezifischen Lernprozess Wesentliche kann so leicht in den Vordergrund gerückt werden. Bei der Konstruktion und Konfiguration von Modellen erstellt der Lernende aus

¹ LEO – Lernumgebung für objektorientiertes Modellieren im Informatikunterricht <http://www.didaktik-der-informatik.de/pgleo/> (aufgerufen am 01.06.04)

² <http://iug.uni-paderborn.de/Projekte/explo/> (aufgerufen am 01.06.04)

vorbereiteten Gestaltungselementen einen Aufbau und bzw. oder parametrisiert einen vorhandenen Aufbau.

Aktivität	Involvierte Schnittstellenelemente
Erkunden <ul style="list-style-type: none"> • Beobachten • Merkmale ein- und ausblenden 	(synchronisierte) Sichten Benutzungsschnittstelle
Konstruieren und Konfigurieren <ul style="list-style-type: none"> • Modell erstellen / bearbeiten • Modell laden / speichern 	Sichten mit Gestaltungselementen Laden- / Speichern-Dialog
Experimentieren <ul style="list-style-type: none"> • Verändern / Manipulieren • Stornierung • Log-Einträge setzen 	(synchronisierte) Sichten Undo, Wiederaufsetzpunkte Log-Fenster
Simulieren <ul style="list-style-type: none"> • Steuern (Start, Nächster Schritt, Stop) 	Simulationssicht
Steuern des Ablaufs <ul style="list-style-type: none"> • Fortführen, Abbrechen 	Benutzungsschnittstelle, insb. Statuszeile

Tabelle 1: Verknüpfung von Aktivitäten und Komponenten der Benutzungsschnittstelle

Beim objektorientierten Modellieren (Informatik) könnte darunter z.B. die Konstruktion eines Objektdiagramms verstanden werden, beim Sprachenlernen (z.B. Englisch) die Verknüpfung vorbereiteter Satzfragmente zu einem Satz mit bestimmter Semantik.

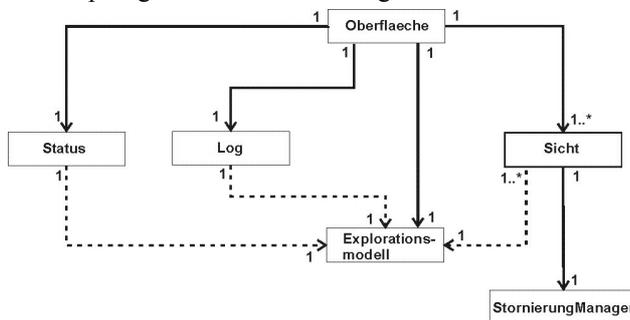


Abbildung 3: OOA-Modell eines Explorationsmoduls [Ho03, 53]

und ggfs. Simulation durchgeführt werden. Im Mittelpunkt steht das *Explorationsmodell*, auf das von Status, Log und den Sichten zugegriffen wird. Aus Gründen der Einfachheit wurde zunächst nur ein Explorationsmodell pro Explorationsmodul unterstützt. Der Ansatz ist aber auf mehrere Modelle beliebig erweiterbar.

Bei der Entwicklung des **objektorientierten Entwurfsmodells** stellte sich die Schwierigkeit, dass sich die Klassenstruktur an keiner bestimmten Anwendung orientieren konnte, da ein wesentliches Entwicklungsziel gerade in der Domänenunabhängigkeit bestand. Die Architektur wurde nach dem Model-View-Controller-Paradigma (MVC) strukturiert. Das Explorationsmodell stellt darin die Model-Komponente dar, die ver-

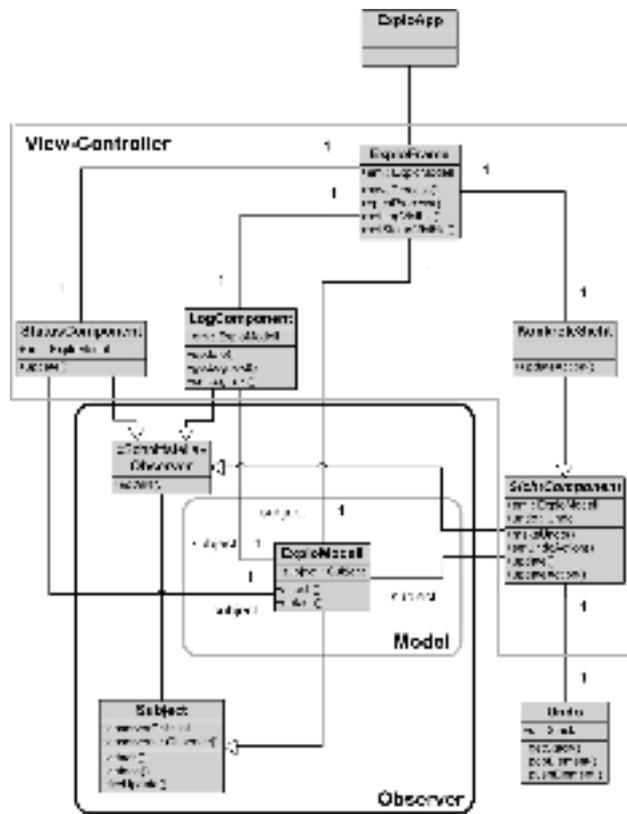


Abbildung 4: Entwurfsmodell für Explorationssoftware (nach [Ho03, 64])

Jede Sicht wird in einer eigenen Klasse gekapselt, Gemeinsamkeiten, wie z.B. der Zugriff auf das Explorationsmodell, wurden in einer abstrakten Basisklasse (*SichtComponent*) zusammengefasst (vgl. Abb. 4). Um die Integration der Sichten in den *ExploFrame* möglichst einfach zu gestalten, erbt *SichtComponent* von der Java-Standard-Klasse *JComponent*, die in Java als Basisklasse für praktisch alle Swing-Komponenten dient.

Die hier vorgestellte Architektur entstand verzahnt mit der Entwicklung zweier Prototypen (vgl. 4), die dazu dienen, einerseits induktiv aus den konkreten Produkten Verallgemeinerungspotential für das Architekturkonzept abzuleiten und andererseits deduktiv den jeweils gegebenen Stand an einer konkreten Realisierung zu überprüfen. Der erste Architekturentwurf wurde bei der Realisierung der Fallstudie *DigiExplo* (vgl. 4.1) angewendet. Die praktische Umsetzung ermöglichte die Spezifikation weiterer Anforderungen. Bspw. wurde in dieser Phase identifiziert, dass neben den Sichten auch der Status- und der Logbereich als Beobachter der Explorationsmodells konfiguriert werden müssen. Bei der Realisierung der zweiten Fallstudie (*LangExplo*, vgl. 4.2) wurde das Konzept der Wiederaufsetzpunkte durch die Einführung von Freezing-Points („Einfrieren“

schiedenen Sichten fungieren als View-Controller. Damit sind neue Sichten leicht integrierbar. Für die Realisierung der Anforderung, dass jede Änderung in einer Sicht eine sofortige Aktualisierung aller anderen Sichten nach sich zieht, wurde das Beobachter-Muster (Observer) nach [Ga95] eingesetzt. Sowohl die Sichten als auch der Status- und der Logbereich wurden als Beobachter des Explorationsmodells (*ExploModell*) konfiguriert. Da die Realisierung mittels Java erfolgen sollte und sowohl Anwendungen als auch Applets unterstützt werden sollten, wurde die Anwendungsklasse (*ExploApp*) von der Benutzungsschnittstelle (*ExploFrame* und zugeordnete Klassen) getrennt. *ExploFrame* regelt die Darstellung der einzelnen Komponenten.

bzw. „Auftauen“ eines Zustandes durch Aktivierung genau einer Schaltfläche) realisiert. Dafür wurde die Klasse *LangExploFrame* um entsprechende Methoden erweitert, das Architekturkonzept diesbezüglich ergänzt und die Fallstudie *DigiExplo* dementsprechend angepasst.

4 Fallstudien zur Überprüfung des Konzepts

4.1 Fallstudie *DigiExplo*

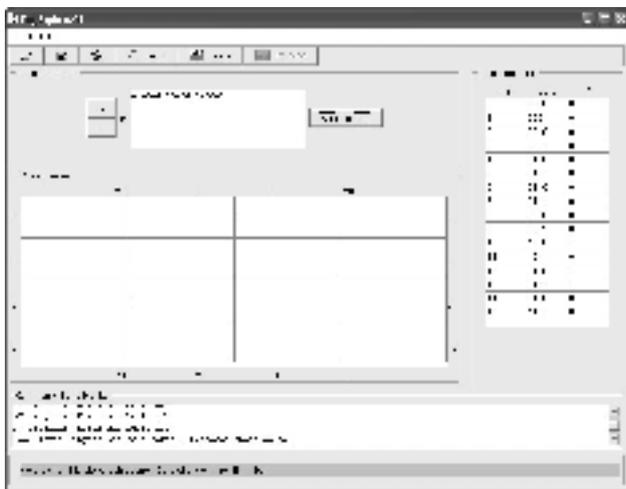


Abbildung 5: *DigiExplo*

DigiExplo ist eine Exploration aus dem Bereich der Elektrotechnik, genauer gesagt aus der Digitaltechnik. Lernende können darin boolesche Funktionen (als Abstraktion digitaler Schaltungen) mit bis zu vier Variablen in drei verschiedenen Sichten (Funktionstabelle, Funktionsterm in disjunktiver Normalform, KV-Diagramm) explorieren (vgl. Abb. 5). Alle drei Sichten sind interaktiv und ermöglichen es dem Lernenden, das Explorationsmodell zu gestalten bzw. zu

modifizieren. Jede Änderung innerhalb einer Sicht wirkt sich unmittelbar auf die anderen Sichten aus. Dadurch und durch die Realisierung von Stornierungsmöglichkeiten (Undo) und durch die Ermöglichung des Setzens von Freezing-Points (s. Button-Panel in Abb. 5) werden experimentierende Interaktionsformen angeregt und gut unterstützt. Eine Simulation wurde nicht realisiert. Eine Repräsentation des Explorationsmodells als Schaltbild durch einfache Gatter, das Anlegen von 0 bzw. 1 an dessen Eingänge, die Simulation

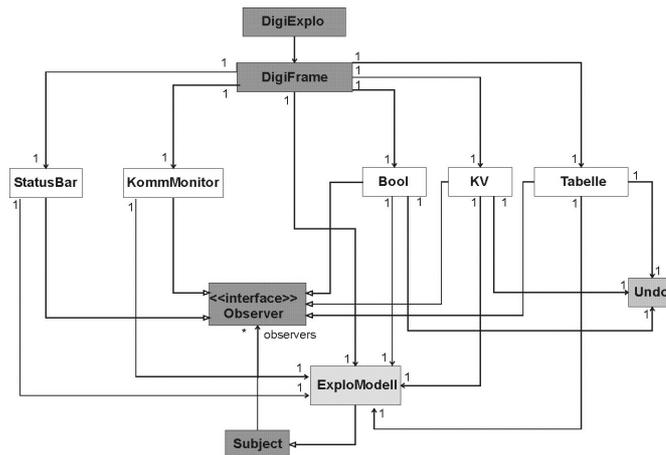


Abbildung 6: Anpassung der Architektur für *DigiExplo* [Ho03, 82]

mehrerer Zeittakte und die Beobachtung der Ausgänge wären aber prinzipiell leicht möglich. Abb. 6 zeigt, wie die Architektur (vgl. Abb. 4) für die Fallstudie präzisiert wurde. Die Klassen *Bool*, *KV* und *Tabelle* repräsentieren die konkreten Sichten, der Kommunikationsmonitor (*KommMonitor*) repräsentiert den Logbereich. Das Explorationsmodell (hier: boolesche Funktion) wurde realisiert als Vektor der Komponenten in disjunktiver Normalform.

4.2 Fallstudie *LangExplo*



Abbildung 7: *LangExplo*

Während für eher technisch orientierte Bereiche (wie z.B. 4.1) interaktive Lernmedien nicht unüblich sind, ist ein solcher Zugang in geistes- oder sprachwissenschaftlichen Bereichen noch eher selten. Aus diesem Grunde wurde als zweite Fallstudie ein konkretes Problem aus der englischen Grammatik gewählt, um eine Umsetzbarkeit auch für einen solchen Bereich zu belegen. Ein für Lernanfänger schwieriges Problem ist die korrekte Verwen-

dung der Präpositionen *is on* (ist auf) bzw. *is in* (ist in), z.B. „The pen is in the picture.“, „The pen is on the picture.“. Lernende können die korrekte Verwendung in drei verschiedenen Sichten (deutscher Satz, englischer Satz, grafische Darstellung der Semantik) am Beispiel der Lagebeziehung von Stift und Kiste explorieren. Ferner besteht die Möglichkeit zur Sprachausgabe des deutschen bzw. englischen Satzes. Unterstützt werden die Lagebeziehungen „Kiste auf Stift“, „Stift auf Kiste“, „Stift in Kiste“, „Stift oder Kiste in eines der neun Gitterfelder“ (vgl. Abb. 7). In allen drei Sichten können Lernende gestalten (Verknüpfung von in

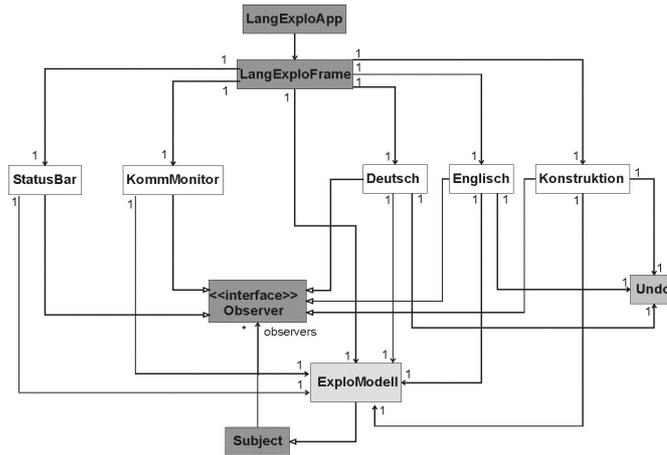


Abbildung 8: Anpassung der Architektur für *LangExplo* [Ho03, 71]

Combo-Boxen bereitgestellten Satzfragmenten (Subjekt – Prädikat und Präposition – Ortsangabe) in der deutschen bzw. englischen Darstellung; Modellierung eines Sachverhaltes durch direkte Interaktion mit der grafischen Darstellung). Durch die Bereitstellung der Satzfragmente in Combo-Boxen wurden die Modellierungsmöglichkeiten des Lernenden bewusst eingeschränkt. Möglich, aber auch erheblich aufwendiger, wäre hier aber auch eine Freitexteingabe für einen definierten Bereich, die dann analysiert und interpretiert werden müsste. Es besteht aber die Gefahr, dass dadurch vom Entwickler nicht vorhergesehene Verknüpfungen eingegeben werden, für die es keine vorbereitete Repräsentation in der grafischen Darstellung gibt. Abb. 8 zeigt die angepasste Architektur. Das Explorationsmodell wurde hier als Vektor der englischen Satzfragmente realisiert. Weitere technische Details zu den Fallstudien liefert [Ho03, 67ff.].

5 Fazit und Ausblick

Dargestellt wurde die Vorgehensweise bei der Entwicklung von Software zur Exploration im Bildungskontext als deren Ergebnis eine Grobarchitektur für solche Lehr-Lern-Systeme entstand. Durch die verzahnte Entwicklung und Anwendung im Zusammenhang mit zwei Fallstudien wurde die Tragfähigkeit des Konzepts zunächst exemplarisch belegt. Hier ist noch weitere empirische Absicherung erforderlich. Gezeigt wurde, wie die zuvor begründeten Anforderungen an die Anwendung bzw. an die zugrunde liegende Architektur umgesetzt wurden. Durch die Wahl von sehr unterschiedlichen Lerngebieten für die Fallstudien konnten ferner Belege für eine, wenn auch eingeschränkte, Domänen-unabhängigkeit des Konzepts geliefert werden. Voraussetzung für die Anwendbarkeit des Konzepts ist ein formalisierbarer Explorationsgegenstand, dessen zu explorierende Eigenschaften formalen, logischen Regeln genügen müssen. Die Eignung des Konzepts für Gegenstände, die diesen Voraussetzungen nicht genügen (z.B. aus den Sozialwissenschaften oder der Theologie) ist daher fraglich.

Im Weiteren müssen die Architektur und die mit ihr verbundene Vorgehensweise zur Erstellung von Software zur Exploration weiter verfeinert werden, z.B. im Hinblick auf die Unterstützung mehrerer Explorationsmodelle pro Anwendung. Ferner wird eine Verknüpfung mit standardisierten Architekturmodellen (Drei- bzw. Fünf-Schichtenarchitektur) angestrebt.

Ziel der Entwicklung eines Architekturkonzepts war es ferner, die Gestaltung von Software zur Exploration zu rationalisieren und damit einen Beitrag zur Verbreitung solcher Lernangebote zu leisten. Offen ist in diesem Zusammenhang noch eine systematische Evaluation (z.B. mit Lehrenden), in der untersucht wird, inwieweit durch das vorgestellte Konzept tatsächlich die vermutete Rationalisierung erfolgt.

Literaturverzeichnis

[An01] Anderson, J. R.: Kognitive Psychologie. Spektrum, Berlin, 2001.

- [BS02] Brinda, T.; Schubert, S. E.: Learning aids and learners' activities in the field of object-oriented modelling. In (Passey, D.; Kendall, M. Hrsg.): *TelE-Learning. The Challenge for the Third Millennium*. Kluwer Academic Publisher, Boston, 2002; S. 37-44.
- [Br04] Brinda, T.: *Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sek. II*. Dissertation, FB Elektrotechnik und Informatik, Universität Siegen, 2004.
- [Ga95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Design Patterns – Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, Inc., Reading, 1995.
- [Ho03] Hoffmann, A.: *Theoretisch begründete Vorgehensweise bei der Entwicklung von Software zur Exploration im Bildungskontext*. Diplomarbeit, FB Elektrotechnik und Informatik, Universität Siegen, 2003.
- [Ke01] Kerres, M.: *Multimediale und telemediale Lernumgebungen – Konzeption und Entwicklung*, 2. vollst. überarb. Aufl., Oldenbourg-Verlag, München, 2001.
- [Ne88] Neber, H.: *Elemente entdeckenden Lernens: Konzeptionelle Aspekte und deren Realisierung*. Zeitschrift für Heilpädagogik, Beiheft 14, 1988, S. 59-65.
- [No02] Nowaczyk, O.: *Explorationen: Lernen durch Konstruktion*. In (Rinn, U.; Wedekind, J. Hrsg.): *Referenzmodelle netzbasierten Lehrens und Lernens* Bd. 19. Waxmann Verlag, Münster, 2002, S. 137-155
- [Pa94] Paul, H.: *Exploratives Agieren: ein Beitrag zur ergonomischen Gestaltung interaktiver Systeme*. Lang (Europäische Hochschulschriften: Reihe 41, Informatik, Bd. 16), Frankfurt am Main, 1994.
- [SM81] Straka, G. A.; Macke, G.: *Lehren und Lernen in der Schule*. Kohlhammer, Stuttgart, 1981.
- [St95] Steindorf, G.: *Grundbegriffe des Lehrens und Lernens*. Klinkhardt, Bad Heilbrunn, 1995.
- [St99] Steinkamp, P.: *Informatik-Experimente im Schullabor*, Diplomarbeit, FB Informatik, Universität Dortmund, 1999.
- [Vy78] Vygotsky, L. S.: *Mind in Society*. Harvard University Press, London 1978.