T U M

I N S T I T U T F Ü R I N F O R M A T I K

Beiträge zum
8. Arbeitstreffen der
GI–Fachgruppe 1.1.5/7.0.1
„Intelligente Lehr–/Lernsysteme“

Christian Herzog
(Herausgeber)

TUM–I9736
August 97

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-08-I9736-150/1.-FI
Alle Rechte vorbehalten
Nachdruck auch auszugsweise verboten

©1997

Druck: Institut für Informatik der
Technischen Universität München

Beiträge zum

8. Arbeitstreffen der GI–Fachgruppe 1.1.5/7.0.1 „Intelligente Lehr–/Lernsysteme“

18. und 19. September 1997

Duisburg

herausgegeben von

Christian Herzog

Inhaltsverzeichnis

Vorwort **iii**

IDEALS MTS – Ein modulares Training System für die Zukunft **1**
Frank Graf und Michael Schnaider, Darmstadt

HyperCons – Eine hypermediale konstruktivistische Lernumgebung für Schulen **13**
Peter Hubwieser, Johann Schlichter und Birgit Mair, München

Dynamic Courseware Generation on the WWW **24**
Julita Vassileva und Ralph Deters, München

D3–WWW–Trainer – Entwicklung einer Oberfläche für die Netzanwendung **31**
Sven Faulhaber und Bettina Reinhardt, Würzburg

Kooperative Lernumgebungen für kleine und mittlere Unternehmen auf Basis eines intelligenten Lernsystems **41**
Matthias Kloth, Dortmund

Ein Konzept zur Modellierung von Handlungszielstrukturen in intelligenten Lehrsystemen **53**
Eva Kluge, Berlin

Kognitive und motivationale Gesichtspunkte in einer Lehrstrategie **65**
Andreas Harrer, München

Wissensbasierte Integration der Komponenten eines intelligenten Lehrsystems **77**
Meike Gonschorek, München

Adapting Knowledge Structures in Dynamic Domains **89**
Dietrich Albert und Rainer Kaluscha, Graz

Eine generische Erklärungskomponente für Autorensysteme **101**
Ute Trapp und Markus Lusti, Basel

Abstraktionsbasierte Erklärungen für Prognose und Diagnose mit Bayes–Netzen **111**
Andreas Lüdtke, Oldenburg

Die Integration von dynamischen und statischen Analysemethoden in einer intelligenten Lern– und Problemlöseumgebung **122**
Janine Willms, Hermann Göhler und Claus Möbus, Oldenburg

ILSE: Ein Interaktives Lehr–/Lernsystem für Spätertaubte **133**
Claudia Wans, Duisburg

Vorwort

Intelligente Lehr-/Lernsysteme sind Systeme, die explizit repräsentiertes Wissen für die Wissenskommunikation einsetzen. Sie soll dem Wissensstand und der Motivation einzelner oder mehrerer Lerner angepaßt werden, um eine hohe Benutzerakzeptanz zu sichern. Dies ist der Wortlaut, mit dem die Fachgruppe 1.1.5/7.0.1 der Gesellschaft für Informatik in ihrer Satzung den Gegenstand ihrer Forschungsarbeiten beschreibt, und dies bietet sich auch als Motto für das achte Arbeitstreffen dieser Fachgruppe an, das am 18. und 19. September 1997 an der Gerhard–Mercator–Universität Gesamthochschule Duisburg stattfindet. Als Arbeitstreffen soll es ein Forum darstellen, um über laufende und abgeschlossene Forschungsprojekte zu berichten und deren Ergebnisse intensiv im Kreis der Kolleginnen und Kollegen zu diskutieren. Die zeitliche und räumliche Nähe zur 7. GI Fachtagung *Informatik und Schule – Infos '97* soll überdies einen Bezug zu einem wichtigen Anwendungsfeld herstellen.

Das Programmkomitee hat aus den eingereichten Beiträgen 13 Arbeiten ausgewählt, die einen guten Überblick über aktuelle Aktivitäten im deutschsprachigen Raum in Forschung und Entwicklung auf dem Gebiet intelligenter Lehr– und Lernsysteme bieten. Dabei werden bestehende und praktisch eingesetzte Systeme ebenso vorgestellt wie Formen der Wissensrepräsentation und –erklärung, Diagnoseverfahren ebenso wie Versuche, bestehende Systeme durch zusätzliches Wissen adaptiver zu gestalten. Schwerpunkte haben sich herausgebildet bei Arbeiten zur Gestaltung der Lernerführung sowie zur intelligenten Unterstützung bei der Benutzung des *World Wide Web*.

An dieser Stelle möchte ich mich herzlich bei Katrin Gaßner und Heinz Ulrich Hoppe bedanken, die die lokale Organisation in Duisburg übernommen haben, bei den Mitgliedern des Programmkomitees für die zügige Beurteilung der eingereichten Arbeiten, bei allen Autorinnen und Autoren für die sorgfältig erstellten und trotz des engen Zeitrahmens pünktlich abgelieferten Endfassungen ihrer Beiträge, sowie bei meiner Kollegin Helma Petra Schneider für die freundliche Hilfe bei der Zusammenstellung dieses Tagungsbandes.

München, im August 1997

Christian Herzog

Technische Universität München, Fakultät für Informatik, D–80290 München
E–mail: Christian.Herzog@informatik.tu-muenchen.de
WWW: http://www.informatik.tu-muenchen.de/~herzog/

Programmkomitee: Katrin Gaßner, Universität Duisburg
Rul Gunzenhäuser, Universität Stuttgart
Christian Herzog, TU München (Vorsitz)
Heinz Ulrich Hoppe, Universität Duisburg
Claus Möbus, Universität Oldenburg
Olaf Schröder, Debis Hamburg

IDEALS MTS - Ein modulares Training System für die Zukunft

Frank Graf, (e-mail: graf@igd.fhg.de)
Michael Schnaider, (e-mail: schnaide@igd.fhg.de)

Technische Hochschule Darmstadt,
Fachbereich Informatik,
FG Graphisch-Interaktive Systeme
Rundeturmstraße 6, 64283 Darmstadt

1 Einleitung

Mit der zunehmenden Leistungsfähigkeit heutiger Informations- und Kommunikationstechnologie (I&K Technologie) haben sich neue Einsatzmöglichkeiten in der Aus- und Weiterbildung ergeben. Gleichzeitig erfordern immer kürzer werdende Produkt- und Innovationszyklen eine ständige Erweiterung und Aktualisierung von Wissen.
In der universitären Aus- und Weiterbildung hat sich I&K Technologie bereits als unterstützendes Werkzeug zur Erstellung von Lehrmaterial etabliert. Mit der zunehmenden Verfügbarkeit des WWW rückt nun auch dessen Nutzung zur Distribution von Lehr- und Lernmaterial verstärkt in den Vordergrund. Auch die Industrie hat die Notwendigkeiten und Möglichkeiten einer fortwährenden Weiterbildung als ein wichtiges Thema identifiziert. Entsprechend werden flexiblere Strukturen in der Aus- und Weiterbildung gefordert, die sich den veränderten Erfordernissen anpassen.
Die Zielsetzungen, die mit dem Einsatz von I&K Technologie im Bereich der Aus- und Weiterbildung verfolgt werden, stimmen trotz unterschiedlicher Motivationen überein. Zentrale Forderungen sind eine Steigerung von Effizienz in der Lehre und eine Verbesserung der Effizienz beim Lernen. Neben einer Unterstützung von eigenständigem Lernen soll gleichzeitig eine Verminderung von zeitlichen und räumlichen Restriktionen erreicht und ein breiterer Personenkreis angesprochen werden.
Für die Umsetzung der genannten Zielsetzungen sind im Rahmen der Europäischen Projekte DEDICATED und IDEALS unter maßgeblicher Beteiligung von GRIS (Fachgebiet Graphisch-Interaktive Systeme) Konzepte entwickelt worden. Dabei wird versucht, aus den bisher dominierenden kurs-orientierten Paradigmen auszubrechen, und zu einer vollständig modularen Struktur von Lehr- und Lernmaterial überzugehen. Diese Konzepte und ihre Umsetzung innerhalb des Projektes IDEALS werden im folgenden dargestellt.

2 Konzepte

2.1 Strikte Modularisierung

Für ein Lehr- und Lernsystem, das den zukünftigen Anforderungen gerecht wird, ist die Modularität des Lernmaterials eine Grundvoraussetzung.
Betrachtet man heutige CBT-Systeme oder das WWW, das in immer stärkerem Ausmaß für Lernzwecke verwendet wird, ist festzustellen, daß diese Lehr- und Lernsysteme konzeptionell auf festen, monolithischen Kursen basieren. In traditionellen Lernsystemen wie auch im WWW ist nur eine horizontale Modularisierung zu finden, bei der ein Kurs in mehrere Kursteile aufgeteilt wird. Diese Kursteile (Module) können separat gespeichert werden. Dabei ist folgendes festzustellen:

- Kursteile verweisen auf- und benötigen einander. Es gibt keine separaten, übergeordneten Strukturen, sondern nur feste Verweise und inhärente Verknüpfungen.
- Einzelne Kursteile sind relativ umfangreich. Die obere Größe hängt eher von technischen Faktoren (z.B. Diskettengröße) ab als von inhaltlichen Aspekten.

- Jeder Kursteil ist ein monolithischer Block und enthält sowohl das Basismaterial, die eigentlichen Träger der Lerninhalte (Texte, Bilder, usw.), als auch die Layout- und Interaktionsdefinitionen und die globale Kurssteuerung.
- Wiederverwendung und Austausch von Teilen eines Kurses (z.B. zur Aktualisierung und Anpassung) sind nur eingeschränkt möglich. Die Wiederverwendung von einzelnen Basismaterialien ist aufgrund der monolithischen Ablage nicht möglich.

Verfolgt man den Gedanken der Modularisierung konsequent weiter, dann gibt es keine Kurse im eigentlichen Sinne mehr, sondern einen Vorrat von Modulen, aus dem bei Bedarf optimale Module ausgewählt und zu einem Kurs zusammengefügt werden.

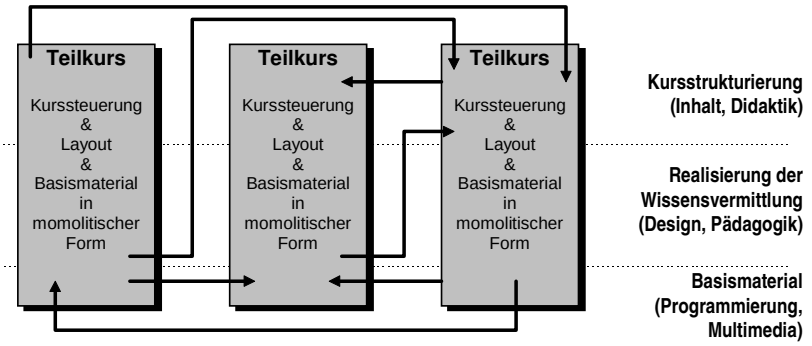


Abbildung 1: Horizontale Modularisierung

Um den Modularitätsforderungen gerecht zu werden muß jedes Modul autark, d.h. vollständig und abgeschlossen, sein. Es muß auch als eigenständige Einheit abgelegt werden können. Module dürfen zwar hierarchisch sein, d. h. ein Modul darf aus anderen Modulen zusammengesetzt sein, aber es gibt drei wichtige Einschränkungen, die man in dieser Konsequenz in heutigen Lernsystemen nicht findet:

(1) Module können andere Module verwenden, aber sie können diese niemals enthalten. Statt dessen enthalten sie immer nur Verweise (Referenzen) auf andere Module (vergl. 2.2). Das IDEALS MTS verfügt daher über einen Mechanismus, mit dem ein Autor inhaltsgesteuert Module referenzieren kann, ohne wissen zu müssen, welche Modul wirklich angesprochen werden.

(2) Module die andere Module verwenden, dürfen keinerlei Annahmen über die Interna der untergeordneten Module machen (Blackbox-Konzept). Statt dessen gibt es eine abstrakte Schnittstellendefinition für Module. Alle Module müssen genau diese Schnittstelle realisieren.

(3) Module dürfen, abgesehen von den Möglichkeiten, die die Schnittstelle bietet, keinerlei Annahmen über die Interna der übergeordneten Module machen.

Mit der Umsetzung des Modulgedanken ergeben sich folgende Eigenschaften:

- Alle Module sind vollständig und deshalb auch eigenständig verwendbar. Teile eines Kurses lassen sich wiederverwenden, ohne daß nicht-benötigte Teile des Ursprungskurs erst entfernen werden müssen.
- In einem Kurs lassen sich Module ohne nennenswerten Aufwand austauschen. Damit läßt sich zum einen ein Kurs optimal an den Benutzer anpassen. Zum anderen kann ein Kurs einfach aktualisiert werden.
- Einzelne Module lassen sich in anderen Kursen wiederverwenden. Das führt zu einer Reduktion der Gesamtkosten pro Kurs. Wiederverwendung ist insbesondere dann interessant für einen Autor, wenn er auf diese Weise auf Material von einem Experten auf diesem Gebiet zugreifen kann.

Eine durchgängige Modularisierung hat allerdings auch Konsequenzen für das Erstellen von Kursen. Ein Autor muß die Modularität bereits in der Designphase und schließlich bei der Produktion eines „Kurses“ beachten. Das heißt insbesondere, daß ein Autor nicht mehr in ganzen Kursen denken darf, sondern in einzelnen, gut strukturierten (gekapselten) und wiederverwendbaren Modulen.

Diese Forderungen erscheinen selbstverständlich und auch einfach umsetzbar. Unsere Erfahrungen haben jedoch gezeigt, daß das Erstellen von modularen Kursen dem Autor ein erhebliches Maß an Umdenken abverlangt. Mit der Idee des Erstellens von wiederverwendbaren Einheiten sind die meisten Autoren bereits vertraut. In vielen Fällen wird dies auch (im Rahmen der Möglichkeiten) praktiziert. Womit Autoren zumeist Schwierigkeiten haben, ist die Einsicht, daß das Konzept eines modularen Kurses mehr beinhaltet, als bei der Kurszusammenstellung aus dem Vorrat der aktuell vorhandenen Module die bestgeeigneten Module herauszusuchen und fest zu einem Kurs zusammenzubinden. Mit IDEALS MTS lassen sich auch derartige herkömmliche, vorkonfigurierte Kurse, bei denen die Auswahl der Kursmodule durch den Lehrer vorgegeben wird, erstellen. Derartige Kurse besitzen durchaus eine Daseinsberechtigung, denn sie eignen sich insbesondere dafür, einer (homogenen) Gruppe von Personen dasselbe Wissen zu vermitteln. Die Wieder- und Mehrfachverwendung von Modulen führt zu einer Reduktion des anfallend Aufwandes.
Ein modulares Kurskonzept ermöglicht darüber hinaus auch Kurse, die sich automatisch an den Benutzer anpassen. Dies wird erreicht, indem Kurse dynamisch, d.h. erst während der Kurslaufzeit, zusammengestellt werden. So können die spezifischen Bedürfnisse und Vorlieben eines Benutzers (z.B. Vorwissen, Kurssprache) bei der Auswahl und Zusammenstellung der Module berücksichtigt werden. Der Benutzer erhält seinen optimalen Kurs. Wegen der inhärenten Dynamik derartiger Kurse ist ihr Verlauf aber nicht vorhersagbar. Diese Form von Kursen eignet sich besonders für eigenständiges Lernen (learning on demand). Für einen Autor stellt sich hier allerdings das Problem, daß er einen Kursverlauf spezifizieren muß, ohne zu wissen, wie die referenzierten Module ihre Aufgaben im Detail erfüllen. Er muß sich darauf beschränken die Aufgaben zu verteilen und zu koordinieren.
Konsequente horizontale Modularisierung erlaubt Austausch, Wiederverwendung und Mehrfachverwendung von Modulen. Damit stellt sie die Grundlage für die effiziente Erstellung und Verwendung von digitalem Lernmaterial dar.
Der hier verwendete Begriff Modul spiegelt sehr stark die Betrachtungsweise eines Programmierers wider. Aus Anwendersicht sind die Module das Lernmaterial. Das führt zu einen zweiten Aspekt, dem bei der Modularisierung Beachtung geschenkt werden soll. Dazu wird zunächst betrachtet, über welche Expertise ein Autor idealerweise verfügen sollte:

- Er sollte Inhaltsexperte sein. Er ist Experte auf dem Fachgebiet, das vermittelt werden soll und verfügt über didaktische Fähigkeiten. Seine Hauptaufgabe ist es festzulegen, welche Inhalte in einem Kurs vermittelt werden sollen und in welcher Form dies geschehen soll. Dazu unterteilt er das Thema in einzelne Lernziele und organisiert diese geeignet.
- Er sollte Experte für die Umsetzung von Lernzielen sein. Hier ist die Aufgabe, die vorgegeben Lernziele in konkrete Lernschritte umzusetzen. Er erstellt die Layout- und Interaktionsdefinitionen für die einzelnen Lernschritte. Seine Expertise liegt hauptsächlich in den Bereichen Pädagogik und Design.
- Er sollte Materialexperte sein. Als solcher verfügt er über das nötige (technische) Wissen und die Fertigkeiten zur Erstellung von (elementarem) Kursmaterial, das die Inhalte enthält. Normalerweise handelt es sich bei diesen Experten um Programmierer für komplexe, interaktive Materialien oder Media-Experten wie Grafiker, Videotechniker und Computeranimateure für elementares Multimediamaterial.

Ein fertiger Kurs ergibt sich erst aus dem Zusammenspiel aller drei Experten. Da das Arbeitsgebiet dieser Experten jeweils nur einen speziellen Teilbereich der Kurserstellung abdeckt, und jeder dieser Experten auch eine spezifische Sicht des Kurses hat, muß ein Lernsystem, den Prozeß der Kurserstellung richtig repräsentieren und geeignet unterstützen und dieses auch in seinem Modulkonzept widerspiegeln. Daher muß es neben der horizontalen Modularisierung auch eine vertikale Modularisierung geben, die diesen unterschiedlichen Experten- und Kursebenen entspricht. Konkret sieht das modulare Kurskonzept von GRIS daher auch drei Kategorien von Modulen vor. Jeder Kategorie von Experten entspricht eine eigene Mo-

dulklasse, die dessen Arbeitsbereich und Sichtweise widerspiegelt. Somit arbeitet jeder Experte auf der für ihn adäquaten Abstraktionsebene in einer natürlichen (intuitiven) Art und Weise. Durch geeignete Autorenwerkzeuge ist in diesem Bereich die Effizienz noch zu verbessern (vergl. 3.2).

2.2 Referenzierung und Benutzerprofilierung

Um eine durchgängig modulare und zugleich flexible Struktur von Lehr- und Lernmaterial auf allen Ebenen unterstützen zu können, ist es essentiell, daß Lernmaterial nicht direkt eingebunden, sondern nur indirekt durch Verweise genutzt wird. Im folgenden wird vorgestellt, wie sich ein derartiger Referenzierungsmechanismus mittels Attributieren mit klassifizierbaren Merkmalen realisieren läßt. Es wird auch auf das verwandte Thema Benutzerprofilierung eingegangen.

Selbstbeschreibung

Unabhängig von der Zugehörigkeit zu einer der drei o.g. Modulklassen besitzt jedes Lernobjekt eine Selbstbeschreibung. Diese ist eine Menge von Attributen, die unterschiedliche Eigenschaften der Lernobjekte beschreibt. Lernobjekte besitzen die üblichen technisch-organisatorischen Attribute wie Name, Erstellungsdatum, Sprache, Größe, Typ, Autor sowie weitere von der Modulklasse abhängige Attribute (erwartete Ausführungszeit, Datenumfang in Byte). Darüber hinaus enthält die Selbstbeschreibung jedes Lernobjektes Attribute, die eine einheitliche Zuordnung nach inhaltlichen oder pädagogischen/didaktischen Kriterien ermöglichen. Das dabei zugrunde liegende Klassifikationsschema unterscheidet zur Zeit vier inhaltliche Merkmale:

Eigenschaft	Attributname	Beispiel
(wissenschaftliche) Disziplin	context	computer graphics
Sichtweise	aspect	algorithms (of)
Schlüsselbegriff(e)	keyword	Weiler-Atherton Algorithm
Themenbereich	subject	clipping

Tabelle 1 Inhaltliche Klassifikationsmerkmale

Zwischen den in Tabelle 1 genannten Attributen gibt es Abhängigkeiten. Dem Attribut 'Disziplin' sind die Attribute 'Themenbereich' und 'Schlüsselbegriff' hierarchisch untergeordnet. Es gibt zu jedem Attributwert 'Disziplin' einen eigenen Satz von zulässigen Werten für 'Themenbereich' und 'Schlüsselbegriff'. Dabei ist es erlaubt, daß gleiche Attributwerte in mehreren Sätzen für unterschiedliche Disziplinen vorkommen. Die Notwendigkeit für eine solche Anordnung ergibt sich aus der Tatsache, daß gleiche Begriffe in unterschiedlichen Disziplinen unterschiedliche Bedeutung besitzen können. Das Attribut 'Sichtweise' ist dem Attribut Disziplin hierarchisch gleichgestellt. Über dieses Attribut wird eine bestimmte (inhaltliche) Sicht auf ein Lernmaterial beschrieben.
Die Menge der verfügbaren Werte für jedes Attribut bildet den Definitionsbereich (oder Wortschatz) für die Beschreibung von Lernmaterial. Um eine Eindeutigkeit der Bezeichnungen zu erreichen, ist es notwendig, die Menge der zulässigen Werte zu begrenzen und nur kontrolliert zu erweitern, z.B. durch den Einsatz von standardisierten Wörterbüchern. Die Information in dem o.g. Viertupel stellt eine Charakterisierung der in einem Modul enthaltenen Information bzw. des enthaltenen Wissens dar. Neben der Nutzung als Zugriffsschlüssel bildet es auch die Grundlage, die den Aufbau eines Wissensmodells in der Benutzerprofilierung ermöglichen soll.

Referenzierung

Bei der Referenzierung von Modulen werden zwei unterschiedliche Ansätze unterstützt. Generell werden die folgenden zwei Arten von Referenzen unterschieden:

- Eine Direkte Referenz verweist über den eindeutigen Namen auf ein Modul. Das Modul wird statisch referenziert.
- Eine Virtuelle Referenz adressiert i.A. nicht ein spezielles Modul, sondern eine Menge von "gleichartigen" Modulen. Mit einer Virtuellen Referenz wird ein Anforderungsprofil definiert, dem ein Modul genügen muß. Bei der Auflösung einer Virtuellen Referenz wird innerhalb der Menge der verfügbaren Module dasjenige identifiziert, dessen Selbstbeschreibung dem definierten Anforderungsprofil am besten entspricht. Im Gegensatz zu den Direkten Referenzen wird mit einer Virtuelle Referenz dynamisch referenziert (vergleichbar mit spätem Binden), d.h. die Virtuelle Referenz wird erst zur Laufzeit aufgelöst.

Benutzerprofilierung

Ein weiterer wichtiger Aspekt für effizientes Lernen ist die Anpassungsfähigkeit des präsentierten Lehr- und Lernmaterials an die Bedürfnisse des Lernenden. Anpassung setzt voraus, daß Informationen über den aktuellen Wissenstand des Lernenden vorliegen. Zu diesem Zweck ist eine Benutzerprofilierung vorzusehen, in der entsprechende Informationen über Lernfortschritt und -ergebnisse protokolliert werden. Die gesammelte Information steht dann zur Abfrage und für Konsolidierungsprozessen zur Verfügung. Durch Anfragen an das Profil kann unter Berücksichtigung des Lernstatus steuernd auf den Verlauf und die Auswahl von neuen oder vertiefenden Lehr- und Lernmodulen eingewirkt werden. Somit (in Anlehnung an die Ansätze des programmierten Lernens) kann bereits rudimentäre Adaption an den Benutzer realisiert werden.
Ein langfristiges Ziel ist aus der gesammelten Information ein kognitives Modell für einen Benutzer zu generieren, mit dessen Hilfe die Qualität der Adapation erheblich verbessert werden kann. Als Ausgangsbasis kann hier die Information, die aus der inhaltlich-didaktischen Indexierung der Modulen (der Selbstbeschreibung) abgeleitet werden kann, dienen.

3 Realisierung in IDEALS

Die oben dargelegten Überlegungen bildeten die Grundlage für die Projekte DEDICATED und IDEALS, an denen GRIS beteiligt war. Vergleichbare Ansätze findet man z.B. in OLA, ARISTO und ARIADNE. In diesem Kapitel wird nun beschrieben, wie diese Überlegungen im MTS, dem Modularen Training System von IDEALS, umgesetzt werden.

3.1 IDEALS WWW basierte System Architektur

Während die Entwicklung in DEDICATED noch auf eigenentwickelter, proprietärer Client-Server Technologie aufbaute, hat sich dies in IDEALS mit dem Aufkommen des WWWs verändert. Ein wichtiges Entwurfskriterium ist die Integration in das WWW und die Nutzung von existierenden Standards. Der Vorteil der Verwendung von HTTP (HyperText Transport Protocol) liegt insbesondere in der plattformübergreifenden Verbreitung sowie in der Verfügbarkeit der für IDEALS notwendi-

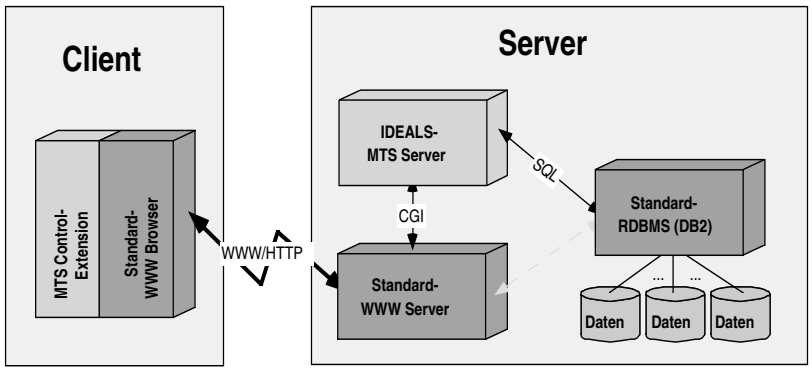


Abbildung 2: Grobschema der Systemarchitektur

gen Basistechnologien wie Integration von Telekommunikation und unterschiedlichsten Multimedia-Formaten. Gleichzeitig ist mit dem Einsatz dieses offenen Standards auch die Integration zukünftiger Erweiterungen von HTTP und HTML (HyperText Markup Language) in IDEALS sicher gestellt. Aufgrund dieses Gedankens wurde die IDEALS Software als eine Erweiterung eines WWW-Server konzipiert und realisiert. Das IDEALS MTS ist dabei als eigenständiger Serverprozeß realisiert, der eng an einen WWW-Server gekoppelt ist. Abbildung 2 zeigt ein Grobschema der Systemarchitektur.

3.2 Course Nodes, LFUs und LMOs

Wie bereits eingehend dargelegt wurde sind eine strukturelle Modularisierung und ein leistungsfähiger Referenzierungsmechanismus Kernpunkte des IDEALS MTS. Um diesen Anforderungen zu entsprechen, wurde ein Schichtenmodell für Lehrmaterial entwickelt. Dieses Schichtenmodell umfaßt drei Schichten und grenzt dabei klar Lerninformation, Informationspräsentation und didaktisch-pädagogische Aspekte voneinander ab. Abbildung 3 zeigt die Schichten und die zugehörigen Modultypen. Im folgenden werden die einzelnen Modultypen detaillierter vorgestellt:

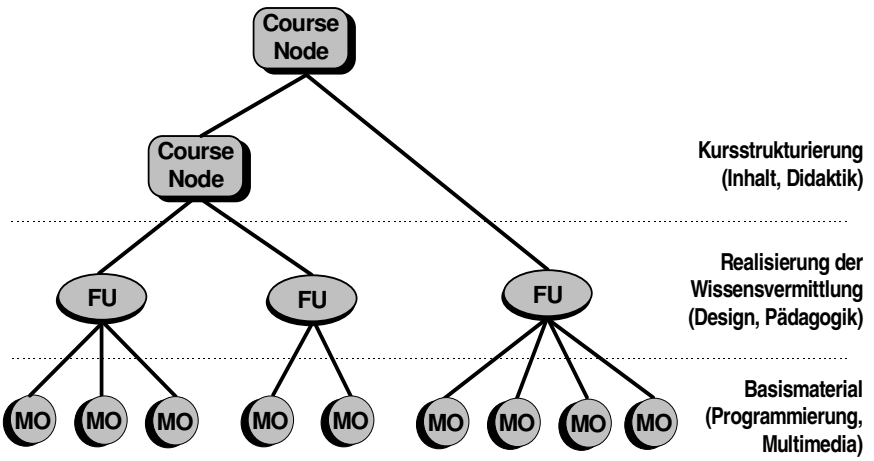


Abbildung 3: Kursmodule und Expertenebenen

Material Object (MO)

MOs sind die atomaren, multimedialen Bausteine für einen Kurs. MOs umfassen zum einen die eigentlichen Träger des Lerninhaltes. Dazu gehören alle Standard-Multimediaformate (wie GIF und TIFF für Bitmaps, WAV für Audio). Aber MOs können durchaus auch komplexe, selbst entwickelte Programmfragmente oder gar eigenständige Anwendungen sein, wie z.B. als Java-Applets realisierte Simulationen oder Microworlds. Zum anderen umfassen MOs auch die vom System für die Realisierung einer Benutzerschnittstelle bereitgestellten Interaktionselemente.
Es war eines der Designkriterien von IDEALS MTS in diesem Bereich keine eigene Formate oder Werkzeuge zu entwickeln, sondern vielmehr die für das WWW verfügbaren vorhandenen Formate und Werkzeuge zu nutzen, soweit deren Ergebnisse auf allen unterstützten Plattformen verwendbar sind. Während es von der technischen Seite hier keinerlei Einschränkung gibt, muß ein Materialexperte beim Erstellen von Material natürlich darauf achten, das der Modularitätsgedanke nicht verletzt wird. Aus der Forderung nach der Vollständigkeit von Modulen ergibt sich, daß mehrere inhaltlich zusammengehörige Elemente (z.B. ein Bild mit zugehörigem Erläuterungstext) auch zu einem MO zusammengefaßt werden müssen. Daher verfügt MTS neben primitiven MOs, die genau einem der oben beschriebenen Elementtypen entsprechen, auch über Compound-MOs.

Function Unit (FU)

FUs fügen MOs zu Einheiten zusammen, die bestimmte Funktionen der Wissensvermittlung (wie Präsentation, Exploration oder Test) realisieren. Die FU regelt dabei das Zusammenspiel der in ihr verwendeten MOs untereinander und mit dem Benutzer. Eine FU definiert für ein

spezifisches Wissensgebiet die Anordnung (Layout) und das Gesamtverhalten (Interaktion untereinander und mit den Benutzer) einer Gruppe von MOs, mit der die gewünschte Funktion zur Wissensvermittlung realisiert wird.
Als Ablage- und Beschreibungsformat für FUs wird HTML-I verwendet. HTML-I ist eine IDEALS-spezifische Erweiterung von HTML, bei der die URLs durch ein eigenes TAG ersetzt werden, das den IDEALS-eigenen Referenzenmechanismus unterstützt. Um FUs in einem WWW-Browser darstellen zu können, müssen diese vor dem Herunterladen von HTML-I nach HTML umgesetzt werden. Dies geschieht durch einen in den MTS-Server integrierten Konverter, der alle in einer FU enthaltenen IDEALS spezifischen TAGs sucht und die darin enthalten Referenzen in eine URL umsetzt.
Da reines HTML keinerlei Möglichkeiten bietet auf Benutzeraktionen zu reagieren, läßt sich damit kein interaktives Dokument erzeugen. Will man das Zusammenspiel zwischen mehreren MOs und Benutzerinteraktionen, die mehrere MOs betreffen, konzeptuell richtig, nämlich in der FU selbst, beschreiben, dann muß man eine HTML-Erweiterung verwenden. Aufgrund der weiten Verbreitung ist JavaScript als Erweiterung ausgewählt worden.
Auch bei den FUs wurde von Anfang an Wert darauf gelegt, auf einen existierenden Standard (HTML) aufzubauen. Somit lassen sich auch hier herkömmliche Werkzeuge wie HTML-Editoren zur Erstellung vom FUs verwenden. Da derartige Editoren allerdings das IDEALS-spezifische TAG nicht unterstützen und sich in der Regel auch nicht entsprechend erweitern lassen, erstellt man mit diesen Editoren eine HTML-Variante der FU. In einem zweiten Schritt wird die HTML-Variante durch das Einbringen der IDEALS-spezifischen TAGs in HTML-I umgewandelt.

Course Node (CN)

In der obersten Schicht werden die inhaltlichen und didaktischen Aspekte und Zielsetzungen abgebildet. Hier werden die einzelnen, zu einem Kontext gehörenden FUs zu Course Nodes zusammengefaßt. Neben dieser ebenfalls durch Referenzierung erfolgenden Zusammenfassung bestimmen die CNs insbesondere die Reihenfolge, in welcher FUs dem Lernenden präsentiert werden. D.h. sie bestimmen die Abfolge der einzelnen Lernschritte innerhalb eines Lernkontextes sowie Übergänge zu vertiefenden oder thematisch weiterführenden CNs. CNs werden in textueller Form als eine Art von Ablaufplan abgelegt. Hierzu wird die selbstentwickelte Beschreibungssprache CDL (Course Description Language) verwendet.
In IDEALS wurde von Beginn an Wert auf ein vollständig ins System integriertes graphisches Autorenwerkzeug gelegt, das es dem Inhaltsexperten erlaubt, CNs graphisch zu erstellen. Das graphische Autorenwerkzeug repräsentiert einen CN durch ein Flußdiagramm, das den Kursablauf darstellt. Der Benutzer kann interaktiv Elemente in das Flußdiagramm einfügen oder Elemente aus diesem entfernen. Desweiteren kann er bestehende Elemente verschieben oder spezifizieren. Die gesamten Interaktionen erfolgen dialog-gesteuert. Dies und der relative geringe Umfang der Sprachmittel ist insbesondere deshalb ein wichtiger Punkt, da ein Inhaltsexperte normalerweise über keinen Programmierhintergrund verfügt und er meistens auch dem Gebrauch einer textorientierten Programmierung ablehnend gegenüber steht.

3.3 Kursnavigation

Ein Problem mit konsequent modularen Lernmaterial ist die Realisierung der Benutzungsschnittstelle für die Kursnavigation. Da ein Autor nicht vorhersagen kann – es auch nicht können soll – wo und wie sein Modul eingesetzt wird, ist er auch nicht in der Lage, eine Benutzungsschnittstelle für die Kursnavigation zu definieren.
Die Kursnavigation wird aus den FUs herausgezogen und dem Benutzer als eigenständige Systemkomponente mit separater Benutzungsschnittstelle parallel zu der aktuellen FU zur Verfügung gestellt. Die Funktionalität zur Kursnavigation wird durch ein Java-Applet, das

Control Panel, realisiert. Ein Autor kann innerhalb einer FU mittels JavaScript auf das Control Panel zugreifen. Daraus ergeben sich folgende Vorteile:

- Da die Systemkomponente in allen Kursen in derselben Form und mit derselben Funktionalität verfügbar ist, garantiert man ein einheitliches Look-and-Feel für alle IDEALS-Kurse und der Benutzer muß sich nicht mit jedem Kurs mit einer neuen Benutzungsschnittstelle vertraut machen.
- Als Systemkomponente hat das Control Panel Zugriff auf die interne Kursdarstellung des aktuellen Kurses und ist die einzige Instanz, die weiß, aus welchen Modulen sich ein Kurs zu Laufzeit zusammensetzt.
- Das Vorhandensein einer Systemkomponente bedeutet eine Arbeitsentlastung für den Autor. Es muß keine Benutzungsschnittstelle für die Navigation durch den Kurs mehr realisieren.
- In IDEALS kommunizieren FUs niemals direkt mit anderen FUs. Die Kommunikation mit Course Nodes erfolgt durch die Verwendung der Schnittstelle, die ihnen von dem Control Panel zur Verfügung gestellt wird. Die Existenz einer standardisierten Schnittstelle garantiert auch die Austauschbarkeit aller Module.

3.4 Virtuelle Referenzen und Mapping

Lernmaterial wird in IDEALS MTS nicht direkt, sondern mittels eines Verweismechanismus eingebunden. Eine bedeutende Rolle spielen dabei die Virtuellen Referenzen, mit denen i.A. nicht ein bestimmtes Lehr- oder Lernmodul, sondern eine Menge von inhaltlich äquivalenten Modulen angesprochen wird. Um eine umfassende Flexibilisierung zu ermöglichen, kann ein Autor bei der Definition einer Virtuellen Referenz den Attributen und Vorgabewerten Gewichte zuweisen. Mit ihnen wird der Grad der Berücksichtigung bei der Auflösung einer Virtuellen Referenz festgelegt. Bei der Definition von Vorgabewerten für Attribute ist ein Autor nicht auf einen Vorgabewert pro Attribut beschränkt. Die Werte aus dem Definitionsbereich der einzelnen Attribute können mit Hilfe von logischen Operator (UND, ODER, NICHT) verknüpft werden. Somit lassen sich komplementäre oder alternative Forderungen für Attribute festlegen und einzelnen Anforderungen einer Virtuellen Referenz verstärken bzw. abschwächen.
Neben der Definition einer Virtuellen Referenz ist besonders der Prozeß der Auflösung, als Mapping bezeichnet, von Bedeutung. Dieser Prozeß erfolgt iterativ und rückgekoppelt. Beim Mapping einer Virtuellen Referenz wird zunächst versucht, Module zu finden, deren Selbstbeschreibung den maximalen Satz an Anforderungen erfüllen. Ist eine ausreichend kleine Menge von Modulen gefunden, wird daraus der Modul ermittelt, der entsprechend der Vorgaben optimal ist. In Abhängigkeit von der Struktur der Virtuellen Referenz findet während des Mapping-Prozesses eine Generalisation oder Deduktion unter Berücksichtigung der gegebenen Gewichte statt, bis eine ausreichend kleine (oder große) Menge an inhaltlich äquivalenten Modulen gefunden ist. Schließlich wird ein optimaler Modul identifiziert.

3.5 Realisierung der Benutzerprofilierung

Das Benutzerprofil soll Auskunft über den aktuellen Wissensstand eines Lernenden geben können. Im Rahmen von IDEALS ist dieser Teilbereich bisher erst rudimentär entwickelt. Zur Zeit wird vom System automatisch protokolliert, welche Lernmodule ein Lernender gesichtet und bearbeitet hat. Zudem werden auch die erarbeiteten Resultate aus dem Bereich der Testate und der zu bearbeitenden Aufgaben in das Benutzerprofil aufgenommen. Unter Berücksichtigung der Information über die Lernobjekte können aufgrund der inhaltlich-didaktischen Attribute der Selbstbeschreibung Rückschlüsse auf das Wissensgebiet, spezielle Thematiken

etc. gezogen werden. Weitere Informationen wie Zeitpunkt der Bearbeitung, Zuverlässigkeitsmaß und anderer Gewichte werden gespeichert und stehen für eine Bewertung zur Verfügung. Es ist zu erwarten, daß auf Basis dieser Information ein umfassendes Bild des Lernfortschritts und -zustandes eines Lernenden erzeugt werden kann, das im Bereich einer automatischen Anpassung des Lernflusses nutzbar ist.
Zur Zeit beschränken sich allerdings die Aussagen über den Wissensstand eines Lernenden darauf, ob und mit welchem Resultat ein bestimmtes Lernobjekt bearbeitet worden ist. Die Bearbeitungszeitpunkte von Lernobjekten werden innerhalb des Benutzerprofils genutzt, um Prozesse der Verhärtung oder Auflösung von Wissen zu berücksichtigen. Aufgrund der Informationen aus dem Benutzerprofil ist es innerhalb der CNs bereits möglich, eine Flexibilisierung bzw. Adaption des Kursverlaufs zu erreichen. Als ein nächster Schritt werden Anfragen an das Benutzerprofil realisiert, in denen analog zu den Virtuellen Referenzen ein Anforderungsprofil definiert werden kann. Das Benutzerprofil wird entsprechend den Anforderungen bewertet und ein Ergebnis zurückgeliefert. Diese Bewertung kann dann, analog zur bereits vorhandenen Abfrage, zur Adaption genutzt werden.
Durch die Nutzung einer Datenbank zur Speicherung der Profil-Daten versprechen wird uns einerseits ein ausreichendes Maß an Offenheit zur Integration kognitionswissenschaftlicher und lernpsychologischer Modelle. Darüber hinaus stellt die Verwendung einer standardisierten Schnittstelle die Offenheit für Erweiterungen sicher.

3.6 Demonstration des MTS

Nachdem 1996 das IDEALS MTS entwickelt wurde, steht 1997 vor allem die Erprobung und Anwendung des MTS im Vordergrund. In einem ersten Schritt wurde anhand von kleinen Beispielen gezielt die Systemfunktionalität verifiziert. Nachdem dieser Schritt erfolgreich abgeschlossen wurde, folgt als nächster Schritt die Demonstration unter praxisrelevanten Bedingungen. Hierzu wurden im IDEALS Projekt zwei Demonstrationsszenarien identifiziert. Eines der beiden, das Universitätsszenario, wird im folgenden skizziert.
In diesem Szenario werden die Universität Coimbra (FCTUC, Portugal), die Universität in Oulu (RATOL, Finnland) und die Universität Darmstadt (GRIS, Deutschland) gemeinsam Lernmaterial zum Thema „Grundlagen der Computer Graphik“ erstellen und es in ihrem Lehrbetrieb einsetzen. Jeder der beteiligten Partner erstellt Material für die unterschiedliche Bereiche der Vorlesung, das dann gemeinschaftlich genutzt wird. So minimiert sich der Aufwand pro Partner für die Erstellung des notwendigen digitalen Lehr- und Lernmaterials bereits erheblich. Gleichzeitig kann an den verschiedenen Standorten eine Adaption des Lehr- und Lernmaterials an lokale Bedürfnisse erfolgen, die sich aufgrund der Modularität immer nur auf Teilbereiche auswirkt. Neben der Mehrfachverwendung ist auch die

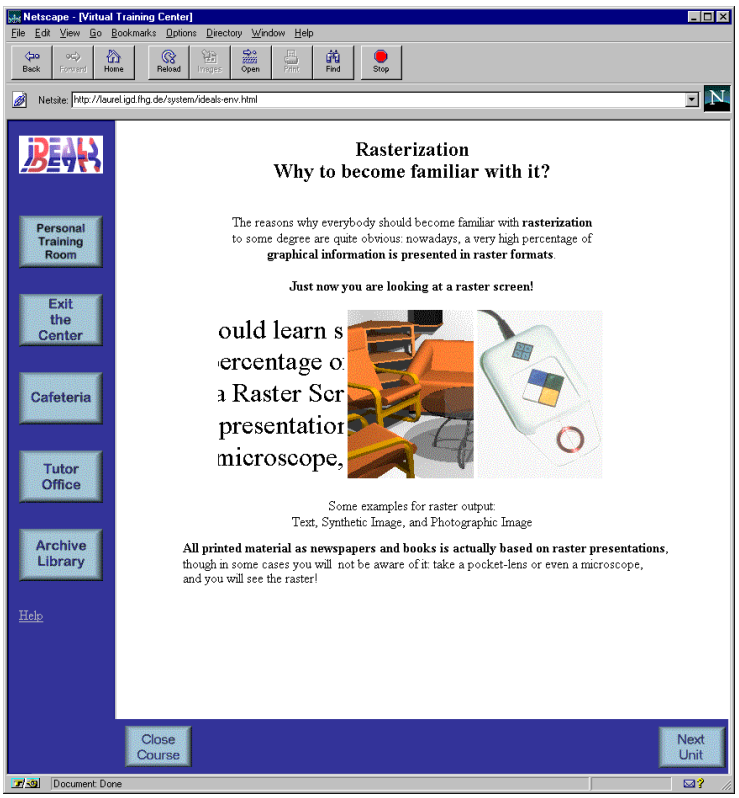


Abbildung 4 IDEALS Benutzungsoberfläche für Studenten

Wiederverwendung von erarbeiteten Mustern zu erwarten, die zusätzlich zu einer Verbesserung der Effizient bei der Erstellung beitragen wird.
Im Rahmen des Vorlesungsbetriebs wird GRIS parallel zu einer traditionellen Vorlesung einen rechnergestützten Kurs anbieten. Den Studenten bleibt es überlassen, ob Sie das Zusatzangebot wahrnehmen oder nicht, wobei volle Anonymität sichergestellt wird. Ihnen wird allerdings die Möglichkeit angeboten, die Ergebnisse des digitalen Kurses (Auswertung des Benutzerprofils und der Testergebnisse) in die Benotung einzubringen. Die gewonnenen Erfahrungen sollen Auskunft über die Effizienz des IDEALS MTS und den Lernerfolg der teilnehmenden Studenten geben. Gleichzeitig soll die Akzeptanz eines solchen Systems untersucht werden.
Die Abbildung 4 zeigt die aktuelle Benutzungsschnittstelle für Studenten, die über einen WWW-Browser realisiert wird. Dabei sind zwei unterschiedliche Regionen zu erkennen. Die Randregionen (dunkle Bereiche links und unten) enthalten den Teil der Benutzungsschnittstelle, die vom System vorgegeben ist, der restliche Bereich ist der Darstellungsbereich für die Lerninformation. Über die WWW-Adresse *http://ltc.gris.informatik.th-darmstadt.de* ist zur Zeit ein "Showcase" verfügbar, der einen kleinen Ausschnitt aus dem vorhandenen Angebot an Lehr- und Lernmaterial zeigt und erläutert.

4 Zukünftige Forschungs- und Entwicklungsaktivitäten

Im Bereich der Kurssteuerung gibt es zwei Bereiche, die der weiteren Forschung und Entwicklung bedürfen. Da ist zum einem die Benutzungsschnittstelle für die Kursnavigation und zum anderen Templates. Darüber hinaus sind im Bereich des Mappings und der Virtuellen Referenzen sowie im Bereich der Benutzerprofilierung noch Erweiterungen der Funktionalität notwendig.

4.1 Kursnavigation

Momentan ist die Kurssteuerungsfunktionalität rudimentär und dementsprechend wird im Moment eine statische Benutzerschnittstelle verwendet. Es stellt sich die Frage, ob eine Erweiterung der Kurssteuerungsfunktionalität empfehlenswert oder gar notwendig ist. Zu betrachten sind Erweiterungen wie Bookmarks oder nutzbare Historien der individuellen Kursverläufe.
Weiterhin ist zu erproben, inwieweit die Benutzungsschnittstelle dynamisiert werden kann und wie und in welchem Ausmaß der Autor dieses beeinflussen kann oder muß. So soll es zukünftig ermöglicht werden, in Tests bestimmte Funktionalitäten zu deaktivieren. Dies wird auch Auswirkungen auf die Benutzungsschnittstelle der Kursnavigation haben.

4.2 Templates

Ein anderer Bereich der Entwicklung betrifft die Funktionalität der Module. Betrachtet man das aktuelle System, so stehen den Modulen standardmäßig nur ein beschränkter Umfang an Funktionalität zu Verfügung. Will ein Autor ein Modul funktional erweitern, so muß er diese explizit von Hand programmieren. Eine solcher Erweiterung ist für den Autor mit erheblichem Aufwand verbunden, jedoch hält dieses Vorgehen ihm alle Möglichkeiten offen.
Zukünftig soll die Erstellung funktional erweiterter Module, insbesondere von Course Nodes, vereinfacht werden, ohne den Autor nennenswert in seinen Möglichkeiten einzuschränken. Hierbei ist daran gedacht, die häufigsten Gebrauchsmuster, die sich in der Demonstrationsphase herauskristallisieren werden, dem Autor als Vorlagen (Templates) anzubieten. Dabei gibt es mehrere Möglichkeiten zur Umsetzung, deren Vor- und Nachteile noch zu evaluieren sind.

- Erstellen einer Sammlung von gut dokumentiertem Beispiel-Lernmaterial, das ein Autor als Vorbild für eigenes Lernmaterial verwenden kann.

- Eine weitere Möglichkeit sind Wizards, d.h. spezielle Autorenwerkzeuge, die anhand von Autorenvorgaben automatisch ein Modulgerüst oder ein ganzes Modul erstellen, das ein Autor dann ergänzen und/oder erweitern kann.
- Generische Module. Auch hier gibt der Autor nur einige gewünschte Parameter vor und das Module wird automatisch erstellt. Im Unterschied zum Wizard, bei dem die Erstellung des Lernmaterials von einem externen Werkzeug übernommen wird, geschieht dies bei einem generischen Modul im System selbst. Typischerweise ist auch keine Erweiterung oder Ergänzung mehr durch den Autor nötig.

Auf lange Sicht soll derartiges intelligentes Verhalten als Standardverhalten in die Module selbst integriert werden.

4.3 Erweiterungen bei Virtuellen Referenzen und der Benutzerprofilierung

Im Bereich der Virtuellen Referenzen ist noch ein erhebliches Potential an Verbesserungen und Erweiterungen zu sehen. Es ist schon jetzt zu erwarten, das dem Mapping-Prozeß mit zunehmender Mächtigkeit der Modul-Domäne zunehmend Aufmerksamkeit bezüglich seines Laufzeitverhaltens gewidmet werden muß. Neben dieser Optimierungsaufgabe ist auch die Integration von kognitionswissenschaftlichen Erkenntnissen zur Lösung des Identifikationsproblems angestrebt. Darüber hinaus sind Erweiterungen in Richtung automatisierter Adaption unter Berücksichtigung des Benutzerprofils denkbar. Insbesondere die Integration von vorhandener Information über den Lernenden kann beim Auflösen von Virtuellen Referenzen genutzt werden.

Da bereits zum jetzigen Zeitpunkt Virtuelle Referenzen auch im Bereich der Erstellung von Lehr- und Lernmaterial zur Auffinden von bestimmten Modulen eingesetzt werden, ist ein erweiterter Einsatz als Navigationsinstrument in der Menge des Lehr- und Lernmaterials sowohl für Autoren als auch für Lernende anzustreben. Mit Hilfe von intuitiven Benutzungsschnittstellen kann durch einen generalisierten Navigationsmechnanismus ein vollständiges Lösen von der Kurs-Metapher erfolgen. Der Lernende kann auf die (gekapselten) Lernmodule (CNs) direkt über benutzerdefinierte Anforderungen zugreifen.

Die genannten Möglichkeiten besitzen allerdings als eine Grundvoraussetzung die Erzeugung eines wirklichkeitsgetreuen Benutzermodells, das sicher die größte Herausforderungen darstellt.

Literatur

[1] H. Kuhlmann, J.M.Brisson Lopes, "A Cross Platform Training Environment, Reusability in Heterogeneous Environments", Proceedings for the Telematics for Education and Training Conference, Düsseldorf/Neuss, 1994

[2] DEDICATED CONSORTIUM, ed.: H. Kuhlmann, "DEDICATED - Development of a new Dimension in European Computer-Aided Teaching and Education - Final Report", DELTA Final Report of Project D2014 DEDICATED, 1995

[3] Oracle Corp, "Oracle Learning Architecture: Bringing Education On-line and Into the Next Century, An Oracle White Paper", Oktober 1996

[4] E.N.Forte, M.Wentland, E.Duval, "ARIADNE: a Framework for Technology-Based Open & Distance Education", Proceedings of the EAEEIE '96, Oulu, Finnland, 1996

[5] M.Beyer, E.Borgmeier, "An open framework for Distributed Education using Multimedia", IEEE Computer Graphics and Applications, special issue on education, 1996

[6] M. Llamas, et.al., "The World-Wide Web as a distributed educational environment", Proceedings of the EAEEIE '96, Oulu, Finnland, 1996

[7] E.Borgmeier, F.Graf, J.M.Brisson Lopes, M.Mengel, "IDEALS, a Telematics Based Training Environment for the Future", Proceedings of the EAEEIE ’96, Oulu, Finnland, 1996

[8] C.Hornung, E.Borgmeier, T.Wang, "Teaching and Training in Intra- and Internet", Online Educa, Berlin 1996

URLs

[w1] Beispiele für universitäre Nutzung des WWW als Lehr- und Lernmedium: http://fernuni-hagen.de/, http://cuonline.edu, http://www.online.edu/

[w2] Beispiele für kommerzielle Nutzung des WWW als Lehr- und Lernmedium: http://www.gglearning.com/, http://www.internet-academy.de/, http://skylab.us.oracle.com/education/

[w3] IDEALS Home-Page und Demonstrator http://ideals.zgdv.de, http://ltc.gris.informatik.th-darmstadt.de

HyperCons

Eine hypermediale konstruktivistische Lernumgebung für Schulen

Peter Hubwieser, Johann Schlichter, Birgit Mair

Institut für Informatik der Technischen Universität München
80290 München, Tel. 089-289-25358,
e-mail: hubwiese@informatik.tu-muenchen.de

Zusammenfassung

Aufgrund der ausufernden Informationsmenge des Internet sowie der wachsenden immanenten Gefahren für Minderjährige bietet sich beim schulischen Einsatz die Vorschaltung eines speziellen Informationssystems an, in dem die Schüler einerseits leicht und schnell brauchbare Informationen finden, andererseits gefahrlos alleine herumsuchen können. Zusätzlich muß dieses System eine intelligente Unterstützung bei der Erstellung, Präsentation und Verwaltung eigener Dokumente bieten. Ausgehend von den im Rahmen des Rosenheimer Schulversuches "Internet an der Schule" geschaffenen Voraussetzungen soll hier ein spezielles Informationssystem für Schulen entworfen, aufgebaut und in einem realistischen Umfeld getestet werden.

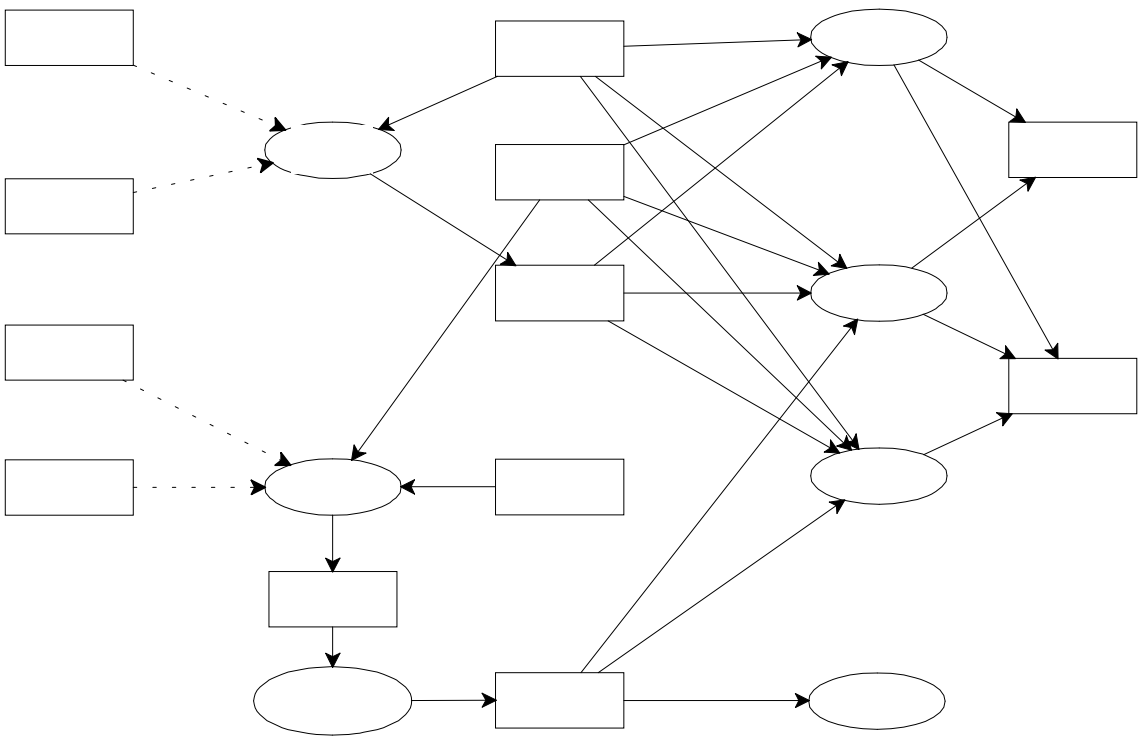
1 Motivation

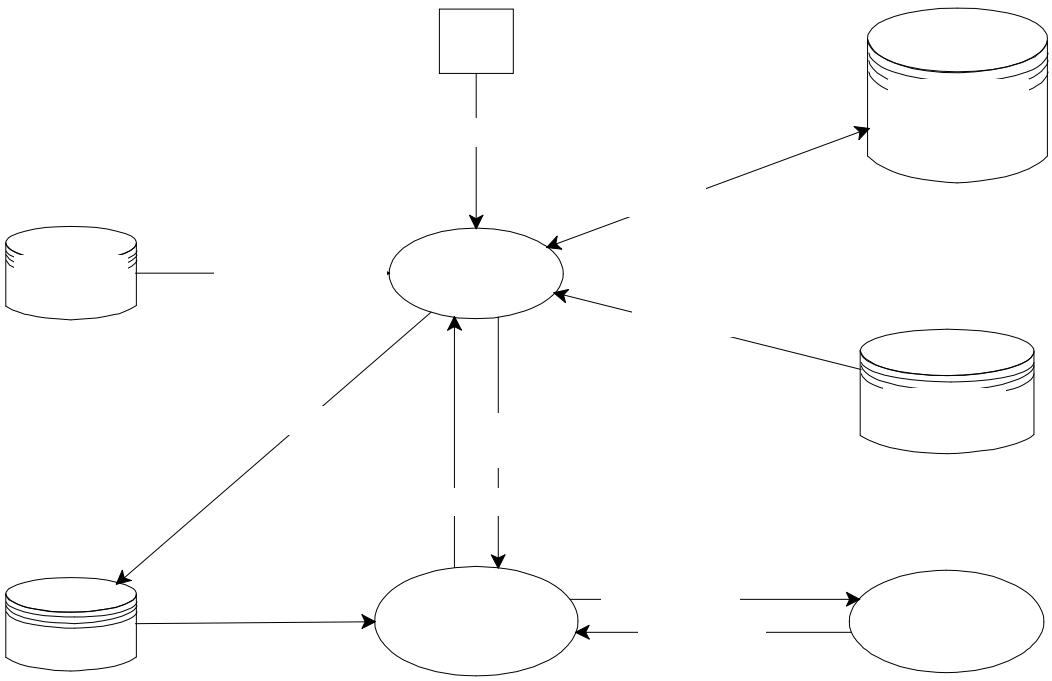
An unseren Schulen wird es in Zukunft wesentlich darauf ankommen, Schüler und Lehrer in effiziente Techniken zur schnellen Beschaffung und Verwertung von aktuell benötigten Informationen einzuführen. Dabei benötigen sie ein gut strukturiertes Informationsangebot, das eine große Menge überwiegend verwertbarer Informationen in ansprechender, also multimedialer Repräsentation anbietet. In Anbetracht der jüngsten Entwicklungstendenzen im Internet erscheint es mehr als fraglich, ob es möglich ist, dieses Medium an den Schulen zur Informationsrecherche einzusetzen, ohne zusätzlich spezielle Hilfsstrukturen aufzubauen.

Neben der Beschaffung von Informationen gewinnt die Produktion von eigenen Informationsangeboten durch Schüler immer mehr an Bedeutung. Der sinnvolle Einsatzbereich der Internet-Technik erweitert sich daher um die Bereiche Dokumentenproduktion sowie Präsentation und Verwaltung dieser Eigenprodukte. Für die Motivation der Schüler ist dabei von entscheidender Bedeutung, wie und von wem ihre Produkte über längere Zeiträume betrachtet und genutzt werden können. Um eine solche Nutzung in ausreichendem Maße gewährleisten zu können, ist ein Ordnungs- und Verwaltungssystem für Dokumente aller Art notwendig, das über die gegenwärtigen Möglichkeiten von Inter- oder Intranetstrukturen weit hinausgeht.

Der dritte, vielleicht pädagogisch fruchtbarste Einsatzbereich von Internet-Technologien an Schulen liegt in der Kommunikation mit Partnern außerhalb der eigenen Schule, des eigenen Landes, vielleicht sogar des eigenen Kontinents. Auch hierbei ist von einem Schulinformationssystem ausgiebige Unterstützung zu leisten.

Im hier beschriebenen Projekt soll in Zusammenarbeit mit ca. 20 Schulen aller Schularten untersucht werden, wie ein derartiges Informationssystem konzipiert werden muß,





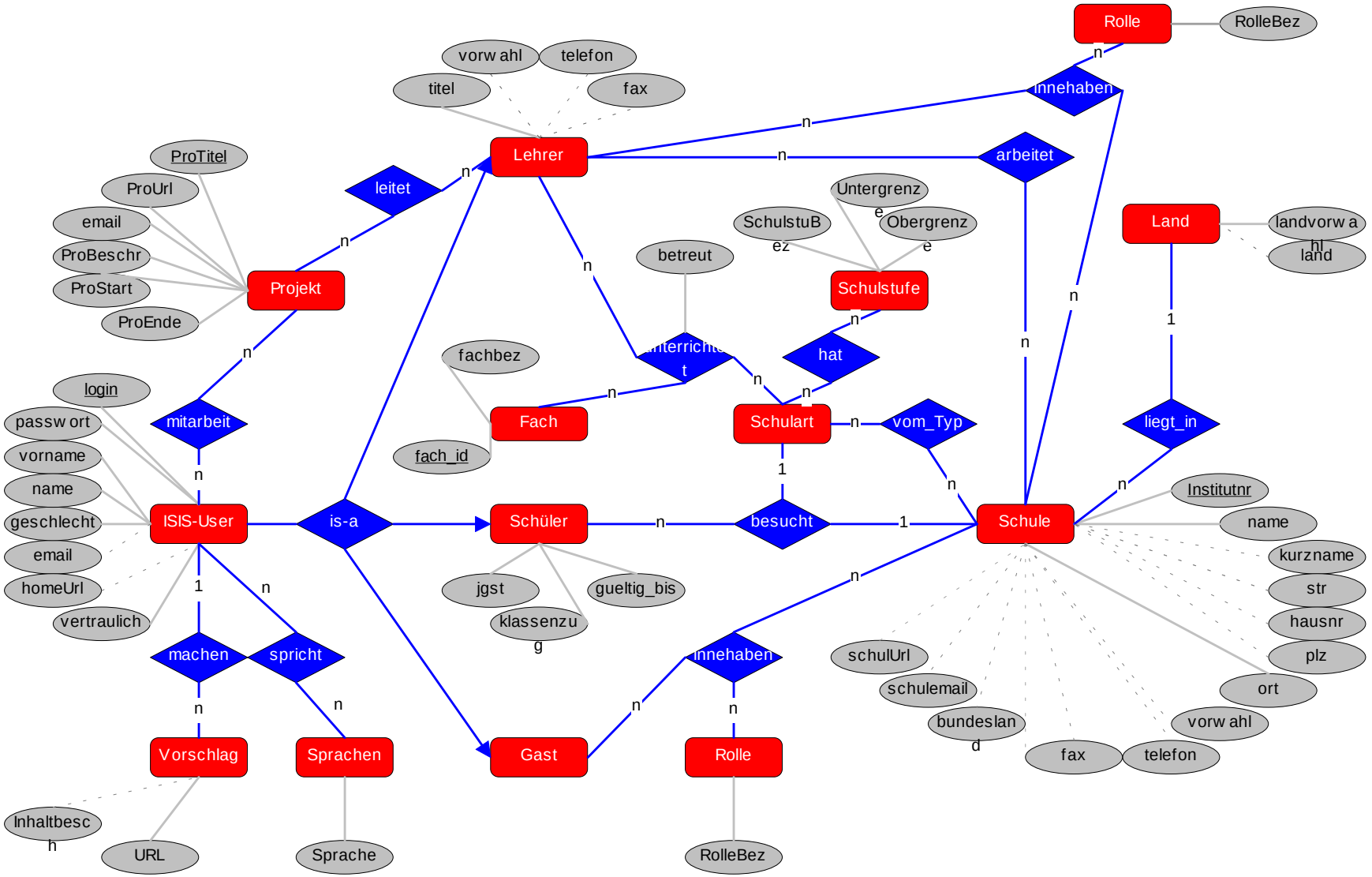


Abbildung 3: Ausschnitt aus dem ER-Modell

Dynamic Courseware Generation on the WWW

JULITA VASSILEVA AND RALPH DETERS
Federal Armed Forces University Munich
Institute for Technical Computer Science
85577 Neubiberg, Germany

[jiv, deters]@informatik.unibw-muenchen.de

Abstract

With WWW it became possible to provide learners with unlimited access to teaching materials, independently of time and their physical location. We have developed a tool for authoring of adaptive CAL courses, called "Dynamic Courseware Generator" (DCG). It generates individual courses according to learner goals and previous knowledge and dynamically adapts the course according to learner success in acquiring knowledge. The DCG runs on a WWW server. The student receives from this server an individualized course targeted to a goal, specified by him/her.

1 Introduction

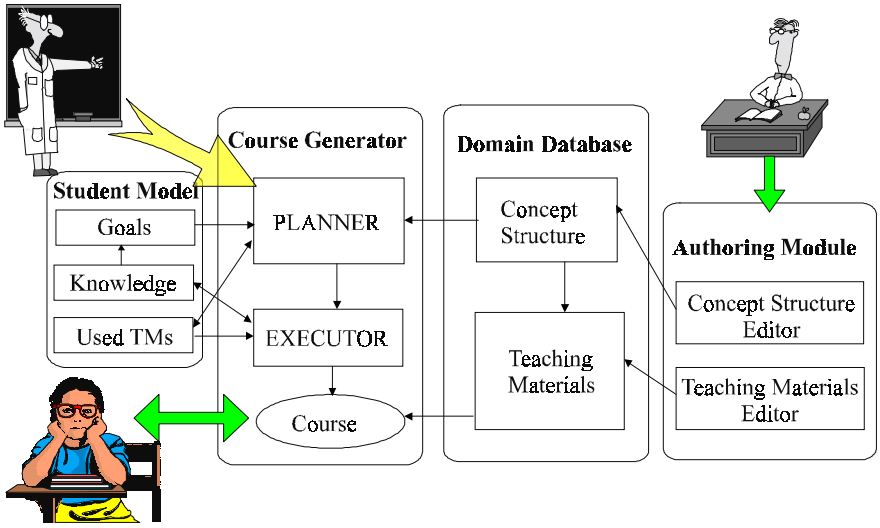
With the emergence of the WWW it became possible to provide learners with unlimited access to teaching materials independently of time and geographic location. However, these materials are spread through the network and it is not easy to find the relevant information when one wants to learn something specific. After visiting several sites one doesn't know where to go from the current point, how to organize/systematize all the information one has browsed so far, one sometimes even forgets what one's initial goal was. Though subjected to severe critics recently, the notion of „instructional course" is very appropriate for implementation on the WWW, because it ensures a goal-oriented, effective way teaching. It is especially suited for adults who are motivated to learn to achieve a specific goal. We have developed a tool, called "Dynamic Courseware Generator" (DCG), for automatic creation of individualized courses according to the learner's goal and previous knowledge which can dynamically adapt the course according to the learner's success in acquiring knowledge (Diessel, Lehmann, Vassileva, 1994), (Vassileva, 1992). It allows:

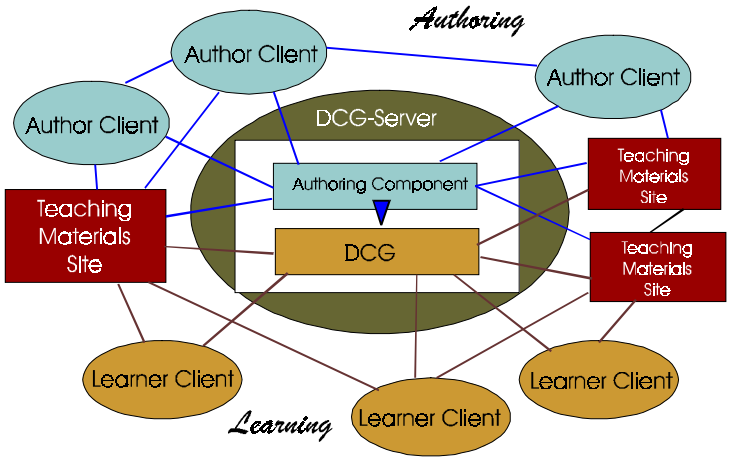
- automatic generation of CAL courses with different goals from existing teaching materials;
- creating different courses for students with different knowledge;
- changing the course dynamically according to the progress of the student.

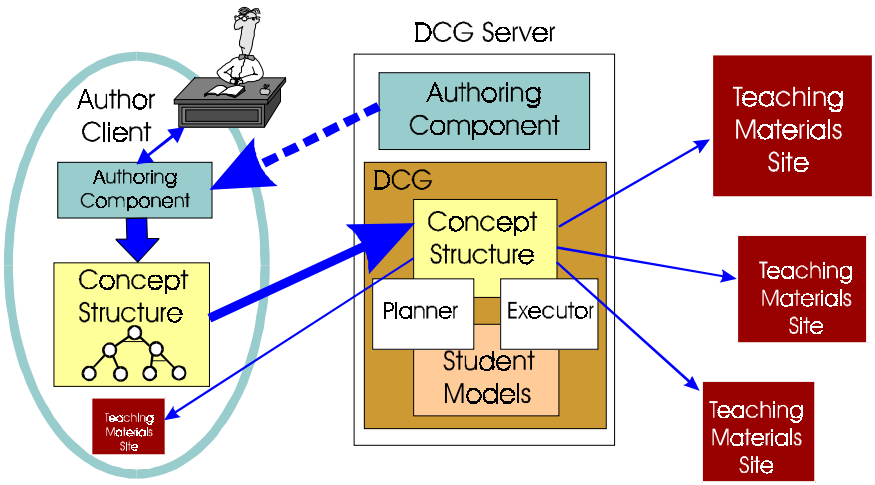
The DCG has been used as an authoring tool for generating adaptive courses in several domains and has proved to be very effective (Vassileva, 1995). We have recently scaled up the DCG so that it can be used for domain-authoring and automatic generation of adaptive courses on the WWW.

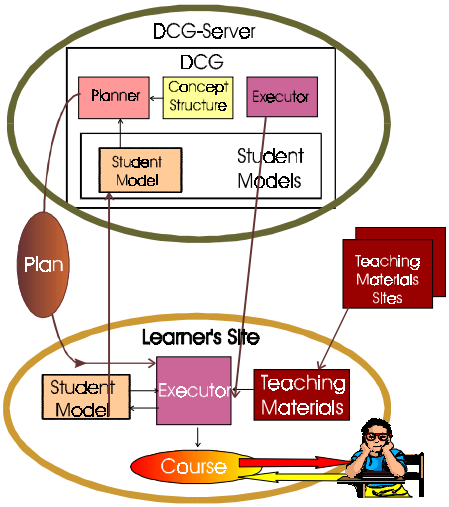
2 DCG on the WWW

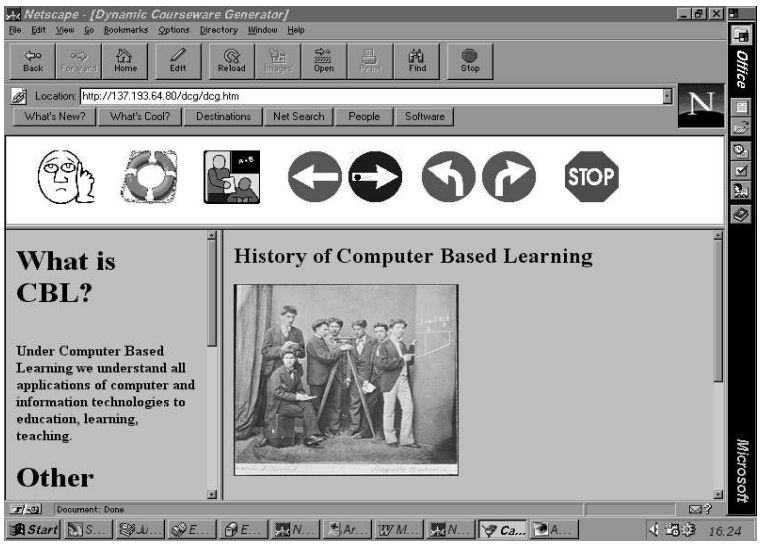
The main idea of the DCG is the explicit representation of the domain structure of the domain, separated from the teaching materials which the student sees on the screen. The domain structure is used by the system as a road-map to generate a plan of the course. Given a


Course Generator
Domain Database
Student Model
Goals
Knowledge
Used TMs
PLANNER
EXECUTOR
Course
Concept Structure
Teaching Materials
Authoring Module
Concept Structure Editor
Teaching Materials Editor


Authoring
Author Client
Author Client
Author Client
DCG-Server
Authoring Component
DCG
Teaching Materials Site
Teaching Materials Site
Teaching Materials Site
Learner Client
Learner Client
Learner Client
Learning


Author
Client
Authoring
Component
Concept
Structure
Teaching
Materials
Site
DCG Server
Authoring
Component
DCG
Concept
Structure
Planner
Executor
Student
Models
Teaching
Materials
Site
Teaching
Materials
Site
Teaching
Materials
Site


DCG-Server
DCG
Planner
Concept Structure
Executor
Student Model
Student Models
Plan
Teaching Materials Sites
Learner's Site
Student Model
Executor
Teaching Materials
Course


Netscape - [Dynamic Courseware Generator]
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Edit Reload Images Open Print Find Stop
Location: http://137.193.64.80/dcg/dcg.htm
What's New? What's Cool? Destinations Net Search People Software
STOP
What is CBL?
Under Computer Based Learning we understand all applications of computer and information technologies to education, learning, teaching.
Other
History of Computer Based Learning
Document: Done
Start
16:24

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

ENTWICKLUNG EINER OBERFLÄCHE FÜR DIE NETZANWENDUNG

Kooperative Lernumgebungen für kleine und mittlere Unternehmen auf Basis eines intelligenten Lernsystems

Dipl.-Inform. Matthias Kloth
Fraunhofer Institut für Materialfluß und Logistik
Joseph-von-Fraunhofer-Straße 2-4, D-44227 Dortmund
E-mail: kloth@iml.fhg.de

Stichworte: Lebenslanges Lernen, Computer Based Training (CBT), Tele-Lernen, Kooperatives Lernen, Intelligente Tutorsysteme (ITS)

Abstract
Der vorliegende Beitrag stellt zunächst den Bedarf an Weiterbildungsmaßnahmen dar, der durch traditionelle Ansätze kaum noch effektiv zu bewältigen ist. Die Entwicklungen der Informations- und Kommunikationstechnologien bieten hier neue Chancen in Richtung computerunterstützter Lernformen (CBT). Die aufgezeigten Grenzen bisheriger computergestützter Lernprogramme können allerdings durch neue „Online"-Ansätze allein auch nicht überwunden werden. Lernen ist ein kommunikativer Prozeß und er bedarf einer angemessenen didaktischen Unterstützung. Im Beitrag wird daher aufgezeigt, wie eine Verbindung von Tele-Kooperationstechniken und intelligenten Tutorsystemen die bisherigen Schwächen abzumildern vermag. Der Beitrag geht dabei auf eine Entwicklung des Fraunhofer IML ein, die in Zusammenarbeit mit anderen Instituten und Unternehmen für verschiedene Anwendungen erprobt wurde. Das Tutorsystem XTEACH bedient sich verschiedener Methoden intelligenter Lernsysteme und liefert eine von einzelnen Themenfeldern unabhängige Lernumgebung. Diese Umgebung wurde mit gängigen Tele-Kooperationstechniken gekoppelt und liefert über das Internet die Möglichkeit, dem steigenden Weiterbildungsbedarf einen wirksamen Baustein gegenüberzustellen:
URL: http://www.iml.fhg.de/projekte/RefPro/SimTeach

1 Neue Wege der Weiterbildung - Lebenslanges Lernen

1.1 Grundlagen Lebenslangen Lernens

„Weiterbildung ist eine Investition in die Zukunft des Industriestandortes Deutschland " - so lautete das Motto der Nationalen Konferenz Deutschland zum Lebenslangen Lernen NKDL [3] von 1996 [Nuiss96]. Mittlerweile ist es gesellschaftlicher Konsens, daß neben der schulischen Grund- und Allgemeinbildung sowie der beruflichen Erstausbildung eine permanente berufliche Weiterbildung unerläßlich ist für die Anforderungen, denen sich das Hochlohnland Deutschland in Zeiten globaler Märkte gegenübergestellt sieht.

In vielen Bereichen verdoppelt sich das Wissen innerhalb der nächsten fünf Jahre. 80% dessen, was ein Elektroniker in fünf Jahren für seine tägliche Arbeit benutzen wird, ist heute noch gar nicht entwickelt oder gar erfunden worden. Wissen und der Erwerb neuen Wissens entwickelt sich damit zu einem der wichtigsten wirtschaftlichen Faktoren. Unternehmen können in der Zukunft nur bestehen, wenn ihre Beschäftigten laufend neu dazulernen. Diese Erkenntnis führt zur Forderung nach einer praxisgerechten Schulung am Arbeitsplatz. G. Dohmen sieht dabei das Lernen im täglichen Lebens-, Arbeits- und Kommunikationszusammenhang als eine Grundvoraussetzung zur Sicherstellung motivierender Faktoren für das Lebenslange Lernen [Dohme97]. Zur Steigerung der Lernmotivation sollten neben der Schaffung geeigneter struktureller Rahmenbedingungen auch individuellen Bedürfnissen Raum gegeben werden. Derartige individuelle lernmotivierenden Faktoren sind u.a. die *Förderung des Selbstlernens* durch die Rückkehr zum natürlichen Erfahrungslernen und die Abkehr von isoliertem Lernen durch *Förderung kommunikativer Lernpartnerschaften*.

Weitere motivierende Faktoren nennt Dohmen in 25 Thesen zur Konzeption und Konsequenzen des Lebenslangen Lernens [Dohme97]. Diese Entwicklungen und Forderungen offenbaren auf der einen Seite, daß allein mit Hilfe traditioneller Weiterbildungskonzepte dem Bedarf sowohl in qualitativer als auch in quantitativer Hinsicht kein hinreichendes Angebot gegenübergestellt werden kann. Auf der anderen Seite ist jedoch auch zu konstatieren, daß die technologische Entwicklung im Bereich der Informations- und Kommunikationssysteme allein kein Allheilmittel darstellen wird.

Hier kommt es auf einen zielgerichteten Einsatz von Informations- und Kommunikationssystemen an, die bedarfsgerecht technologische und pädagogische Aspekte miteinander verbinden.

$IBODFO VOE (SFO[FO DPNQVUFSHFTU U[UFS 8FJUFSCJMEVOHTLPO[FQUF

Impulse für einen verstärkten Einsatz von Informations- und Kommunikationssystemen in der Weiterbildung sieht E. Schoop vor allen Dingen unter folgenden Gesichtspunkten [Schoop95]: Die fortschreitende Entwicklung bei den Informations- und Kommunikationssystemen führt

- zu einer Verstärkung interaktiver Elemente, die verschiedene Sinne beim Lernprozeß mit einbeziehen (Multimedialität),
- zu verstärkter Kooperation verteilter Lerngruppen (Computer Supported Cooperative Learning),
- zu einer Entkopplung der im heutigen Präsenzunterricht noch üblichen räumlichen und zeitlichen Synchronität von Lehrer und Schülern und
- zu vermehrtem Einsatz kontextbezogenen Lernens am Arbeitsplatz.

Demgegenüber sehen sich Anbieter traditioneller Weiterbildungsmaßnahmen vermehrt mit neuen Anforderungen konfrontiert, die aufgrund der sich ändernden Arbeitsweisen und Organisationsstrukturen Einfluß auf den Bedarf nach neuen Lernformen ausüben. Computergestützte Weiterbildung kann hier wesentliche Beiträge liefern. Sie wird klassische Formen nicht ersetzen, aber in erheblichem Umfang sinnvoll ergänzen. Multimediale, netzwerkbasierte Schulungen bieten:

- *Situationsbezogenes Training*: Mit Hilfe computergestützter, netzwerkbasierter Schulungen ist *Just-in-Time-Training* möglich, da der Zugang zu verschiedensten Kursen online schnell und einfach möglich ist. Die Nutzer bestimmen die Zeitpunkte des Lernens selbst.
- *Mitarbeiterpräsenz*: Computergestützte Weiterbildungen ermöglichen *In-Place-Training*, bei dem Lernen am Arbeitsplatzrechner stattfindet. Die Lernphasen können sich flexibel in den Arbeitsablauf einpassen. Die von externen Schulungen bekannten langen Abwesenheitsphasen entfallen.
- *Unterstützung des Wissenstransfers*: Bei computergestützter Schulung fallen Lern- und Arbeitsprozesse zusammen, daher kann die Verbesserung der Lernsituation und Umgebung direkt zu einer Verbesserung der Arbeitsprozesse führen (insbes. bei Themenstellungen zur Computerarbeit).
- *Aktualität*: Netzwerkbasierte Schulungsangebote bieten *aktuelle Informationen* an, da die Kursinhalte sofort nach ihrer Aktualisierung allen Kursnutzern in der neuen Form zur Verfügung stehen.
- *Kosteneffizienz*: Bei computergestützter Weiterbildung entfallen Kosten für Anfahrt sowie Übernachtung und die Ausfallzeiten der Mitarbeiter durch die bessere Einpassung der Lernphasen in den Arbeitsablauf verringert sich.

Weitere Vorteile sind die Lerner-Motivation, die mit interaktivem Kursmaterial erzielt werden kann, sowie die Möglichkeit zur Visualisierung dynamischer Vorgänge durch Videomaterial und Simulationen. Dies kann das Verständnis für Prozeßverhalten und komplexe Zusammenhänge fördern.

Trotz anerkannter Vorteile derartiger Schulungen wurden Potentiale bisher nur zu einem geringen Teil genutzt. Die Kritik am Lernen mit neuen Medien bezieht sich vor allem auf folgende Punkte:

- *Hohe Aufwände für das Erstellen von Kursen*: Ein Gutachten des VDI/VDE stellt fest, "...daß der Aufwand für eine didaktisch anspruchsvoll gestaltete Courseware außerordentlich hoch ist. Zur Erstellung einer Programmstunde muß heute ein Aufwand von etwas 400 Programmierstunden eines qualifizierten Autors gerechnet werden."[Lück96]
- *Mangelnde didaktische Kurskonzeption*: Viele Vorbehalte gegenüber der computergestützten Weiterbildung richten sich gegen die zugrunde liegenden didaktischen Modelle. In der Tat beruhen heute verbreitete Systeme häufig noch auf Konzepten wie programmiertem Lernen oder Instruktionsdesign [Stran95]. Ein weiteres Problem liegt in der Gleichstellung von informieren und lernen. Informationssysteme, die keine didaktischen Modelle unterstützen, führen eher zu einer Verwirrung der Nutzer durch ein Übermaß an Inhalten und einen Mangel an Strukturierungshilfen.
- *De-Sozialisation des Lernens*: Von verschiedenen Seiten wird die computergestützte Weiterbildung dahingehend kritisiert, daß sie kommunikative und sozialisierende Aspekte des Lernens unterschätzt. Sie fördere damit eine Isolation des Lernens, bei der Verständnisprobleme

unentdeckt oder unberücksichtigt bleiben. Die Fähigkeit, das Gelernte gegenüber anderen zu vertreten, wird nicht trainiert und so der Wissenstransfer in die Arbeitsabläufe erschwert.

Diese Kritikpunkte an computerunterstützter Weiterbildung sind bei der Entwicklung von XTEACH gezielt adressiert worden. Bevor jedoch auf die Systemplattform im einzelnen eingegangen wird, sollen zunächst noch die Tendenzen neuerer Entwicklungen der computergestützten Weiterbildung offengelegt werden, da sie integrativer Bestandteil der anschließend präsentierten Systemplattform sind.

/FVF "OT»U[F EFS DPNQVUFSHFTU U[UFO 8FJUFSCJMEVOH

5FMFMFSOFO VOE LPPQFSBUJWFT -FSOFO

Die Entwicklung des Lernens in Richtung einer computergestützten Weiterbildung hat sich nach E. Hörbst in folgenden Schritten vollzogen [Hörb96]:

1. *Traditionelles Klassenraum-Lernen*: In zeitlich und räumlich festgelegten Strukturen präsentiert ein menschlicher Tutor den Anwesenden den Lehrstoff. Die Kosten für die Maßnahme steigt in der Regel proportional zur Anzahl Teilnehmer. Die zeitliche Beanspruchung ist vorgegeben.
2. *Computerunterstütztes Lernen (CUL, CBT)*: Computerbasierte Trainingsprogramme werden an Arbeitsplatzrechnern lokal installiert. Dem Vorteil der ständigen Verfügbarkeit und einer Selbstbestimmung des Lerntempos steht der Nachteil isolierten Lernens gegenüber.
3. *„CBT-on-demand“*: Hier erfolgt die Verteilung der Lehrmaterialien über Netze. Sie können bedarfsgerecht und aktualisiert abgerufen werden, was eine weitgehende Zeit- und Orts unabhängigkeit gewährleistet. Das Einzellernen wird hierdurch allerdings nicht abgeschafft.
4. *Tele-Learning*: Das Tele-Lernen versucht die Vorteile des CBT-on-demand mit den Möglichkeiten, einer individuellen Betreuung durch einen menschlichen Tutor sowie dem Erfahrungsaustausch verschiedener Lerner untereinander zu verbinden.

Anwendungsbeispiele für das Tele-Lernen gibt es vor allen Dingen in großen Unternehmen:

- Audi schult Auszubildende in einem Lernzentrum mit interaktiven Comupter- & Video-Kursen,
- das Ford/VW-Joint Venture AutoEuropa setzt für die Qualitätssicherung interaktive Kurse ein,
- Mercedes-Mitarbeiter nehmen im Rahmen des Akubis-Systems an Videokonferenzen teil,
- SNI will mit einem Selbstlernzentrum „CBT on Demand“ und „Teletutoring“ realisieren und
- die Deutsche Telekom bietet ihre Global Learning Plattform für unterschiedliche Anbieter computergestützter Schulungsmaterialien an und integriert dazu Kommunikationsdienste.

All die hier aufgeführten Beispiele orientieren sich an partiellen Eigenschaften eines Lernprozesses. Zum einen reduziert sich in den Anwendungen der Lernprozeß auf eine Informationsbeschaffung, eine strukturierte, didaktische Führung durch das Material ist nicht vorhanden oder basiert auf Regeln des programmierten Lernens und kann daher auf die Fähigkeit einzelner Schüler nicht angemessen reagieren. Zum anderen wird die Präsentation der Materialien mit Hilfe von Videokonferenzen durch Tutoren begleitet. Dies impliziert jedoch wieder eine zeitliche Synchronisierung der Lernprozesse. Auch die Nutzung der WWW-Technologie allein bietet zwar den Vorteil eines direkten Zugriffs auf Lernmaterialien, ohne die Verwendung spezieller Tutorsysteme überwiegen für die Lernunterstützung allerdings eine Reihe von Nachteilen: Die Nutzung ist gedächtnislos, d.h. es gibt keine Verwaltung von Benutzerprofilen und dementsprechend keine angemessene Benutzerführung und Adaptivität auf individuelle Fähigkeiten.

*OUFMMJHFOUF 50VUPSTZTUFNF *54

Die ersten ‘Lehrmaschinen’ wurden in den 50er Jahren entwickelt und erfuhren eine weitere Verbreitung durch sogenannte CAI-Systeme (Computer Aided Instruction). Der pädagogische Hintergrund für diese Systeme bildete der Behaviorismus, dessen Grundgedanke - Lernen bedeute die Entstehung von Reiz-Reaktions-Verbindungen - zu festen Instruktionsschritten führte [Baumg94]. Die Kritik macht sich besonders daran fest, daß ihm ein sehr mechanisches Modell des Lernens zugrunde liegt, in dem innere Vorgänge als nicht erforschbar gelten. Aufgrund der geringen Entwicklungsaufwände dieser CAI-Systeme werden sie auch heute noch trotz aller pädagogischen Vorbehalte eingesetzt. Der größte Teil der angebotenen CBTs besteht daher immer noch aus traditionellen CAI-Ansätzen [Schul96].

Die KI-Forschung der 70er Jahre hat zu einer Erweiterung der Tutormodelle geführt. Die daraus entstandenen ITS orientieren sich an den drei Lernmodellen:

1. Behaviorismus: Lernen stellt die Entstehung von Reiz-Reaktions-Verbindungen dar. Innere Vorgänge der Informationsverarbeitung bleiben unberücksichtigt.
2. Kognitivismus: Es wird versucht die Verarbeitungsprozeduren des Gehirns zu simulieren. Im Vordergrund steht der Versuch, Verfahren zu lernen, durch deren Anwendung die Lösung des Problems erreicht werden kann.
3. Konstruktivismus: Lernen wird als aktiver Prozeß betrachtet, bei dem das Wissen in der Bezie - hung zu früheren Erfahrungen und komplexen realen Lebenssituationen konstruiert wird.

Die meisten ITS verwenden für das Lernermodell das sogenannte Overlay-Modell. Das Wissen des Lerners wird als Untermenge des Expertenwissens betrachtet, die während des Lern prozes- ses an das Wissen des Experten angenähert werden soll. Damit kann für jede Wissensein heit des Experten, die im Wissensmodell gespeichert ist, eine Menge von Attributen hinzugefügt werden, die den Wissensstand des Lerners bzgl. dieser Wissenseinheit angibt. Die Tutormodelle verwenden dabei didaktische Operationen und konstruieren damit einen Plan, zunächst als Cur ri- culum in Form einer globalen Pla nung und dann auf Basis einzelner Instruktio nen als lokale Pla - nung. Die Präsentation des Materials orientiert sich an den einzelnen Schritten dieser Planung, wobei das Tutorsystem bei der Planerstellung auf das Lernermodell und damit das „Wissen“ des Lerners zurückgreift.

Eine Hemmschwelle der ITS-Anwendung besteht in dem erforderlichen großen Entwicklungsauf - wand. Die wichtigsten Möglichkeiten, diesen Nachteil zu überwinden sind die Entwicklung von Tools, die die Erstellung von ITS unterstützen, oder die Wiederverwendung von bereits erstellten Komponenten ermöglichen. Bei der Wiederverwendung existieren verschieden Ebenen:

1. Wiederverwendung der Wissensbasis bzw. bereits vorhandener Datenstrukturen: Teile des für ein implementiertes Wissen werden in einem anderen verwendet.
2. ‘Libraries of teaching resources’: Wiederkehrende Pädagogische Konzepte wie Erklärun gen.
3. ‘Libraries of strategies’: Z.B. der Planer für den Instruktionsprozeß.
4. ‘structured ontologies’: Definition und Organisation von Eigenschaften für die verschiedenen Wissensarten, wie z.B. deklaratives, prozedurales oder kausales Wissen.
5. Wiederverwendung kompletter Module, damit wird das Wissen, welches in dem entsprechen - den Modul implementiert wieder verwendet.
6. Nutzung einer bereits bestehenden Architektur, hier findet die Wiederverwendung auf der höchsten Ebene statt.

Zwischen den verschiedenen Ebenen der Wiederverwendung und der Anpassung des ITS an die Bedürfnisse des speziellen Anwendungsgebietes und dem Adaptionsaufwand bestehen Trade- Off-Beziehungen. Eine ideale computergestützte Lernumgebung sollte daher auf der einen Seite für die Schüler ein breites Spektrum an Nutzungsformen anbieten. Dies sollte sowohl Telekom- munikationstechnologien mit einschließen als auch eine geeignete Aufbereitung und Steuerung des Materials durch ein intelligentes Tutorsystem ermöglichen. Den Entwicklern und Anbietern sollte eine Umgebung zur Verfügung stehen, die die Materialerstellung effizient unterstützt und weitestgehend auf Standards basiert. Im folgenden wird eine kooperative Lernumgebung beschrieben, die in ihrer Entwicklung konsequent auf diese Anforderungen ausgerichtet war.

95EACH FJO #FJTQJFM G S FJOF LPPQFSBUJWF -FSOVNHFCVOH NJU JOUFMMJHFOUFS5VUPSTZTUFN

)JOUFSHSVOE EFS &OUXJDLMVOH

Im Auftrag der Berkom GmbH - einer Tochter der Deutschen Telekom AG - haben sich Fraun - hofer Institute zusammengeschlossen, um Telekommunikationsanwendungen für kleine und mittlere Unternehmen zu entwickeln und zu erproben. Eines dieser Szenarien beinhaltete die Entwicklung einer kooperativen Lernumgebung [Encar96]. Die Anwen dung "Schulung, Training, Information" umfaßt innerhalb dieses Vorhabens aktuelle Themen wie Computerge stütztes Trai- ning mit fortschrittlichen Lerntechnologien sowie flexibles Lernen und Informationsabfrage. Der Prozeß des rechnergestützten Lernens und der Informationsbeschaf fung wurde dabei aus meh- reren Blickwinkeln betrachtet. Diese hängen jewei ls mit der unterschiedlichen An zahl von beteilig-

ten Personen und der Charakteristik der Kommunikation zusammen. Typische Lernszenarien sind hierbei individuelles Lernen und individueller Fernunterricht sowie verteilter Gruppenunterricht oder Gruppenlernen.

Im Rahmen dieser Anwendung konnten verschiedene Klassen von Beteiligten gewonnen werden:
- *Schulungsteilnehmer und Informationssuchende*: Benutzer von Informationstechnologie-Systemen,
- Anbieter von Schulungen und Information: Technologieproduzenten, Träger des Know-hows, Lehrer, Spezialisten, Tutoren,
- Schulungsmaterialentwickler: Kursmaterialdesigner, Kursautoren, Pädagogen, Didakten und
- Systemdesigner und -verwalter: Systementwickler, Operatoren und Netzwerkmanager.

Das wesentliche Ziel dieses Vorhabens war die Bereitstellung eines Systems für aktives Lernen mit multimedialen Kursmaterialien und dem damit gekoppeltem wahlfreien Zugriff auf die zugrunde liegenden Informationen. Hierbei sollen die ständig aktualisierten Lern- und Informationsmaterialien über Computer-Netzwerke und öffentliche Kommunikationsleitungen auch über große Entfernungen verteilt werden. Mit Hilfe eines intelligenten Tutorsystems kann dabei eine situationsangepaßte Reaktion des Systems auf den Benutzer erreicht werden. Für bestimmte Situationen wird jedoch ein über Netzwerke und audiovisuelle Mechanismen zugeschalteter menschlicher Tutor unabdingbar sein, der damit integraler Bestandteil des Gesamtsystems wird. Weiterhin werden kooperative Mechanismen zum Lernen und Informationsaustausch in der Gruppe zur Verfügung gestellt.

Die Erprobung dieser Technik erfolgt in den vier Themenfeldern
1. Teleberatung für die Qualitätssicherung,
2. Einsatz von kundenspezifischen ASICs,
3. Simulation für Produktion und Logistik sowie
4. Rechnergestützter Entwurf mit CAD-Systemen.

Die Anwendung ist jedoch nicht auf diese Bereiche beschränkt, sondern kann in beinahe beliebigen Gebieten eingesetzt werden, in denen eine computerunterstützte Lern- und Informationsumgebung realisierbar und sinnvoll ist.

Im Gegensatz zu vielen exisiterenden Anwendungen ist dieses Szenario an kleine und mittlere Unternehmen (kmU) als vordergründige Zielgruppe adressiert. Die Konzepte zur Mitarbeiterentwicklung in kmUs unterscheiden sich von denen großer Konzerne. Der Kostenrahmen ist häufig enger, die Mitarbeiterpräsenz besitzt noch höhere Bedeutung. Projektbegleitende Studien und Evaluationen zeigten, daß die Unternehmen in erster Linie eine Verbesserung der Ausbildungsqualität erwarten und weniger auf eine Kostensenkung setzen. Die computergestützte Weiterbildung soll eine Lücke schließen zwischen Printmedien und Seminaren.

KmUs erwarten vor allem, daß der Arbeitsprozeß dahingehend unterstützt wird, daß konkret vorliegende Arbeitssituationen sowie neuartige Arbeitsaufgaben besser bewältigt werden. Lernen soll zu Hause und am Arbeitsplatz stattfinden können, die Lerngeschwindigkeit soll individuell anpaßbar sein. Auch auf die individuelle Zusammenstellung der Lerninhalte wird großer Wert gelegt. Der Möglichkeit, einen menschlichen Experten im Bedarfsfalle als Berater hinzuziehen zu können, wird große Bedeutung beigemessen. Wichtig ist zudem die Unterstützung des gemeinsamen Lernens von Teams in fester Zusammensetzung an eventuell verschiedenen Standorten. Durch diese Unterstützung von Kooperation und Gruppenarbeit wird auch eine Zunahme organisatorischer Qualität erhofft.

Um den oben genannten Anforderungen Rechnung zu tragen, hat das Fraunhofer-Konsortium ein modulares Trainingssystem aufgebaut. Dieses System besteht aus verschiedenen Komponenten, die nach außen auf mittlerweile etablierte Standardschnittstellen aufbauen. Kern des Systems ist ein CBT-Server, auf dem das Tutorsystem XTEACH läuft. Dieses System verwaltet die Kurse sowie die für den Lernprozeß relevanten Benutzerdaten und ist für die Ablaufsteuerung der CBT-Kurse verantwortlich. Das Kursmaterial liegt in Form von HTML-Dokumenten auf diversen WWW-Servern und wird während der Lernsitzungen vom Tutorsystem adressiert. Darüber hinaus sind in die Umgebung von XTEACH weitere Schnittstellen zu Telekooperationssystemen integriert, so daß verschiedene Lerner untereinander und mit einem Lehrer direkt am System miteinander

kommunizieren können. Die Kursentwickler können auf Standard-Applikationen zur Erstellung der HTML-Dokumente zurückgreifen. Außerdem existiert eine Autorenumgebung, die die Erstellung der Wissensrepräsentation (s.u.) und deren Verknüpfung mit den Materialien unterstützt.

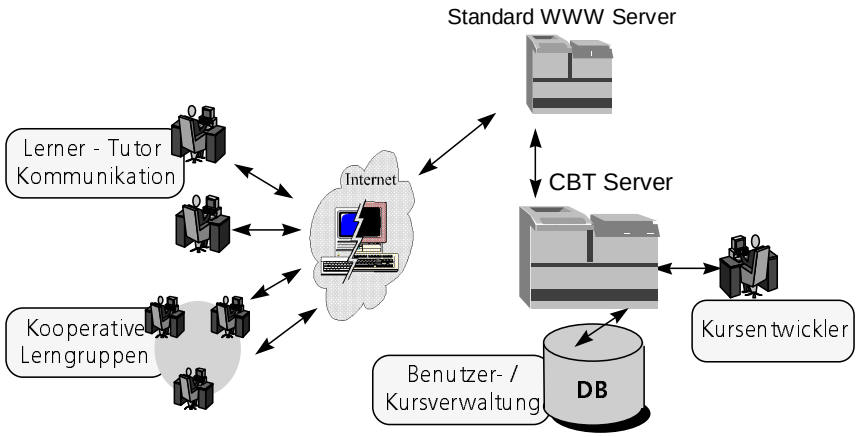


Abbildung 1: Komponenten und Teilnehmer der XTEACH-Umgebung

%BT95 EACH 5VUPSTZTUFN

Der Kern dieser modularen Lernumgebung besteht aus einem Tutorsystem XTEACH, das am Fraunhofer IML entwickelt wurde. Dieses System verwendet Techniken aus dem Bereich ITS, um die Konfiguration des Kursablaufes auf die individuellen Bedürf nisse einzelner Schüler anzupas-sen und bei der Steuerung einzelner Einheiten auf didaktische Regeln und einer Struktur der Domäne zurückgreifen zu können. Dieser komplexen internen Ablaufsteu erung steht eine ein-fache Schnittstelle zu dem extern abgelegten Material gegenüber. XTEACH basiert auf einer strikten Trennung von Kursmaterial und dem Skript, das die Abhängigkeiten, Strukturen und Relationen zwischen den Materialien beschreibt. XTEACH verwendet zwei Ansätze zur Senkung der Kurserstellungs-Kosten: die mehrfache Nut zung von Material innerhalb eines Kurses, sowie die Wiederverwendung des Materials f ür neue oder weiterentwickelte Kurse. Diese Ans ätze werden durch eine hochgradig modulare Kursstruktur realisiert, der sowohl eine vernetzte als auch eine hierarchische Strukturierung des Basiseinheiten zugrunde liegt. Das MTS unterschei-det zwischen *Kursinhalten, Kurslayout* und *Kursstruktur.* Diese drei Ebenen werden unabh ängig voneinander entworfen und ermöglichen so eine mehrfachen Nutzung . Kursinhalte sind multi-mediale Materialien, die mit Standardeditoren erstellt wer den. In der Auto renumgebung des MTS werden die Kursstrukturen definiert. Bei der Erstellung eines neuen Kurses k önnen existierende Kursinhalte direkt in die neue Struktur eingebunden werden.

XTEACH besteht aus folgenden Modulen:

1. Das *Wissensmodul* dient zur Repräsentation der Domäne. Eine Art Overlay-Modell deckt die Struktur des zu vermittelnden Wissensgebietes ab, in dem über verschiedene Hierarchiestufen Einheiten und ihre Abhängigkeiten abgebildet werden. Da nicht feste Verzweigungen sondern nur kontextsensitive Abhängigkeitsrelationen beschrieben werden, findet hier eine Abkehr von Konzepten des programmierten Lernens statt hin zu einem konnektionistischen Modell, das Lerneraktionen und -vorkenntnisse berücksichtigt.
2. Das *Benutzermodul* repräsentiert das Lernerverhalten. Unter Verwendung von Methoden des maschinellen Lernens wird hier die Adaptivität des Systems an die Lernerfähigkeiten verbes-sert. Das System berücksichtigt bei der Auswahl von didaktischen Regeln das bisherige Lern -verhalten, um besonders erfolgversprechende Strategien zu bevorzugen.
3. Das *Tutormodul* ist für die dynamische Kurssteuerung verantwortlich. Die vom Lerner selektier-ten Themenstellungen konfiguriert das Tutormodul zur Laufzeit. Ab hängig von Informationen aus Wissens- und Schülermodul und unter Verwendung didakti scher Regeln stellt es dann Einheiten auf den konkreten Bedarf eines einzelnen Schülers zusammen.
4. Die Integration von *Simulationswerkzeugen* in das System unterstützt das explorative Lernen in Mikrowelten. Mit Hilfe vorgegebener Modelle können die Benutzer in Form von Exkursen an Simulationsinstrumenten experimentieren und so ihr erlerntes Wissen explorativ anwenden.

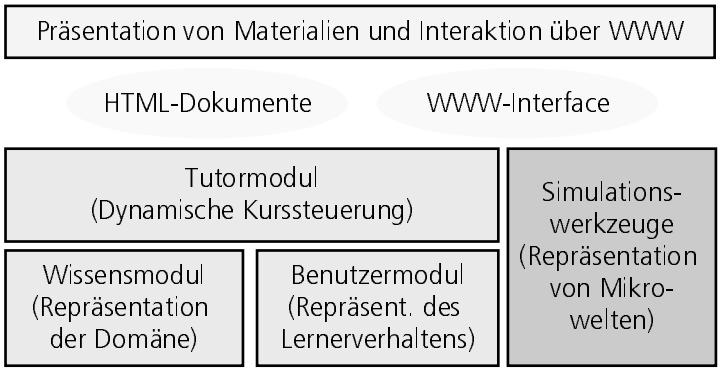


Abbildung 2: Komponenten des Tutorsystems XTEACH

8JTTFOTNPEVMUSVLUVSJFSVOH EFS %PN»OF

Das *Wissensmodul* repräsentiert die Struktur der Domäne. Die Domäne ist partitioniert in eine Menge von Wissenseinheiten, die den gesamten Lehrstoff aufspannen. Den Zugang zur Domäne erhält der Benutzer über eine Zielhierarchie, die die verschiedenen Sichten auf die Domäne in Form unterschiedlicher Fragestellungen definiert. Teilziele sind in der Zielhierarchie in Form eines UND-Baums angeordnet. Das Tutormodul arbeitet die ausgewählten Ziele im Tiefendurchlauf ab. Ein Ziel gilt dabei als erfüllt, wenn sämtliche seiner Teilziele erfüllt sind. Die Blätter des Zielbaums verweisen auf eine Folge von Wissenseinheiten, die bis zu einem angegebenen Lerngrad bearbeitet werden müssen, um dieses Blatt als erfüllt markieren zu können.

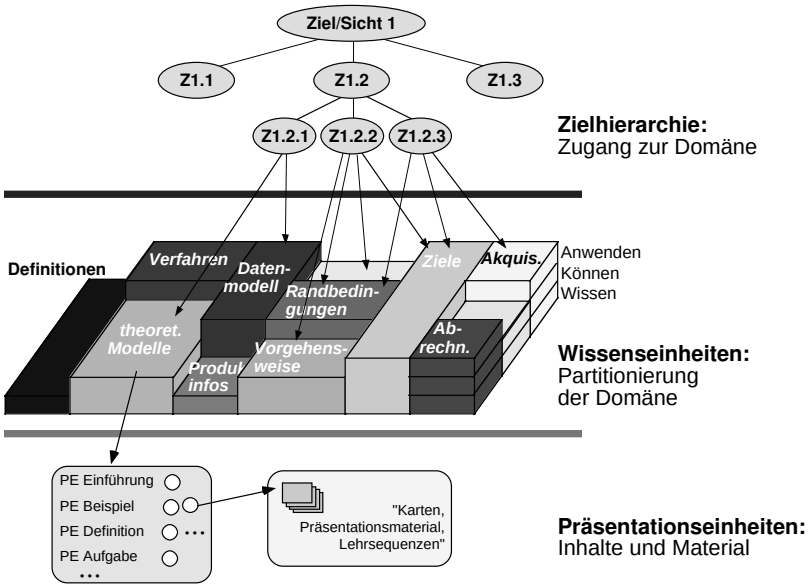


Abbildung 3: Beispiel der Strukturierung des Lehrstoffes

Der Lerner kann während einer Konsultation ein Ziel auswählen, bspw. eine Fragestellung, die ihn im Augenblick besonders interessiert. XTEACH erweitert das selektierte Ziel um den dazugehörigen Unterzielbaum und stellt dann eine Sequenz von Wissenseinheiten zusammen, die diesen Unterzielen zugeordnet sind. Das präsentierte Material setzt sich letztlich aus den für die Vermittlung der Wissenseinheiten zuständigen Präsentationen zusammen. Neben der Zuordnung zu Zielen besitzen die Wissenseinheiten noch eine Vorgängerrelation, die Wissenseinheiten angeben, die bereits in einem gewissen Lerngrad erreicht worden sein müssen, bevor die eigentliche Wissenseinheit präsentiert wird. Falls der Benutzer nicht den vom Tutor vorgeschlagenen Weg einschlagen möchte, sondern selbst ein Unterziel selektiert, stellt die Vorgängerrelation sicher, daß bei der Zusammenstellung einer Sequenz von Wissenseinheiten auch unbekanntes aber notwendiges Vorwissen mit präsentiert wird.

Folgende Lerngrade sind in XTEACH realisiert:

– Wissen: Wenn der Benutzer diesen Lerngrad erreicht hat, kennt er die wichtigen Begriffe zu diesem Themenkomplex und kann sie in einen größeren Kontext einordnen.

- Anwenden: Dieser Lerngrad umfaßt darüber hinaus noch Kenntnisse über Zusammenhänge mit und Abgrenzungen zu anderen Themenbereichen.
- Transferieren: Die Erreichung dieses abschließenden Lerngrads versetzt den Benutzer in die Lage, die Themen umzusetzen und für eigene individuelle Problemstellungen zu benutzen.

Mit der erfolgreichen Abarbeitung der Einheiten, zu denen immer auch Aufgaben zur Lernzielkontrolle gehören, erreicht der Benutzer in der Wissenseinheit den korrespondierenden Lerngrad.

Mit Hilfe dieser Ebenen ist es dem Kursentwickler zum einen möglich, die Domänenaufteilung (konzeptuell, Ebene Wissenseinheiten) und die Materialerstellung (ressourcenorientiert, Ebene Präsentationseinheiten) zu trennen. Die so repräsentierte Domäne erlaubt nun über die Zieldefinition verschiedene Zugänge, d.h. über die Zielbaumerstellung spannt der Kursersteller den gesamten Raum möglicher Folgen von Wissenseinheiten auf, die der Benutzer auswählen kann.

5VUPSNPEVM)POUFYUBCI»OHJHF ,POGJHVSBUJPO WPO ,VSTTFRVFO[FO BVG #BTJT EJEBLUJTDIFS 3FHFMO

Der Lerner kann XTEACH in verschiedenen Nutzungsformen anwenden. In Abhängigkeit einer Navigationsart liegt dabei die Initiative zur Zielselektion vermehrt beim Benutzer (benutzergeführte Sitzungen), eher beim System (tutorgeführte Sitzungen) oder ausschließlich beim System (Guided Tour). In tutorgeführten Teilen der Konsultation wendet das Tutormodul didaktische Regeln an, um aus vorhandenen Ressourcen einer zu vermittelnden Wissenseinheit eine Sequenz von Präsentationen zusammenzustellen. Die Regeln sind in einer generischen Modustabelle hinterlegt. Die Sequenzermittlung erfolgt dabei über eine Instanziierung eines generischen Moduseintrags. Die tutorgeführte Sitzung verkörpert die eigentliche Nutzung des Systems als Schulungssystem im engeren Sinne. Der Benutzer kann sich zunächst aus dem Komplettangebot der Ziele eine Untermenge auswählen, die dann vom Tutor nach didaktischen Gesichtspunkten zu einem Lehrblock konfiguriert wird. Bei der Abarbeitung dieser Einheiten analysiert der Tutor die Aktionen des Benutzers und benutzt sie zur Planung und Konfiguration weiterer Einheiten. Der Tutor führt außerdem eine Lernzielkontrolle durch und bietet im Falle einer unzureichenden Aufgabenbearbeitung ein Paket an Wiederholungsmaßnahmen an. Bei benutzergeführten Sitzungen wird dem Benutzer die Initiative zur Auswahl einzelner Einheiten übertragen. Während er im Tutormodus nur Themenblöcke (Ziele) auswählt, werden ihm in der benutzergeführten Sitzung einzelne Einheiten (Präsentationen) zur Auswahl vorgestellt. Hier kann das System als Informationssystem genutzt werden, um gezielt einzelne Themen präsentiert zu bekommen.

Die im Tutormodul gespeicherten Lehrstrategien werden über die Auswahl der Präsentationen aus den dazugehörigen Wissenseinheiten umgesetzt. Damit kann z.B. die induktive Lehrstrategie 'Vom speziellen zum generellen' modelliert werden. Die angewendete Strategie gibt weiterhin vor, welchen Lerngradwechsel der Lerner für die entsprechende WE vollziehen soll. Es wird zwischen folgenden übergeordneten Modusklassen unterschieden:

Modus	Modus-Intention	Lerngrad Wissen	LG Können	LG Anwenden
Minimal	kürzest mögliche, sinnvolle Sequenz			
Maximal	längste mögliche, sinnvolle Sequenz			
Induktiv	vom Bekannten zum Neuen			
Deduktiv	vom Abstrakten zum Konkreten			

Wenn bspw. das Erreichen eines Ziels die Wissenseinheit WEx im Lerngrad Anwenden beinhaltet und der Schüler diese WE im Lerngrad Wissen bereits erfolgreich bearbeitet hat, setzt das System im minimalen Modus aus den Einträgen der Tabelle
- LG Können: Intro-K, Bsp-K, Inhalt-K, Aufg-K
- LG Anwenden: Intro-A, Inhalt-A, Transfer-A, Aufg-A

eine neue Strategie zusammen (bspw.: Intro-K, Bsp-K, Inhalt-K, Inhalt-A, Transfer-A, Aufg-K, Aufg-A).

#FOVU[FS C[X 4DI MFSNPEVMEBQUJWJU»U NJU)JMGF NBTDIJOFMMFS -FSOWFSGBISFO

Im *Benutzermodul* wird das Lernerverhalten repräsentiert, um eine Adaptivität des Systems an die individuellen Lernerfähigkeiten erreichen zu können. Diese Adaptivität, d.h. die automatische Anpassung des Systems an die Bedürfnisse des Benutzers, vollzieht sich in der Auswahl und Instanziierung des Lehrmodus. Aus dem Pool an Inhalten werden an Hand dieser Modi die Themen und Materialien ausgewählt, die für den Lerner den größten Lernerfolg versprechen.

Ziel war es hier ein Modul zu schaffen, das aus den eigenen Aktionen und deren Ergebnissen „lernt“, um seine Aufgabe, die Vermittlung von Wissen, besser realisieren zu können. „Besser “ bedeutet hier, den durchschnittliche Erfolg zu steigern, den Lerner bei der Bearbeitung von Aufgaben haben. Dazu müssen der Benutzer und seine aktuelle Sitzungssituation mit bereits erfaßten Daten verglichen, d.h. klassifiziert werden. Für diese Aufgabe wurde ein Verfahren aus dem Bereich des maschinellen Lernens herangezogen, der kNN-Algorithmus (vgl. Nearest-Neighbor Algorithmen [Wett94]). Das Verfahren vergleicht die aktuelle Situation mit an deren bekannten und wählt die Lehrstrategie aus, die bei der „ähnlichsten “ Situation erfolgreich war.

Das unter 3.2.2 aufgeführte Instanziierungsbeispiel ist eine Alternative aus der Vielzahl an Einträgen in den einzelnen Feldern der Modustabelle. Die Entscheidung für diese Alternative fällt das System aufgrund der Einstufung des Benutzers in eine bestimmte Benutzerklasse, die der kNN-Algorithmus gebildet hat. Der Erfolg oder Mißerfolg dieser Strategie (bei der Bearbeitung von Lernzielkontrollen) hat dann nachträglich eine Verstärkung bzw. Abschwächung der Zugehörigkeit zur Benutzerklasse und eine Bestätigung bzw. Abänderung des Auswahlverfahrens bei der Modusinstanziierung zur Folge. Das maschinelle Lernverfahren schafft so die Vorausset zung, daß zu den einzelnen CBT-Kursen die Benutzerklassen immer schärfere Konturen gewinnen und die dazu korrelierenden Auswahl- und Instanziierungsverfahren immer erfolgreicher werden.

,PQQMVOH NJU 4JNVMBUPSFO "SCFJUFO NJU .JLSPXFMUFO

Neben der Präsentation Lehrmaterial ermöglicht XTEACH auch die Anwendung extener Applikationen. Dies ist unter anderem angebracht, wenn der Lehrstoff auf spezielle Softwaresysteme Bezug nimmt, bspw. auf CAD-Programme. In solchen Fällen kann in Form von Exkursionen die Applikation real demonstriert werden. Technisch wird dazu eine Funktion der XTEACH-Umgebung verwendet, die über das WWW-Protokoll eine Umleitung des Outputs an den Client veranlaßt.

Eine weitergehende Integration ist mit der Kopplung von speziellen Simulationsinstrumenten vorgesehen. Hier werden dem Lerner Simulationsmodelle angeboten, in denen er im Sinne von Mikrowelten explorativ arbeiten kann. Die Auswirkungen seines Handels kann er sich über die Simulatoranimation visualisieren oder über eine spezielle Parameterbewertung des Tutorsystems auswerten lassen.

%JF95 &"$) 6NHFCVOH

In die XTEACH-Umgebung sind Kommunikationsdienste eingebettet und zu einem *Virtuellen Training Centers* (VTC) integriert. VTC bezeichnet die Design-Metapher für die WWW-Umgebung, mittels derer die Kommunikationsdienste mit dem XTEACH-Kern zusammengefaßt werden. Es beinhaltet den Zugang und die Nutzung dieser Kurse mittels des Tutorsystems XTEACH als auch die Funktionen, die zur Realisierung eines kooperativen Trainings-Ablaufs notwendig sind. Innerhalb des VTC verteilen sich die Funktionen auf die „Räume “ Cafeteria, Tutor Offices, Personal Training Rooms und Archiv:

- Cafeteria: Die Cafeteria dient zur Realisierung des asynchronen Group Learning. Sie bietet die Möglichkeiten zur Kommunikation über Email, Mailinglisten und ein durch Internet-Newsgroups realisiertes schwarzes Brett. Um die Zuordnung von Benutzern zu Gruppen sowie die Administration von Gruppen und das Scheduling von Gruppen zu unterstützen, wurden die Cafeteria-Interfaces um Applets mit Gruppeninformationen erweitert. Mit Hilfe dieser Applets können Benutzer sich eine Liste aller Gruppen, zu denen sie Zugang haben, anzeigen lassen. Sie können jeweils eine dieser Gruppen als aktive Gruppe selektieren, und sich die aktiven Mitglieder anzeigen lassen. Über die ‘Call’ und ‘Mail All’-Funktionen können sie mit Gruppenmitgliedern Nachrichten austauschen. Sie können temporäre Lerngruppen erzeugen, die eine Untergrup -

pe einer festen Benutzergruppe bilden und für Mitglieder dieser Benutzergruppe offenstehen.

- Tutor Offices: In den Tutor-Offices finden die Nutzer Beratung bei Fragen zum Kurs oder zur XTEACH-Umgebung. Der Verbindungsaufbau zum Tutor und die Selektion der Medien (mail, talk oder Audio/Video-Kommunikation) wurde dabei durch den Einsatz eines Coordinating-Tools automatisiert. Dieses Tool ist als eigenständiges Java-Programm realisiert, das auf allen Java-fähigen Plattformen einsetzbar ist, und jeweils so konfiguriert werden kann, daß die lokal installierten Kommunikationssysteme genutzt werden.
- Personal Training Rooms: Dies ist der Zugang zu den eigentlichen Kursen. Neben einer kurzen Online-Bedienungsanleitung und einem Beispiel findet sich der Button zum Start des Navigator-Applets, das als eigentlicher Client des Kurs-Servers fungiert.
- Archiv: Das Archiv bietet einen strukturierten Zugang zu weiteren Informationen im WWW. Hier wurden für die verschiedenen Kurse jeweils eigene Verweislisten realisiert, die den spezifischen Interessen der Kursanbieter entsprechen.

In Bezug auf die verfügbare Kooperationsunterstützung stehen jetzt 3 Stufen zur Verfügung, je nach Umfang der Konfiguration der Clients und Verfügbarkeit von Kommunikationssystemen: ohne jede spezielle Ausstattung oder Konfiguration können die EMail-basierten Gruppenkommunikations-Mechanismen der Cafeteria genutzt werden (mit einem Java- und Mail-fähigen WWW-Browser). Ist das Coordinating-Tool installiert und mit entsprechenden Kommunikationstools für Chat oder Audio-Videoconferencing verbunden, können die Benutzer die direkten Kommunikationsmöglichkeiten der Tutor Offices nutzen. Sind zusätzlich Systeme für Application Sharing installiert (z.B. *VirtualX* oder *MS-NetMeeting*), so können synchrone Gruppenlernsitzungen stattfinden. Beim Gruppenlernen erarbeiten die Schüler gemeinsam einen Kurs, wobei der Tutor ggfs. Diskussionsrunden leitet. Die Sicht der Schulungsteilnehmer auf das Kursmaterial kann nach belieben geteilt bzw. synchronisiert werden. Es wird die Erfahrung menschlicher Experten bei der Wissensvermittlung genutzt und es wird gemeinsames Lernen und Bearbeiten durchgesetzt. Die Gruppenmitglieder können dabei über Kursinhalte bzw. Fragestellungen diskutieren, sich bei Unklarheiten oder Fragen gegenseitig unterstützen und Aufgaben gemeinsam lösen.

%JF &OUXJDLMVOHTVNHFCVOH

Kursmaterial wird in Standardformaten (z.Z. HTML) repräsentiert und unabhängig vom Kursskript gespeichert. Damit ist eine Wiederverwendbarkeit sichergestellt und der Aufwand zur Generierung einer Wissenseinheit reduziert, da jede Ressource nur einmal vorhanden ist. Zur Erstellung der Materialien kann der Autor auf Standard-Tools für Multimedia- und HTML-Dokumente zurückgreifen. Mit Hilfe einer Kursentwicklungsumgebung beschreibt der Autor das Kursdesign in Form eines Skriptes, in der alle Objekte und deren Abhängigkeiten definiert sind. Der Ablauf einer Sitzung wird auf Basis dieses Skriptes zur Laufzeit vom Tutormodul in Abhängigkeit der Schüleraktionen geplant und konfiguriert. Bei der Auswertung von Schüleraktionen verwendet XTEACH vom Autor vorgegebene Lernzielkontrollen, um auf Basis des Schülerwissensstandes weitere Lernsequenzen zu planen.

95&"$) "OXFOEVOHFO VOE &SGBISVOHFO

Im Rahmen des Projektes waren die Fraunhofer Institute IAO (Stuttgart), IBMT (St. Ingbert), IGD (Darmstadt), IIS (Erlangen), IML (Dortmund) und IPK (Berlin) mit jeweils eigenen Industriepartnern an der Entwicklung und Anwendung beteiligt. Die Anwendungsbereiche, in denen das System eingesetzt und evaluiert wurde und für die entsprechende Kurse entwickelt wurden, umfassen eine ganze Bandbreite technischer Disziplinen. Zielgruppen in allen Anwendungsbereichen sind die Mitarbeiter und Anwender, die in kleinen und mittleren Unternehmen dem oben beschriebenen Anspruch der ständigen Weiterbildung in ihrem Aufgabenfeld gerecht werden müssen. Die Erfahrungen zeigten generell, daß die Verbindung von intelligenter Lernsoftware mit Telekooperationsmechanismus einen signifikanten Qualitätssprung im Rahmen der betrieblichen Weiterbildung ermöglicht.

Bei den Kursen handelt es sich im einzelnen um die Themenbereiche:

- *Kompetenzzentrum Tele-BIT*, bei der sich kleine und mittlere Unternehmen bspw. in der Thematik Qualitätsmanagement informieren und weiterbilden können,

- *ASIC-Entwurf*, dem Entwurf integrierter Schaltungen mit einem ersten Schwerpunkt " Field Programmable Gate Arrays",
- *Logistik*, das zunächst das Thema „Simulation in Produktion und Logistik“ vermittelt sowie
- *Rechnergestützte Produktentwicklung - CAD*, in dem das Wissen zu allen Phasen der Produktentwicklung unter schwerpunktmäßiger Betrachtung von Rechneranwendungen vermittelt wird.

Exemplarisch wird im folgenden der Anwendungsbereich Logistik näher beschrieben.

Im Bereich von Produktion und Logistik müssen Unternehmen sich immer wieder au fgrund der Entwicklung moderner Technologien neuen Anforderungen stellen und kö nnen auf den Nutzen der Simulationstechnik nicht verzichten. Die Simulation hat sich als zentrales, unverzicht bares Hilfsmittel herauskristallisiert, um Neuerungen oder Änderu ngen von Produktionsstätten, Produktionsmethoden, Geschäfts- oder Logistikprozessen im Vorfeld zu bewerten. Mit ihrem Einsatz wird das kostengünstige Experimentieren am Modell einer geplanten Anlage ermöglicht und der Anwender kann mittels der Simulat ionsergebnisse komplexe Zusammen hänge zwischen Ursache und Wirkung verstehen und analytisch schwer faßbare Vorgänge bewerten .

Das Fraunhofer IML in Dortmund hat einen Kurs zum Thema Simulation mit Schwerpunkt im Bereich Logistik entwickelt. Zukünftige Simulationsanwender werden an die angemessene Modellierung und korrekte Parametrisierung eines logistischen Systems herangeführt und ihnen werden Kenntnisse über die Möglichkeiten und Interpretati onen der Simulationsergebnisse vermittelt. Nach Ablauf des Kurses sollen die Teilnehmer Simulationsprojekte richtig abschätzen und effizient durchfü hren können. Die Wissensinhalte vermitteln dem Lerner spezifisch für den Bereich Produktion und Logistik
- eine Übersicht über die Methode Simulation
- eine Vertrautheit mit der simulationspezifischen Fachterminologie
- eine Entscheidungskompetenz über einen Simulationseinsatz und
- eine Unterstützung für die Begleitung einer Simulationsstudie

Zielgruppe für den Logistik-Kurs ist die kaufmännische oder technische Führungskraft, die sich über den eventuellen Einsatz der Simulation in ihrem Unternehmen informieren möchte. Der typische Anwen der ist der zukünftige Projektleiter bei einem Unternehmen, das di e Beauftragung einer Simulationsstudie erwägt. Eine Rechnervertrautheit, d.h. eine gewisse Routine im Umgang mit einem Rechner, ist von Vorteil, aber nicht zwingend notwendig. Kenntnisse im Bezug auf Simulation werden nicht vorausgesetzt, es sollte aber ein thematisches Orientierungs- und Konzeptwissen zum Thema Produktion und Logistik vorhanden sein.

Die Zielhierarchie präsentiert mögliche Sichtweisen auf die Domäne und splittet sie in drei Zugän- ge Grundwissen, Theorie und Praxis. Der Benutzer kann sich durch die Auswahl des ersten Teil - zieles das Grundwissen über das Thema Simulation aneignen. Die Teilziele Theorie und Praxis vermitteln dem Lernenden theoretisch-abstrakte Konze pte bzw. praktisch-konkrete Aspekte.

;VTBNNFOGBTTVOH VOE "VTCMJDL

Die Erfahrungen des abgelaufenen Vorhabens verdeutlichen folgende Vorteile der hier vorge - stellten kooperativen Lernumgebung:
1. Überwindung der Nachteile unstrukturierter Informationsbeschaffung durch ein angemessene Lehrsteuerung mit Hilfe eines benutzeradaptiven, intelligenten Tutorsystems,
2. Verhinderung der De-Sozialisation des Lernens durch die Integration mit Kommunikations tech- nologien und Förderung von Lernpartnerschaften sowie Lern en als kommunikativen Prozeß,
3. Einfacher Zugang über das Internet mit Hilfe gängiger Standard-Tools sowie skalierbare Anwendung der Telekommunikationsumgebung sowie
4. Unabhängigkeit der Systemkomponenten von speziellen Themenstellungen und Übertragbar- keit auf weitere Anwendungsfelder.

Derartige neue Anwendungsfelder sind derzeit auch geplant bzw. bereits in Angriff genom men. Ein aktuelles Vorhaben am Fraunhofer IML bezieht sich auf das Thema Supply Chain Manage- ment und will in einer Weiterbildungsmaßnahme die Entwicklung von kooperativen Lernnetzen zu kooperativen Produktionsnetzen fördern. Dieses Vorhaben verfolgt mit der Entwicklung und

Erprobung einer verteilten Lernumgebung für das kooperative Supply Chain Management folgende Ziele, die Modellcharakter für vergleichbare Aufgabenstellungen haben können :

- *Kooperation erleben durch kooperatives Lernen - Verbinden von Lerninhalt und Lehrmethode*
  „Kooperation" im Supply Chain Management soll gelehrt werden auf Basis kooperativen Lernens. Curricula und Lerninhalte werden gezielt darauf ausgelegt, Synergien von Lehrmethode (Arbeiten in räumlich verteilten Teams) und Lerninhalt (überbetriebliche Kooperation im Supply Chain Management) zu nutzen. Verteiltes, kooperatives Lernen wird als Schlüssel für den Aufbau von *lernenden Netzen* gesehen, die Kooperationen in der Lernumgebung bereits *erleben*.

- *Supply Chain Management als ganzheitliches Instrument erfahren*
  Die Vermittlung der Lehrinhalte beinhaltet ein breit gefächertes Spektrum zum Supply Chain Management und berücksichtigt dabei sowohl die verschiedenen Rollen der Beteiligten, damit verbundene organisatorische und strukturelle Ausgangslagen wie auch neue Methoden und Instrumente, die im späteren Arbeitsalltag eine Rolle spielen. Die ganzheitliche Sicht auf die kooperative Vermittlung von Aufgaben und Methoden soll dabei nicht nur den Wissenserwerb sondern vor allen Dingen Kompetenzerwerb fördern. Ziel ist es hier von den *lernenden Netzen* zu *Kompetenznetzwerken* zu kommen.

- *Die Entwicklung von Lernpartnerschaften zu Produktionspartnerschaften unterstützen*
  Das Lernen im lernenden Netz bis hin zur Bildung von Kompetenznetzwerken bietet den beteiligten Unternehmen letztlich sogar die Möglichkeit, von den Lernpartnerschaften zu Produktionspartnerschaften zu kommen. Sowohl die Erfahrungen im Umgang mit verteilten aber kooperativ agierenden Lerngruppen als auch die mit der Umsetzung des Erlernten verbundene Reorganisation wirken als Katalysator für den Aufbau regionaler und internationaler Partnerschaften. Die während der Maßnahmen entstandenen Kontakte zu anderen Unternehmen und ihren Teilnehmern können im realen Geschäftsleben weitergeführt werden. Quasi als Nebeneffekt erwachsen aus den Kompetenznetzwerken selbst wieder *Produktionsnetzwerke* (Supply Networks), die ihre Erfahrungen gemeinsam am Markt umsetzen können.

-JUFSBUVS

[Baumg94] Baumgartner, P.: Wie Lernen am Computer funktioniert. In: c't 8, 1994, S. 138-142

[Borgm96] Borgmeier, E.; Kloth, M.; Köhler, C.: Schulung, Training, Information. In: Encarnacao, J.L.; Hornung, Ch.; Osterfeld, U. (Hrsg.): Telekommunikationsanwendungen für kleine und mittlere Unternehmen, Springer Verlag, Berlin, Heidelberg, New York, 1996, S. 145-162

[Dohme97] Dohmen, G.: Lebenslanges Lernen. In: Günther, W.; Mandl, H. (Hrsg.): Telelearning, Telekom Multimedia Systemhaus, 1997, S. 19-30

[Encar96] Encarnacao, J.L.; Hornung, Ch.; Osterfeld, U. (Hrsg.): Telekommunikationsanwendungen für kleine und mittlere Unternehmen. Springer Verlag, Berlin, Heidelberg 1996

[Euler92] Euler, D.: Didaktik des computerunterstützten Lernens - Praktische Gestaltung und theoretische Grundlagen. B&W Verlag, Nürnberg 1992

[Glowa92] Glowalla, U.; Schoop, E.: Entwicklung und Evaluation computerunterstützter Lehrsysteme. In: Glowalla, U.; Schoop, E. (Hrsg.): Hypertext und Multimedia: Neue Wege in der computergestützten Aus- und Weiterbildung. Springer Verlag, Berlin, 1992, S. 21-36

[Gräse97] Gräsel, C. et al.: Lernen mit Computernetzen aus konstruktivistischer Perspektive. In: Günther, W.; Mandl, H. (Hrsg.): Telelearning. Telekom Multimedia Systemhaus, 1997, S.19-30

[Gudjo95] Gudjons, H.: Pädagogisches Grundwissen. Klinkhardt Verlag, Bad Heilbrunn, 1995

[Hörb96] Hörbst, E.: Moderne Technikangebote für Bildung und Weiterbildung. In : Konferenzband Lebenslanges Lernen der NKDL3, 25.-26. Nov. 1996, ISBN 3-8167-4592-X, S. 41-60

[Kears87] Kearsley, G.: Artificial Intelligence and Instruction - Applications and Methods. Addison-Wesley Publishing Company, Reading Mass., 1987

[Lück96] Lück, W. van: Lehrerinnen und Lehrer im 2010 - Brauchen wir Multimediapädagogen? In: Die Informationsgesellschaft, Report Bundesminist. für Wirtschaft, Bonn 1996, S. 30f

Ein Konzept zur Modellierung von Handlungszielstrukturen in intelligenten Lehrsystemen

Eva Kluge
Freie Universität Berlin – Institut für Informatik
Takustr. 9, D–14195 Berlin
Tel: +049 (30) 838 75 146 FAX: +049 (30) 838 75 109
e-mail: kluge@inf.fu-berlin.de

1 Einleitung

Aufgabe von *Lehre* – ob menschlicher oder computergestützter – ist "das absichtliche Bewirken von Lernprozessen."[7]. In einem Intelligenten Lehrsystem (ILS) hat die Tutorkomponente die Aufgabe, die Lernprozesse zu unterstützen. Strategien wie entdeckendes Lernen, sokratischer Dialog, das "content-planing" und "delivery-planing" können in ILS umgesetzt werden (vgl.[24]). Parallel dazu wurden Systeme entwickelt, die sich bei der Modellierung an kognitionspsychologischen Annahmen orientieren. Besonders zu erwähnen sind der LISP-Tutor von J.R. Anderson [3], [2] und SOAR von Newell, Laird, Rosenbloom(s. z.B. [28]), deren Ziel die Modellierung kognitiver Lernprozesse ist. Ein System, welches seine Lehrstrategie auf einer kognitiven Annahme aufbaut, ist PETRI-HELP (s. z.B.[6]). PETRI-HELP und seine Folgesysteme basieren auf der Impasse-success-problem-driven-learning-theory von van Lehn [26]. Gerade die zuletzt genannten Ansätze haben gezeigt, daß die Modellierung eines ILS nach kognitionspsychologischen Erkenntnissen die Qualität der Systeme erhöht.

Diese Arbeit hat zum Ziel, eine *sachgebietsunabhängige* Methode für die Modellierung einer tutoriellen Komponente zu definieren, die auf der Basis einer *handlungstheoretisch begründeten Lerntheorie* agiert. Aus einer handlungstheoretischen Position heraus ist ein Lernprozeß ein *zielgerichteter* Prozeß in dem ein hierarchisch-sequentiell organisiertes Handlungsgrundschema zur Bewältigung eines Problems/einer Aufgabe aufgebaut wird [19] (vgl. Abschnitt 3). Ziel von Lehre ist es demnach, den Aufbau geeigneter Handlungsgrundschemata zu bewirken. Orientiert man sich an den Eckpfeilern dieses Schemas, den Zielen, so soll Lehre die Lernende darin unterstützen, sich geeignete Ziele zu setzen. Damit erscheint die *Lernzielorientierung* in einem neuen – nicht behavioristischen – Licht (vgl. Abschnitt 2).

Die Basis für die Lernerführung ist eine Menge von lernrelevanten Handlungszielen über ein Wissensgebiet. Auf einer solchen Menge lassen sich formale Relationen definieren, die die Formulierung von Tutorregeln erlauben, ohne explizit Bezug zum Sachgebiet nehmen zu müssen (vgl. Abschnitt 4.1). Um eine größtmögliche Individualität im Lernprozeß zu ermöglichen, werden die Relationen mit Attributen belegt, die in der Interaktion der Lernenden mit dem System bewertet werden. Beim Entwurf einer Handlungszielstruktur,

bezogen auf ein bestimmtes Wissensgebiet, entsteht zunächst eine formale, noch unbewertete Struktur. Damit ist diese Methode im Prinzip auf jedes ILS anwendbar: Einzige Voraussetzung ist, daß Handlungsziele über dem Sachgebiet formuliert und vor allem vom System überprüft werden können.

Im letzten Abschnitt wird die Umsetzung am Sachgebiet von SYPROS (vgl. [14]), entwickelt an der TU München, skizziert.

2 Die Bedeutung von Lernzielen bei der Steuerung von Lernprozessen

Lehre – personale oder computergestützte – muß die Brücke zwischen "Innen" und "Außen" schlagen: Ziel von Lehre ist das Bewirken von Lernprozessen. Ein Lernprozeß ist ein innerer Vorgang, der eine Veränderung des mentalen Zustandes der Lernenden bewirkt ([12], S. 85). Festgestellt werden können solche Zustandsänderungen jedoch nur am *beobachtbaren Verhalten* der Lernenden[1]([12],S. 90). Eine Möglichkeit, den Erfolg von Lehre prüfen zu können, ist die Formulierung von Lernzielen. "Ein Lernziel formuliert einen veränderten *Zustand* des Schülers, der durch [den] Unterricht bewirkt werden *soll*"([12],S. 85). Eine Übertragung dieser Methode auf ILS verlangt eine *Operationalisierung* von Lernzielen. Ein Lernziel ist so zu formulieren, daß es " das durch den Unterricht veränderte *beobachtbare Verhalten*"([12],S. 91) beschreibt. Da ein Lernziel gleichzeitig einen inneren Zustand repräsentiert, der durch seine Erarbeitung bewirkt werden soll, erfordert die Aufstellung von Lernzielen einen Bezug zu einer lerntheoretischen Position, welche die inneren Vorgänge erklärt. Zudem vollziehen sich Lernprozesse immer an einem *Inhalt* [12]. Die Formulierung von Lernzielen über ein Wissensgebiet muß demzufolge die Besonderheiten dieses Wissens berücksichtigen. Hier sind Lernziele ein didaktisches Mittel zur Aufbereitung des Wissens, so daß sein Lernen erleichtert wird.

3 Lernen aus handlungstheoretischer Sicht

Bezüglich der lerntheoretischen Annahme wird in dieser Arbeit die handlungstheoretische Position vertreten. Aus dieser Position heraus wird Lernen als ein Prozeß erklärt, in dem ein auf ein Ziel gerichtetes Handlungsgrundschema[2] (vgl. [19], S. 107) aufgebaut wird. Ein Handlungsgrundschema ist ein hierarchisch-sequentiell organisiertes Gefüge von kleineren, aber in sich vollständigen und ebenfalls zielgerichteten Handlungseinheiten ([9], S. 39f). Die kleinste Handlungseinheit ist die **T**est**O**perate**T**est**E**xit-Einheit ([9], S. 34). Eine `TOTE`-Einheit beschreibt einen kompletten Handlungsentwurf (vgl. [19], S. 115ff. und auch [25], S. 72 ff.): In einer Orientierungsphase (Test) wird ein vorläufiger Handlungsplan entworfen, welcher durch die aktuelle Situation (Ist–Zustand), den antizipierten Endzustand (Soll-Zustand) und die zum Zwecke der Zielerreichung antizipierten Handlungen (Operationen) konstituiert wird. Sowohl die Interpretation des Ist-Zustandes als auch die

[1] Diese Aussage trifft in der Tat auf beide Arten von Lehre zu: auch ein menschlicher Lehrer muß sich auf *"Äußerungen"* der Lernenden beziehen, wenn er den Erfolg seiner Lehre beurteilen will. Im Gegensatz zu einem ILS stehen ihm jedoch mehr Möglichkeiten zu Verfügung.

[2] Synonyme für den Begriff Handlungsgrundschema sind *Handlungsplan* oder *Aktionsprogramm* (siehe [25] S. 58) oder *Handlungsstruktur* ([27], S.22f)

Antizipation des wünschenswerten Endzustandes und die Auswahl von "passenden" Operationen resultieren aus dem Wissen und Können, welches der Lernenden zur Verfügung steht. Der antizipierte Handlungsplan wird in der Phase der Realisierung durchgeführt (Operate). Die durch die Handlungsausführung entstehenden Veränderungen sind gleichzeitig wieder Ausgangspunkt für eine neue Testphase: Während der Realisierungsphase wird eine neue Orientierungsphase aktualisiert, so daß die Lernende schon während der Durchführung ihrer Handlung neue (Korrektur-) Handlungen antizipieren kann. Während der Ausführung einer Handlung entsteht so ein aktualisierter Handlungsplan, der neben Anfangs- und Endzustand auch die Zwischenzustände repräsentiert.
Die Operate-Phase wird durch eine Testphase abgelöst, die das antizipierte Ziel und das Erreichte vergleicht. Bei hinreichender Konkordanz wird der aktuelle TOTE-Zyklus verlassen (Exit). Der entsprechend aktualisierte Handlungsplan wird zusammen mit der Zielorientierung in das Könnensrepertoire übernommen.

Die hier sehr verkürzt dargestellte Erklärung von Lernen aus handlungstheoretischer Sicht (nach Kaminski[19]) zeigt drei Aspekte sehr deutlich:

Erstens ist eine Lernhandlung ein hierarchisch-sequentieller Prozeß von Handlungseinheiten. Diese Handlungseinheiten selbst lassen sich in unterscheidbare Phasen zerlegen, vergleichbar mit dem Phasenmodell der RUBICON - Theorie (vgl. [10], S. 330 ff.). Anders als im RUBICON - Modell betont die Handlungstheorie die *Ziegerichtetheit* einer Handlung, was im Hinblick auf die Umsetzung in tutorielle Aktionen eines ILS sinnvoller erscheint, da sich Ziele klarer artikulieren und abgrenzen lassen als die Phasen einander überlappender Handlungseinheiten.

Zweitens erscheint das Faktenwissen gegenüber dem Handlungswissen nachgeordnet. Das Ergebnis eines Lernprozesses ist in erster Linie eine innere Handlungsstruktur und erst in zweiter Linie eine Wissensstruktur. Andererseits sind beide Wissensformen eng miteinander verbunden. In einem ILS sind daher das Handlungswissen (Handlungsziele und -pläne) und das deklarative Wissen als Einheit zu repräsentieren[3].

Drittens ist eine Lernhandlung *zielgerichtet*. Die Lernende antizipiert sowohl in der Test- als auch in der Orientierungsphase ein Ziel; zur Überprüfung des Erfolgs werden Zwischenziele generiert.

Im Zusammenhang mit den Bemerkungen in Abschnitt 2 und bezogen auf eine handlungsorientierte Unterstützung von Lernprozessen erscheint die Konzentration auf die **Ziele einer Handlung** sinnvoll. Ziel eines gelenkten Lernprozesses soll die Förderung des Aufbaus solcher Handlungsgrundschemata sein, die bezogen auf den aktuellen Lerngegenstand adäquat strukturiert sind und gleichzeitig im Sinne der Transferierbarkeit allgemeine Konzepte beinhalten. Bezugspunkt von "Lehranweisungen" sind von der Sache her wünschenswerte *Handlungsziele*, die eine *Lernhandlung* initiieren. Diese *Lernhandlungsziele* sind so zu formulieren, zu wählen und anzuordnen, daß das resultierende beobachtbare Verhalten – die Handlung – den Schluß auf ein passendes Handlungsschema zuläßt.

[3] Aus der Perspektive der Repräsentation könnte dieser Ansatz als eine Erweiterung des Ansatzes von Anderson angesehen werden (vgl. [3], [2]), in dessen LISP-Tutor deklaratives und prozedurales Wissen getrennt und prozedural-flach abgelegt ist.

4 Übertragung der Lernhandlungszielorientierung auf ILS

Die klassische Situation in der computergesteuerten Lehre ist die 1:1 – Unterrichtssituation. Die Lernende erwartet von dem System individuelle Betreuung, Hilfestellung und Führung im Lernprozeß. Andererseits geben das Wissensgebiet und didaktische Entscheidungen eine günstige, zum Teil notwendige Strukturierung des Lernprozesses vor: Die Formulierung von Lernhandlungszielen zu einem Wissensgebiet ist eine didaktische Entscheidung, die von erfahrenen Pädagogen getroffen wird. Lernhandlungsziele repräsentieren eine didaktische Aufbereitung des Expertenwissens so, daß die lernrelevanten Wissenseinheiten besonders hervorgehoben werden und weniger lernrelevantes Wissen in den Hintergrund tritt.

Um eine größtmögliche Individualität im Lernprozeß zu sichern, aber dennoch den Aufbau von Handlungswissen zu fördern, welches der Struktur des Wissensgebietes angemessen ist, soll eine **dynamische** Auswahl und Sequenzierung von Lernhandlungszielen ermöglicht werden. Dazu werden über der Menge der Lernhandlungsziele Relationen definiert, die unterschiedliche Zusammenhänge zwischen Lernhandlungszielen herstellen. Zusätzlich werden diese Relationen mit Attributen belegt, die erst in einer aktuellen Sitzung mit Hilfe der Informationen aus dem Studentenmodul und den Informationen aus dem Wissensmodul bewertet werden. Diese Attributbewertung bildet die Grundlage für die Konstruktion eines individuellen Handlungsplanes zu einem Lernhandlungsziel sowie für die Entscheidung, welches Lernhandlungsziel als nächstes anzubieten ist. Eine eindeutige, dem Wissen der Lernenden angepaßte Lernsequenz, entsteht also erst in der Interaktion der Lernenden mit dem System.

4.1 Entwurf einer relationalen Handlungszielstruktur

Sei im folgenden eine Menge von Handlungszielen über einem beliebigen Wissensgebiet vorausgesetzt[4]. Zwischen Handlungzielen lassen sich sachgebietsunabhängig **formale** Relationen definieren: Eine Relation ist die *ist_Voraussetzung_von* – Relation, bei Lesgold die *prerequisites* genannt ([21], S. 135). Diese Relation verknüpft genau jene Wissenseinheiten aus dem Sachgebiet mit dem Handlungsziel, die notwendig gekonnt (gewußt) werden müssen, um das Ziel erreichen zu können. Instanzen dieser Relation über einem konkreten Sachgebiet werden allein aus dem Sachgebiet heraus spezifiziert.

Um die Lernende bei der Entwicklung *ihres* Handlungsgrundschemas unterstützen zu können, definiere ich als zusätzliche Relation die *hat_als_Teilhandlungsziel* - Relation. Diese Relation stellt die Beziehung zwischen einem Handlungsziel und seinen möglichen Teilhandlungen her. Mit Hilfe dieser Relation werden zu dem Handlungsziel gehörende, möglicherweise mehrere unterschiedliche Handlungsschemata repräsentiert. Die Entscheidung, welche in der Lernsitzung aktualisiert wird, hängt vom Verhalten der Lernenden ab.

Aus der handlungstheoretischen Position heraus hängt Lernerfolg im besonderen Maße von der "Wandlungsfähigkeit" von Handlungsstrukturen ab (vgl. [27], S.99 f), d.h. inwieweit bereits erworbene Handlungsstrukturen von der Lernenden an neue Lernsituationen/ Lernproblemstellungen angepaßt werden können. Da diese Anpassungsleistung

[4]Eine Teilmenge von Handlungszielen über dem Wissensgebiet SYPROS ist in Abb. 4 gegeben.

selbst eine Lernleistung ist, wird eine solche *abstrakte* oder auch *generalisierte* Handlungsstruktur explizit als *Handlungsziel* definiert. Da der Begriff der Abstraktion relativ ist, wird entsprechend eine Relation *ist_Abstraktion_von* definiert[5].

Eine weitere Relation bezieht sich, wie auch die *ist_Abstraktion_von* – Relation, hauptsächlich auf den transfertheoretischen Aspekt von Lernen. Sie betrifft die Frage der gegenseitigen *Beeinflussung* von Handlungsstrukturen, d.h. inwieweit unterstützt der Erwerb einer Handlungsstruktur den Erwerb anderer, neuer Handlungsstrukturen, oder welche Handlungsstrukturen erschweren oder verhindern gar neues Lernen. Ich nenne sie die *wird_beeinflußt_von* – Relation. Diese Relation erhält ein Attribut zur Bewertung des Vorzeichens und der Stärke der Beeinflussung. Ein Handlungsziel kann ein anderes positiv oder negativ beeinflussen, der Einfluß kann stark oder schwach sein[6], Grad und Vorzeichen der Beeinflussung können sich mit dem Lernfortschritt der Lernenden verändern. Interessant ist, daß eine Kombination von Handlungszielen insgesamt ein anderes Ziel positiv beeinflussen kann, während die einzelnen Handlungsziele der Kombination eher neutral oder negativ einwirken[7].

Mit diesen Relationen sind lerntheoretisch bedeutsame Aspekte in einer Lernhandlungszielstruktur repräsentiert:

(1) zu jedem Lernhandlungsziel läßt sich mindestens ein passendes Handlungsgrundschema entwerfen $\longleftrightarrow$ *ist_Teilhandlungsziel_von* – Relation;

(2) die Orientierung des Lernprozesses auf den Erwerb "wandlungsfähiger" Handlungsschemata wird ermöglicht $\longleftrightarrow$ *ist_Abstraktion_von* – Relation;

(3) positive Transfereffekte können erzeugt, negative verhindert werden $\longleftrightarrow$ *wird_beeinflußt_von* – Relation.

Die Instanziierung dieser Relationen und die Initialisierung ihrer Attribute über einem konkreten Wissensgebiet ist zunächst noch Aufgabe des Autors der Handlungsstruktur.

Um diese relationalen Beziehungen in einer Lernsitzung zur individuellen Unterstützung in Form geeigneter **Handlungsziel-Angebote** nutzen zu können, müssen auch die Handlungsziele bezogen auf das aktuelle Wissen der Lernenden beurteilt werden können. Dazu werden auch die Handlungsziele mit *Attributen* belegt. Von besonderer Bedeutung für die Entscheidung über ein nächstes Ziel ist die *Komplexität* eines Handlungszieles und die Sicherheit, mit der die Lernende die Handlung beherrscht. Die *Komplexität* eines Zieles ist eine Abschätzung für den *Lernaufwand*, der notwendig ist, um ein zum Ziel gehöriges Handlungsschema zu erwerben. Zur formalen Abschätzung der *Komplexität* eines Handlungszieles kann seine Zerlegung in die *Teilhandlungsziele* sowie die Anzahl seiner *Voraussetzungen* genutzt werden: Ein Ziel ist um so komplexer, je mehr ineinander geschachtelte Teilhandlungsziele zu erreichen und je mehr Voraussetzungen zu erfüllen sind. Zudem erhöht auch die Verschachtelungstiefe den Lernaufwand und damit die Komplexität. Abbildung 1 illustriert dies.

Die *Komplexität* der Handlungsziele verändert sich während einer Lernsitzung: Hat die Lernende erste angebotene Handlungsziele bearbeitet und erfüllt, so hat sie neues Hand-

[5]Gleiche Konzepte finden sich auch bei Anderson, der sie *generalization* bzw. *discrimination* nennt ([3], S. 300 und 304). Die mit der Fähigkeit zur *discrimination* verbundenen notwendigen Konkretisierungen von Handlungszielen steckt als inverse Relation in der *ist_Abstraktion_von* – Relation.

[6]siehe z.B. die Ziele *HZ 4* und *HZ 5* im SYPROS Beispiel Abb. 4

[7]Ein Beispiel aus dem Lerngebiet von SYPROS ist das Problem der richtigen Initialisierung eines Semaphors: Werden diesbezüglich zu oft Aufgaben vom gleichen Typ (Initialisierung mit 0) gestellt, so ist negativer Transfer auf Aufgaben zu vermuten, in denen ein Semaphor ungleich Null zu initialisieren ist. Wurden mehrmals Probleme mit unterschiedlichen Initialisierungen gestellt, so läßt der negative Transfereffekt nach.

lungswissen erworben, vormals Einzelhandlungen sind zu einer neuen Handlungseinheit verschmolzen (vgl. auch "compilation" bei Anderson [3]). Dieses neue Handlungwissen kann nun Teilhandlung eines anderen Handlungsziels HZ_j sein. Dort repräsentiert es jedoch nicht mehr eine zu lernende Handlung, sondern eine Voraussetzung. Der individuelle Lernaufwand für das Ziel HZ_j reduziert sich damit. In Abbildung 1 ist die Reduktion bildlich durch die gestrichelte Umrandung skizziert. Um solche individuellen Lernfortschritte berücksichtigen zu können, werden zu jedem Lernhandlungsziel die Attribute *Erfülltheit* und *Zielsicherheit* ($zs(HZ_j)$) definiert. Das Attribut *Erfülltheit* bestimmt, ob ein bearbeitetes Handlungziel vollständig richtig gelöst wurde, beziehungsweise wieviel Prozent richtig gelöst wurden. Das Attribut *Zielsicherheit* definiert einen Schätzwert für das System, wie gut die Lernende die neu erworbene Handlung beherrscht.

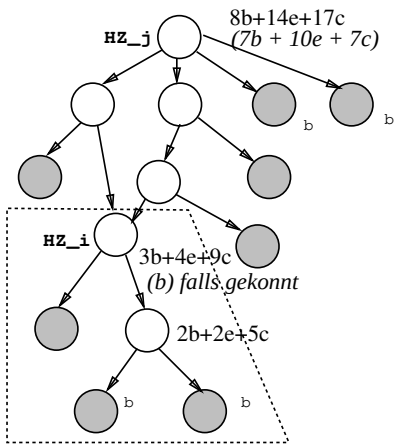


Abbildung 1: Bsp. Teilhandlungsziel-Graph

Zur Berechnung der *Komplexität* eines Handlungszieles HZ_j wird die *ist_Teilhandlungziel_von* – Relation zu HZ_j als azyklischer, gerichteter Graph repräsentiert und aus diesem folgende Größen herangezogen:

(1) die Anzahl der inneren Knoten und die Anzahl der Blattknoten: Letztere repräsentieren gerade die *Voraussetzungen* von HZ_j, während die inneren Knoten mögliche Teilhandlungsziele repräsentieren. Es wird angenommen, daß die Komplexität eines Blattzieles gleich einer Konstanten b ist;

(2) die Anzahl der Verzweigungen an jedem inneren Knoten HZ_i : Dies ist eine Abschätzung für den Zerlegungsaufwand, den die Lernende betreiben muß, um HZ_i lösen zu können ("decompiling" in der ACT* – Theorie von Anderson [3]);

(3) die Tiefen, die durch die inneren Knoten erzeugt werden: Jeder Knoten, der eine neue Ebene öffnet, impliziert zusätzlichen Zerlegungsaufwand und erhöht somit die Komplexität von HZ_j. Für die *Zielsicherheit* $zs(HZ_j)$ nehme ich eine Bewertung zwischen 0 und 1 an[8]. Es ergibt sich folgende Berechnung:

$$Kompl(HZ_j) :=$$

$$\begin{cases} \sum\limits_{\substack{HZ_i \\ \in succ(HZ_j)}} Kompl(HZ_i) + (1 - zs(HZ_j))(d^+(HZ_j) * e + w(HZ_j) * c), & 0 \leq zs(HZ_j) < 1; \\ b \quad , zs(HZ_j) = 1 \text{ oder } d^+(HZ_j) = 0 \text{ oder } (zs(HZ_j) = 0 \text{ und } HZ_j \text{ schon bewertet}) \end{cases}$$

wobei $succ(HZ_i)$ die Menge der direkten Teilhandlungen von HZ_i; $d^+(HZ_i)$ und $d^-(HZ_i)$ der Ausgangsgrad bzw. Eingangsgrad des Knotens HZ_i; $b,\ e,\ c \in \mathbb{N}^+$;
$w(HZ_i)$ ist das *Gewicht* von HZ_i im Teilhandlungszielgraph $G(HZ_j)$: $w(HZ_j) := 1$ und $w(HZ_i) := w(pred(HZ_i)) + 1$ für alle inneren Knoten und $d^-(HZ_i) = 1$;
$w(HZ_i) := \max\limits_{w}\ w(HZ_i)$, falls $d^-(HZ_i) > 1$.

Der Fall $Kompl(HZ_i) = b$, falls HZ_i schon bewertet ist, sowie die Berechungsvorschrift für $w(HZ_i)$ für $d^-(HZ_i) > 1$ definiert die Bewertung der *Komplexität* von HZ_j in dem Falle, daß ein Handlungsziel HZ_i mehrfach als Teilhandlungsziel von HZ_j vorkommt.

[8]Basis ist die Häufigkeit des "Erfüllens" eines Handlungszieles in den letzten t Versuchen (Trend).

Es erscheint damit einerseits in unterschiedlichen Kontexten, andererseits fordert es von diesem Kontext losgelöst eine identische Handlungsweise. Der Lernaufwand basiert in einem solchen Fall auf den einmaligen (!) Erwerb dieser Handlungsweise und auf dem Durchführungsaufwand in unterschiedlichen Kontexten. Ist $zs(HZ_i) = 0$ für alle HZ_i im Graph zu HZ_j dann ist die *Komplexität* maximal und entspricht gerade dem Lernaufwand, den die Lernende bewältigen muß, wenn sie über das zu bearbeitende Ziel noch nichts weiß. In Abbildung 1 ist an den ausgezeichneten Knoten die Komplexität angegeben. Die Werte in Klammern repräsentieren die Komplexität im Falle $zs(HZ_i) = 1$.

Die individuelle Bewertung der *Komplexität* ermöglicht dem System die Einbeziehung des diagnostizierten Vorwissens in die Entscheidung über die Auswahl des Handlungsziel – Angebotes: Um eine Unterforderung der Lernenden zu vermeiden, kommen nur solche Ziele als mögliche nächste Ziele in Betracht, deren *Komplexität* mindestens gleich oder höher ist als die der bereits bearbeiteten Ziele. Auch beim Wechsel zu einem anderen Themenbereich können die Werte der Komplexität zur Wahl eines geeignet komplexen Handlungszieles führen. Man kann als Auswahlkriterium einen Vergleich der bereits bewältigten maximalen *Komplexitäten* ($zs(HZ_i) = 0$) mit Schätzwerten für die im neuen Themenbereich zu bewältigenden maximalen *Komplexitäten* heranziehen. Um die Handlungsziele auch thematisch miteinander vergleichen zu können, schlage ich zusätzlich eine Klassifikation der Handlungsziele nach *Lernbereichen* vor.

4.2 Dynamische Navigation durch die Handlungszielstruktur

Ziel einer dynamischen Navigation durch die formale Handlungszielstruktur zu einem Wissensgebiet ist die Instanziierung einer individuellen Handlungszielstruktur, bezogen auf das individuelle Können der Lernenden. Diese soll während der Lernsitzungen der Lernenden mit dem System sukzessiv aufgebaut werden, d.h. zu jedem Zeitpunkt, an dem die Lernende ein Handlungsziel bearbeitet hat, wird die gesamte Handlungszielstruktur neu bewertet und aufgrund dieser Bewertungen aus der Menge aller potentiellen "nächsten" Handlungsziele (das sind alle noch nicht bearbeiteten Handlungsziele) diejenigen ausgesucht, deren Bearbeitung zu diesem Zeitpunkt sinnvoll und erfolgreich erscheint. Hilfreich wäre eine Information aus dem Studentenmodul darüber, wie die Lernende hinsichtlich ihrer kognitiven Fähigkeiten (z.B niedrig, mittel, hoch) oder nach dem Niveau (Anfänger, Fortgeschrittener, Experte) eingeschätzt wird (die Diplomarbeit von Andreas Harrer [13] zeigt hierzu eine Lösung).

Auf der Basis der im vorangegangenen Abschnitt definierten Attribute und Relationen kann nun mittels einer Menge von Regeln **während der Lernsitzung** eine an das Können der Lernenden angepaßte Handlungsstruktur entwickelt werden. Beispiele für Regeln sind:

1. WENN L(ernender) "Fortgeschrittener" UND L hat das letzte Ziel nicht richtig bearbeitet, DANN suche in den Teilhandlungen nach den nicht richtig bearbeiteten UND dazu die Abstraktion UND gebe dieses Teilziel als Handlungsziel;

2. WENN L "Anfänger" UND L hat das letzte Ziel nicht richtig bearbeitet, DANN suche in den Teilhandlungen nach den nicht richtig bearbeiteten UND gebe eine (Fehler)Erklärung UND dieses Ziel als neues Handlungsziel;

3. WENN L "Fortgeschrittener" UND L das letzte Ziel erreicht hat UND die Zielsicherheit gross ist UND der aktuelle Lernbereich hinreichend bearbeitet ist, DANN wähle ein neues Ziel von (höherer ODER gleicher) maximaler Komplexität UND aus einem anderen Lernbereich;

4. WENN L "Fortgeschrittener" UND L das letzte Ziel erreicht hat UND die Zielsicherheit groß ist UND der aktuelle Lernbereich nicht hinreichend bearbeitet ist, DANN wähle als nächstes ein Ziel aus, welches von dem vorangegangenen positiv beeinflußt wird;
5. WENN L "Fortgeschrittener" UND L das letzte Ziel erreicht hat UND die Zielsicherheit niedrig ist, DANN wähle ein Ziel welches das letzte Ziel als Teil enthält UND eine höhere Komplexität hat;
6. WENN L "Anfänger" UND L das letzte Ziel erreicht hat UND die Zielsicherheit 'sehr niedrig' ist, DANN wähle eine weitere Aufgabe zu diesem Ziel;
7. WENN L "Anfänger" UND L das letzte Ziel erreicht hat UND die Zielsicherheit ' niedrig – mittel' ist, DANN wähle ein neues Ziel mit höchstens gleicher Komplexität;
8. WENN L "Fortgeschrittener" UND L hat das letzte Ziel nicht erreicht UND es gibt ein Ziel, welches schon erfüllt wurde UND die Ziele viele Teilhandlungen identisch haben, DANN gebe Hinweis auf dieses Ziel;
9. usw.

Diese Regeln können sowohl auf der Bedingungs- als auch auf der Aktionsseite beliebig komplex werden und somit die Individualisierung des Lernprozesses ermöglichen.

Hier ist es wichtig zu bemerken, daß diese Regeln allein auf der Basis der formalen Struktur (vgl. Abschnitt 1) von Handlungszielen formuliert werden können. Welche Regel "feuert", hängt von der Anzahl der einbezogenen Attribute und deren aktueller Bewertung ab. Dies wird allerdings vom Wissen beeinflußt, das das System über die Lernende hat. Existiert ein "poor model", so werden sich die Auswahlen, die das System trifft, von Individuum zu Individuum kaum unterscheiden. Andererseits ermöglicht ein detailliertes Studentenmodul auch eine detaillierte Bewertung der Regelbedingungen.

5 Umsetzung am Beispiel SYPROS

In diesem Abschnitt soll das oben beschriebene **sachgebietsunabhängige Konzept** an einem Beispiel-ITS konkretisiert werden. Dieses Beispiel-ITS ist SYPROS, welches an der TU München unter der Leitung von Christian Herzog entwickelt wird.

5.1 Das Sachgebiet von SYPROS und seine Wissensrepräsentation

Das Sachgebiet von SYPROS[9] ist das Lösen von Synchronisationsproblemen mit Semaphoren (vgl. [4]). "Ein Semaphor `s` ist eine Variable, die nur nicht-negative ganze Zahlen als Werte annehmen kann. Nach der Initialisierung sind nur noch zwei Arten von Operationen darauf erlaubt: `V(s)` erhöht den Wert der Variablen um 1, `P(s)` erniedrigt ihn entsprechend. Ist der Wert von `s` Null, dann wird der Prozeß, der `P(s)` ausführen will, solange wartend gesetzt, bis ein anderer Prozeß `V(s)` ausgeführt hat" [16]. Zur Lösung eines Synchronisationsproblems müssen im Prinzip drei Typen von Restriktionen und deren Kombinationen erfüllt werden: Dies sind die Restriktion des *gegenseitigen Ausschlusses* von Prozessen beim Zugriff auf gemeinsame Ressourcen und die Restriktionen des *Zugriffes auf 'leere' bzw. 'volle' Ressourcen.* In der Sprache von SYPROS werden diese die `MutuallyExclusive-`, `GetOnlyIfNotEmpty-` und `BenchLimit`-Restriktion genannt. Ein Beispiel für eine Aufgabenstellung und ihre Lösung zeigt Abbildung 2.

[9] Ausführlichere Informationen zu SYPROS finden sich z.B. in [14],[15], [18], [17].

```
 empty:=0; limit:=5; mutex:=1; Bench B:=0;

prozess HER;              prozess VER;
 Forever Do                Forever Do
 {                         {
   1:<<erzeuge>>;          P(empty);
    P(limit);              P(mutex);
    P(mutex);             1:get(B);
   2:put(B);               V(limit);
    V(empty);              V(mutex);
    V(mutex);}            2:<<verbrauche>>;
 }                         }
 end;                      end;

MtuallyExklusive(HER:2, VER:1);
GetOnlyIfNotEmpty(B);
BenchLimit(B,5);
```

Abbildung 2: Beispielaufgabe in SYPROS

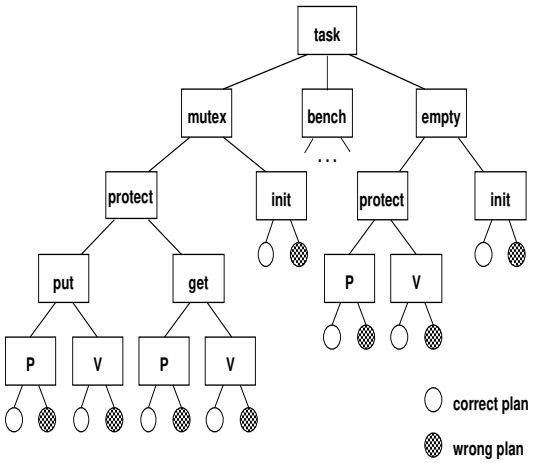


Abbildung 3: Ziel-Plan-Baum in SYPROS

In SYPROS ist das Wissen zur Lösung einer Aufgabe in einem erweiterten **Ziel-Plan**-Baum repräsentiert. Die inneren Knoten des Baumes repräsentieren das konzeptuelle Wissen, wie man vom Problem zum Programm kommt, die Blattknoten repräsentieren die Umsetzung des Konzeptwissens in eine Anweisung oder Operation der Programmiersprache. Abbildung 3 skizziert den **Ziel–Plan**–Baum zur obigen Aufgabe.

5.2 Beispiel: Entwurf einer möglichen Handlungszielstruktur

Falls – wie im Beispiel SYPROS – das Sachgebietswissen in einem **Ziel–Plan**–Baum repräsentiert ist und zudem die Generierung dieses Baumes dem menschlichen Problemlöseprozeß (vgl. das 2 - Phasen – Modell von SYPROS (z.B. [14])) sehr ähnlich ist, lassen sich die Ziele des Ziel–Plan–Baumes bereits als Handlungsziele formulieren. Der Knoten `mutex` repräsentiert das Handlungsziel `Lösen der Restriktion MutuallyExclusive`, die Knoten `empty` und `bench` entsprechend die Handlungziele zum Lösen der beiden anderen Restriktionen. Der Wurzelknoten `task` repräsentiert eine Inkarnation des Handlungszieles HZ_0 `Verknüpfe die zu lösenden Restriktionen richtig`. In Abbildung 4 ist dieses als Beispiel eines zu bearbeitenden Handlungszieles mit seinem Handlungsgrundschema dargestellt. Im Vergleich des Ziel-Plan-Baumes mit dieser Struktur wird deutlich, daß die Knoten des Ziel-Plan-Baumes allein nicht ausreichen, um das zum Handlungsziel gehörende Handlungsschema zu modellieren. Um nämlich eine Aufgabe von diesem Typ lösen zu können, muß die Lernende neben den Basisschemata lernen, daß die Reihenfolgen, in der die P-Operationen gesetzt werden, richtig zu setzen sind. Ziel HZ_4 bzw. HZ_5 repräsentieren das Reihenfolgenproblem für die Kombinationen `GetOnlyIfNotEmpty` bzw. `BenchLimit` mit `MutuallyExclusive`. Die zu diesen Zielen gehörende Handlungsschemata sind quasi identisch, d.h. wenn eines richtig gemacht wurde, dann erleichtert dies das Erlernen des jeweils anderen Schemas. Zusätzlich muß die Lernende noch lernen, daß in einer solchen Situation für jede Restriktion ein gesondertes Semaphor zu initialisieren ist (HZ_1) und sie muß diese auseinanderhalten können (HZ_3). In der Abbildung 4 sind die unterschiedlichen Zusammenhänge, in denen die Handlungsziele stehen, an den Kanten

markiert.

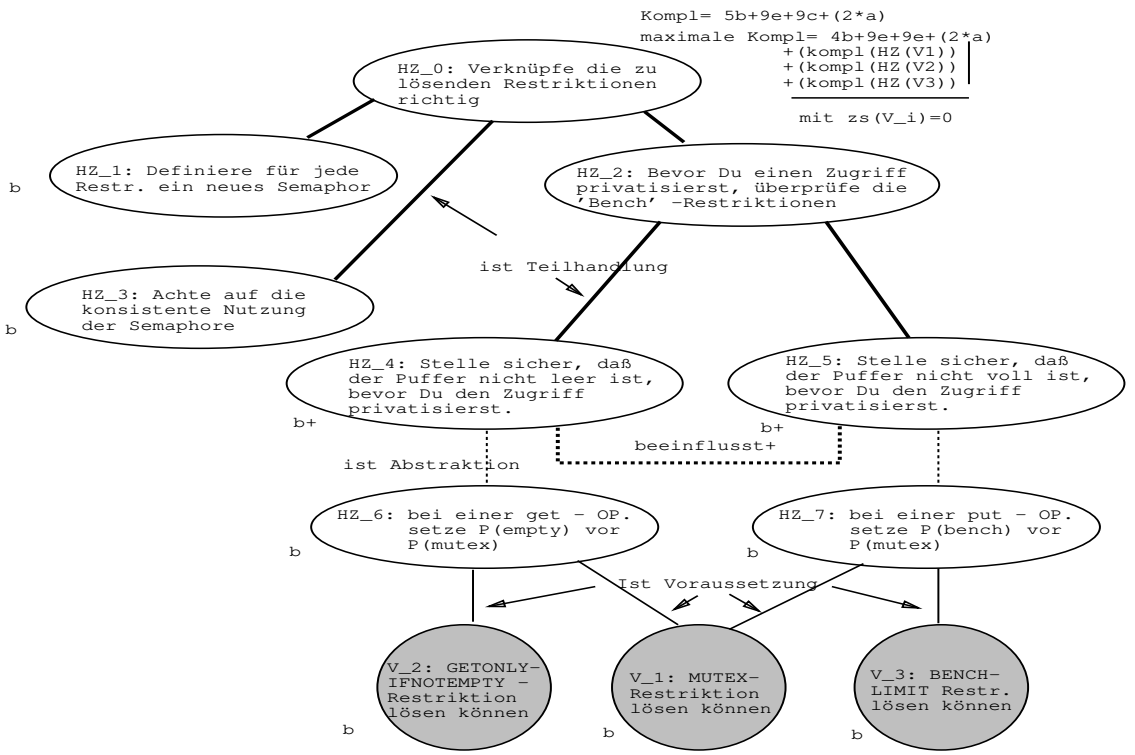


Abbildung 4: Ein Ausschnitt aus einer Handlungszielstruktur

5.3 Beispiel für eine Entscheidung

Nehmen wir an, die Lernende hätte bisher nur Probleme bearbeitet, in denen die Restriktionen – versteckt in den Blattknoten V_2, V_1, V_3 – als Handlungsziele isoliert behandelt wurden. Ihre Sicherheit sei dort als hoch (=1) eingestuft. Die Komplexität der Restriktionen ist dann b, während sie vorher deutlich größer war. Damit reduziert sich die Komplexität des Zieles HZ_0, wie in der Abbildung angedeutet. Das System interpretiert diese Aufgabe nun für die Lernende als "leicht" und bietet es als Ziel an. Die Lernende nehme das Angebot an und löse das Problem fast richtig: Es sei nur die Reihenfolge der `P` – Operationen im Verbraucherprozeß vertauscht. D.h. die Voraussetzungen sowie die Ziele HZ_1, HZ_3, HZ_5 sind erfüllt, nicht aber HZ_4. HZ_4 wird positiv beeinflußt von HZ_5. Damit könnte z.B. Regel 4 Seite 8 greifen: Das System könnte die Lernende mit dem Hinweis auf das bereits richtig gelöste gleichartige Problem im Herstellerprozeß hinweisen, anstatt unabhängig von dem als richtig diagnostizierten Wissen eine Fehlermeldung zu geben.

6 Zusammenfassung und Ausblick

In den Abschnitten 2 und 3 wurde dargelegt, daß die Verbindung eines lerntheoretischen Ansatzes, der Handlungstheorie mit einer bestimmten Lehrmethode, der Lernzielorientierung, zu einer sinnvollen Unterstützung von Lernen werden kann. Im Abschnitt 4 wurde

gezeigt, daß eine Übertragung dieser Strategie auf ILS möglich ist und zur Individualisierung der "Lernerführung" beitragen kann. Zudem kann diese Strategie weitestgehend sachgebietsunabhängig entwickelt werden: Es kann eine formale, sachgebietsunabhängige Struktur als Muster für die Erzeugung von Handlungszielstrukturen über einem bestimmten Sachgebiet angegeben werden.

Bei dem Versuch, diese Idee an einem Beispiel umzusetzen, zeigt sich, daß schon durch die Formulierung weniger Relationen ein komplexes Geflecht von Handlungszielen entstehen und zusammen mit unterschiedlichen Attributen die Tutor-Aktionen individualisieren helfen kann. Andererseits zeigt sich auch, daß eine sehr genaue Analyse von möglichen, zielorientierten Interaktionsszenarien und Situationen (z.B. Einstiegssituation vs. Problemlösesituation, Interaktion nach Fehlern vs. Interaktion nach Erfolg ..) notwendig ist, da sonst die Gefahr besteht, daß das Regelwerk zu undifferenziert (und falsch) ist. Ein weiterer aufwendiger Punkt ist die formale Beschreibung der Relationen, welche zur Zeit noch nicht abgeschlossen ist. Hier besteht auch der Anspruch, die Anzahl der Relationen möglichst gering zu halten, um möglichst wenig restriktiv zu sein, andererseits möglichst viele Informationen aus dem Studentenmodul zur Bewertung der Beziehungen heranzuziehen, um die Adaptivität des Systems an die Lernende zu erhöhen.

Literatur

[1] John R. Anderson (Hrsg.). *The architecture of cognition.* Harvard University Press, 1983.

[2] John R. Anderson (Hrsg.). *Rules of the Mind.* Lawrence Erlbaum Associates, 1993.

[3] J.R. Anderson. *Knowledge Compilation: The general learning mechanism*, Kapitel 11, S. 289–319. In Michalski [23], 1986.

[4] E.-W. Dijkstra. *Cooperating Sequential Processes.* In Genuys [8], 1968.

[5] R. Gunzenhäuser et. al. 7. Arbeitstreffen der GI–Fachgruppe „Intelligente Lernsysteme". Technischer Bericht TR 94003, FAW Ulm, Mai 1994.

[6] Jürgen Folckers, Claus Möbus, Klaus Pitschke, Olaf Schröder. PETRI-HELP: Expertiseerwerb, Hilfeakzeptanz und Systemerweiterungen. In *7. Arbeitstreffen der GI–Fachgruppe „Intelligente Lernsysteme"* [5].

[7] Helmar Frank. *Kybernetik – Brücke zwischen den Wissenschaften.* Umschau -Verlag, 7. Auflage, 1970.

[8] F. Genuys (Hrsg.). *Programming Languages.* Academic Press, 1968.

[9] Eugene Galanter Georg A. Miller, Karl H. Pribram. *Strategien des Handelns – Pläne und Strukturen des Verhaltens.* 1973.

[10] Peter Gerjets, Elke Heine, Rainer Westermann. Diagnostik von Handlungsphasen in tutoriellen Lernsystemen. In F. Huber-Wäschle, H. Schauer, P. Widmayer (Hrsg.), *GISI 95 Herausforderungen eines globalen Informationsverbundes für die Informatik*, S. 330 – 337. Springer, 18 – 20. September 1995.

[11] Rul Gunzenhäuser, Hainz Mandl (Hrsg.). *5. Workshop Intelligente Tutorielle Systeme der GI-Fachgruppe "Intelligente Lehrsysteme"*, Juni 1993.

[12] Hans Hannapel. *Lehren lernen.* Kamps pädagogische Taschenbücher Bd. 93. Ferdinand Kamp, 1988.

[13] Andreas Georg Harrer. Ein didaktisches Konzept für die Lernerführung in einem Intelligenten Lehrsystem. Diplomarbeit, Technische Universität München, TU München, Institut für Informatik, Februar 1996.

[14] Christian Herzog. From Elementary Knowledge Schemes Towards Heuristic Expertise — Designing an ITS in the Field of Parallel Programming. In *2nd International Conference on Intelligent Tutoring Systems*, S. 183–190, Juni 1992.

[15] Christian Herzog. SYPROS – Ein intelligentes Lehrsystem auf dem Gebiet paralleler Programmierung. In Gunzenhäuser, Mandl [11], S. 2.1 – 2.10.

[16] Christian Herzog. Wissensrepräsentation, Diagnose und Erklärungen im intelligenten Lehrsystem SYPROS. Unveröffentlichtes Manuskript, 1997.

[17] Christian Herzog, Meike Gonschorek. Generation of Adaptive Explanations in ITS with Hierarchical Knowledge Domains. In *International Conference on Computers in Education*, S. 1 –6, Dezember 1993.

[18] Christian Herzog, Daniela Rothenhöfer. An Architecture for an Intelligent Tutoring System in a Complex Domain. In *World Conference on Artificial Intelligence in Education*, August 1993.

[19] Gerhard Kaminski. *Überlegungen zur Funktion von Handlungstheorien in der Psychologie*, S. 93 – 121. Band III von Lenk [20], 1981.

[20] Hans Lenk (Hrsg.). *Handlungstheorien – interdisziplinär*, Band III. Wilhelm Fink-Verlag, 1981.

[21] Alan Lesgold. *Towards a Theory of Curriculum for Use in Designing Intelligent Instructional Systems*, S. 114 – 137. In Mandl, Lesgold [22], 1988.

[22] Heinz Mandl, Alan Lesgold. *Learning Issue for Intelligent Tutoring Systems.* Springer, 1988.

[23] R.S. Michalski (Hrsg.). *Machine Learning: An Artificial Intelligence Approach II.* Morgan Kaufmann, 1986.

[24] J.J.G. van Merriënboer S. Dijkstra, H.P.M. Krammer (Hrsg.). *Instructional Models in Computer-Based Learning Environments*, Band Serie F, Nr. 104 von *NATO ASI Series.* Springer, 1992.

[25] Wolfgang Schnotz. *Lerndiagnose als Handlungsanalyse.* Beltz – Verlag, 1979.

[26] Kurt van Lehn. *Toward a Theory of impasse-driven learning*, S. 19 – 41. In Mandl, Lesgold [22], 1988.

[27] Carel F. van Parreren. *Lernprozeß und Lernerfolg.* Westermann, 1972.

[28] M.M Waldrop. SOAR: A Unified Theory of Cognition? *Science*, 241:296 – 298, Juli 1988.

Kognitive und motivationale Gesichtspunkte in einer Lehrstrategie

Andreas Harrer
Institut für Informatik
Technische Universität München
80290 München
Tel. +49 (89) 48095 - 152
E-mail: harrer@informatik.tu-muenchen.de

Zusammenfassung

Die Lehrstrategien in intelligenten Lehrsystemen berücksichtigen oft die individuellen Eigenschaften des Tutanden nicht in ausreichendem Maße. Aber die Betrachtung von kognitiven und auch motivationalen Eigenschaften ist notwendig, um es dem Studenten zu gestatten, seine Lernmöglichkeiten voll auszuschöpfen. Die kognitiven Aspekte müssen bei der Wahl der Genauigkeit von Hilfestellungen berücksichtigt werden. Außerdem benötigen einige Studenten viel Rückmeldung, andere bevorzugen ein herausforderndes Umfeld und andere arbeiten lieber auf sich alleine gestellt. Dies begründet sich in nicht-kognitiven, motivationalen Eigenschaften der Studenten. Im Klassenunterricht (v.a. Frontalunterricht), Vorlesungen oder Übungen ist es wegen der großen Anzahl von Schülern nicht möglich, jeden in der für ihn geeigneten Art und Weise zu betreuen. In intelligenten Lehrsystemen ist allerdings die individuelle, auf den Studenten zugeschnittene, Unterrichtung als Anspruch vorgegeben. Dieser Artikel bemüht sich darum, kognitive und motivationale Gesichtspunkte in eine Lehrstrategie zu integrieren, die zum Ziel hat, für die einzelne Studentin ein angemessenes Lehrverfahren auszuwählen. Außerdem werden die Anforderungen an die Studentenmodellierung dargestellt, damit jene Strategie in einem intelligenten Lehrsystem umgesetzt werden kann. Schließlich wird die Umsetzung der Lehrstrategie im intelligenten Lehrsystem SYPROS beschrieben und zukünftige Entwicklungsmöglichkeiten aufgezeigt.

Einleitung

Im Bereich der intelligenten Lehrsysteme ist das Tutormodul ein wichtiger, aber dennoch oft vernachlässigter Bestandteil. Ein Grund dafür ist wohl, daß die Didaktik eine sogenannte „weiche Wissenschaft“ ist, d.h. absolute Wahrheiten sind nicht formalisierbar und eine Vielzahl von nebeneinander existierenden und teilweise widersprechenden Theorien versucht, die Gesamtheit von Lehren und Lernen auszuleuchten. Aufgabe des Tutormoduls eines intelligenten Lehrsystems

ist es, die Kompetenz eines menschlichen Lehrers möglichst gut nachzubilden und auf den Schüler so individuell einzugehen, wie es ein Lehrer tun würde. Da Lehrer aber selten eine festgeschriebene Strategie verfolgen (oder es zumindest schwerfällt eine zu formalisieren), ist es schwierig, nach dem Vorbild von menschlichen Lehrern eine Tutorkomponente zu entwickeln, die sich an den jeweiligen Schüler anpaßt und geeignete Lehrmaßnahmen ergreift. Durch diese Unklarheit, wie eine Lehrstrategie aufgebaut sein soll, entstanden diverse Konzepte des Unterrichtens in intelligenten Lehrsystemen, wie z.B. der „Instructional Planning Approach“ [VW96] oder die „Learning Companion Systems“[1] ([CB88], [Cha96]) oder die Lehrstrategie, die in der Implementierung von ACT-R [And93] verwendet wird. Beim Entwurf der lokalen Tutorkomponente, deren Aufgabe es ist, den Dialog mit dem Tutanden nach didaktischen Gesichtspunkten während der Bearbeitung von Aufgaben zu führen, die also Entscheidungen darüber zu treffen hat, in welcher Situation und auf welche Weise die Tutorkomponente eingreifen soll, sind nach [LC88] folgende Gesichtspunkte zu beachten:

- Kontrolle der Intervention: Wer initiiert die Tutorintervention? Das Lehrsystem oder der Tutand?
- Zeitpunkt einer Intervention: Wann soll eine Tutorintervention erfolgen? Sofort nach einem oder nach wiederholtem Fehler oder nach einem Abweichen von der Optimalität oder sogar, wenn das System im nächsten Schritt einen Fehler prognostiziert?
- Inhalt der Intervention: Welcher Art ist die Tutorintervention? Dient sie der Motivationsförderung oder der Fehlerverbesserung?
- Stil der Intervention: Wie ist die Intervention abgefaßt? Ist sie alleine als Informationsmitteilung zu sehen oder sozialer Natur (Motivation, Rückmeldung)?

In diesem Artikel wird der Versuch gemacht, unter Berücksichtigung sowohl kognitiver als auch motivationaler Aspekte, diese Fragen zu beantworten und aus den Ergebnissen Prinzipien für eine Tutorkomponente zu entwickeln. Diese soll nach didaktischen Gesichtspunkten entworfen und in der Lage sein, sich dem Tutanden individuell anzupassen.

Kognitive Aspekte

Ein wichtiger Einflußfaktor bei der Entscheidung, wie das Tutormodul auf die Aktionen der Studentin reagieren soll, ist das Wissen der Studentin. In der Kognitionspsychologie wird der Begriff kognitive Komplexität zur Beschreibung der Struktur und der Beziehungen zwischen einzelnen Wissenseinheiten verwendet [Ein81]. Hat die kognitive Struktur nur wenige Elemente und wenige Verbindungen zwischen ihnen, so ist die Struktur von geringer kognitiver Komplexität. Hat sie hingegen viele Elemente und vielfältige Verknüpfungen, so spricht man von einer Struktur hoher kognitiver Komplexität.

Die Untersuchung der Wirksamkeit von unterschiedlichen Lehrmethoden in [TH71] zeigt folgende Ergebnisse auf:

[1] in diesen Systemen wird dem Schüler ein computergenerierter Mitschüler zur Seite gestellt

- Lehrverfahren, in denen der Lehrer sehr viel darbietet und in denen ein hoher Strukturierungsgrad vorliegt, begünstigen Schüler mit niedriger kognitiver Komplexität[2] und sind schlecht geeignet für Schüler mit hoher kognitiver Komplexität (diese arbeiten lieber selbständig und erbringen weniger Leistung, wenn sie dies nicht können).

- Lehrverfahren, in denen der Lehrer und die Schüler zusammen den Stoff erarbeiten, erbringen gute Resultate für Studenten mit hoher kognitiver Komplexität und schlechtere für jene mit niedriger kognitiver Komplexität.

- entdeckenlassende Lehrverfahren[3] sind gut geeignet für Studenten mit hoher kognitiver Komplexität und überhaupt nicht geeignet für Schüler mit niedriger kognitiver Komplexität. Dies kommt daher, daß diese ohne Anleitung Probleme haben, die Lernerfahrungen in ihre kognitive Struktur zu integrieren.

Motivationale Gesichtspunkte

Kognitive Gesichtspunkte sind jedoch nicht der einzige Einflußfaktor auf die Lernleistung eines Schülers. Ein weiterer Aspekt ist die Motivation, Untersuchungsgegenstand der Motivationspsychologie. Nach [Sch74] und [Hec89] kann Motivation folgendermaßen unterteilt werden:

- **Stärke der Motivation**:
  Hochmotivierte streben höhere Ziele an und bevorzugen eine vom Lehrer unabhängige Arbeitsweise. Sie benötigen wenig Rückmeldung und ihre Leistung läßt unter Druck nach, während weniger motivierte Schüler unter Druck bessere Leistung erbringen.

- **Erwartungsrichtung der Motivation** (Motivdisposition):
  Studenten, die erfolgsmotiviert sind, hoffen auf Erfolg, während mißerfolgsmotivierte Studenten versuchen, Mißerfolg zu vermeiden. Dies drückt sich darin aus, daß Erfolgsmotivierte Problemstellungen suchen, die ihrem Leistungsstand entsprechen und realistisch in der Auswahl ihrer Aufgaben sind. Mißerfolgsmotivierte hingegen vermeiden Situationen, in denen Erfolg und Versagen mit gleicher Wahrscheinlichkeit erwartet werden können. Sie wählen Probleme, die viel zu leicht sind, weil dort Versagen sehr unwahrscheinlich ist, oder Probleme, die viel zu schwer für ihr Leistungsvermögen sind, da hier ein Versagen das Selbstwertgefühl kaum belastet. Außerdem ist Rückmeldung für mißerfolgsmotivierte Studenten sehr wichtig, da die bei fehlender Rückmeldung entstehende Unsicherheit ihre Leistung drastisch abfallen läßt.

Diese Ergebnisse legen nahe, erfolgsorientierte und mißerfolgsorientierte Schüler in ganz unterschiedlicher Weise zu betreuen, insbesondere natürlich in der Um-

[2] im weiteren Verlauf wird unter der kognitiven Komplexität eines Studenten die Komplexität seiner kognitiven Struktur – bezogen auf das betreffende Fachgebiet – verstanden. Als Alternative zu diesem Begriff könnte man auch „kognitive Fähigkeit bezogen auf das Sachgebiet“ verwenden.

[3] die Studentin erkundet das Wissensgebiet selbst; sie kann zwar Informationen und einige Hilfen anfordern, aber das zugrundeliegende Prinzip dieser Lehrverfahren ist die selbständige Aktivität des Lerners

gebung eines intelligenten Lehrsystems mit der Möglichkeit zur individuellen Anpassung der Lehrstrategie an den einzelnen Tutanden. Schwierigkeiten bestehen darin, festzustellen, welche Erwartungshaltung ein Student hat. Der Schwierigkeitsgrad von selbstgewählten Aufgaben im Vergleich zum Wissensstand sowie Ergebnisse der Selbst-Attribution (man fragt die Studentin, ob sie lieber selbständig arbeitet oder nicht) geben wertvolle Anhaltspunkte bezüglich der Erwartungshaltung der Motivation. Als weiteren Hinweis auf die Erwartungsrichtung könnte man die Anzahl von tutandeninitiierten Interventionen heranziehen; ein mißerfolgsmotivierter Schüler wird wohl öfter Interventionen initiieren, weil er auf Rückmeldung bedacht ist, als ein erfolgsmotivierter. Dies wurde aber im weiteren Verlauf nicht in den Betrachtungen berücksichtigt, da solche Indizien für eine bestimmte Erwartungsrichtung nicht fundiert genug sind.

Prinzipien der Lehrstrategie

Nachdem in den vorausgegangenen Abschnitten kognitive und motivationale Gesichtspunkte genannt wurden, die im Tutormodul berücksichtigt werden sollten, kann zu den in der Einleitung aufgeworfenen Fragen über Zeitpunkt, Art und Weise von Tutorinterventionen zurückgekehrt werden. Die Beantwortung jener Fragen ermöglicht es, Prinzipien für eine Tutorkomponente zu skizzieren:

- **Kontrolle**: Wenn die Kontrolle der Intervention [VW96] vom Studenten ausgeht, so liegt eine entdeckenlassende Lehrstrategie vor, die für Schüler mit hoher kognitiver Komplexität gut ist. Kommt die Initiative für Interventionen vom intelligenten Lehrsystem, so befindet man sich in einer strukturierteren Umgebung mit häufiger Rückmeldung, die für mißerfolgsmotivierte Studenten und solche mit niedriger kognitiver Komplexität angemessen ist.

- **Zeitpunkt**: Erfolgsmotivierte Schülerinnen und jene mit hoher kognitiver Komplexität sollten mit weniger tutorinitiierten Interventionen versehen werden, als Schüler mit niedriger kognitiver Komplexität und Mißerfolgsmotivation. Mißerfolgsmotivierte Studenten erbringen bessere Lernleistung, wenn Rückmeldungen stattfinden, und Schüler mit niedriger kognitiver Komplexität könnten inkorrekte Elemente in ihre kognitive Struktur einbauen, wenn von der Tutorkomponente keine Interventionen initiiert werden.

- **Inhalt**: Die Variationsmöglichkeiten für den Inhalt einer Intervention hängen stark von den vom Lehrsystem bereitgestellten Mitteln ab. Interventionen werden gemacht, um Fehler zu korrigieren, die Motivation zu fördern oder um die Integration der gelernten Wissenselemente in die kognitive Struktur zu erleichtern. Wenn ein ILS Abstufungen von Korrekturmechanismen kennt, sollte eine erste Intervention recht ungenau sein (ein Hinweis) und folgende Interventionen zu diesem Problem immer genauer werden. Wurde durch die Studentenmodellierung eine spezielle Schwäche in einem bestimmten Wissenselement festgestellt, sollte früher Hilfestellung geleistet werden. Motivationsfördernde Interventionen sollten die Erwartungshaltung der Studentin berücksichtigen, d.h. erfolgs- und mißerfolgsmotivierte Schülerinnen sollten nicht dieselbe Intervention präsentiert bekommen (s. auch Stil). Ein Mittel zur Integration neuerworbenen Wissens ist

die Präsentation einer Zusammenfassung des Inhalts einer Sitzung mit dem Lehrsystem, nachdem der Student die eigene Arbeit beendet hat[Ein81].

- **Stil**: Erfolgsmotivierte Studenten sollten anspornende und evtl. sogar fordernde Rückmeldung erhalten, wohingegen für mißerfolgsmotivierte Schüler ein lobender und unterstützender Interventionsstil anzuraten ist.

Aus diesen Ergebnissen lassen sich allgemeine Prinzipien für eine Betreuungskomponente entwickeln, die sowohl kognitive, als auch motivationale Gesichtspunkte berücksichtigt:

- Prinzipien basierend auf der kognitiven Komplexität des Tutanden:
  - besitzt der Student eine hohe kognitive Komplexität, so ist ein Lehrverfahren mit hoher Selbstbestimmung und wenigen Interventionen zu wählen;
  - ist die kognitive Komplexität niedrig, dann sollte ein stärker geführter Unterricht mit häufigeren Interventionen und genaueren Hilfestellungen stattfinden.
- Prinzipien basierend auf der Motivdisposition des Tutanden:
  - handelt es sich um einen Erfolgsmotivierten, so ist Rückmeldung nicht so wichtig und der Stil von Interventionen soll den Tutanden zu Leistung anspornen.
  - ist der Tutand mißerfolgsmotiviert, dann ist häufige Rückmeldung wichtig und ein lobender, aufmunternder Stil anzuschlagen, selbst wenn keine Fehler aufgetreten sind.
- allgemeine Prinzipien:
  - Hat der Student in der Vergangenheit bereits größere Probleme mit einem gewissen Wissenselement gehabt und treten erneut Schwierigkeiten damit auf, dann sollte früher und evtl. auch genauer geholfen werden.
  - Dargebotene Hinweise sollten zunehmende Genauigkeit aufweisen, wenn sie sich auf denselben Problempunkt beziehen (vom Hinweis bis hin zu einem Lösungsvorschlag).

Anforderungen an die Studentenmodellierung

Um die im vorigen Abschnitt genannten Prinzipien in einer Betreuungskomponente realisieren zu können, muß das Studentenmodul in der Modellierung des Tutanden berücksichtigen, daß Informationen gesammelt werden, die nach geeigneter Bearbeitung Rückschlüsse auf folgende Eigenschaften (Attribute) des Tutanden zulassen:

- **kognitive Komplexität**: In intelligenten Lehrsystemen läßt sich auf die kognitive Komplexität des Tutanden durch Betrachtung des Problemlösevorgehens schließen. Ist der Tutand in der Lage, die Problemstellungen des

Sachgebiets korrekt zu behandeln, so kann von einer kognitiven Struktur hoher Komplexität ausgegangen werden. Andererseits ist bei fehlerhafter Bearbeitung ein Mangel in der kognitiven Struktur (seien es Wissenselemente oder Verbindungen dazwischen) anzunehmen. Somit sollte ein Studentenmodell festhalten, wie gut im bisherigen Verlauf der individuellen Lerngeschichte die Leistungen des Tutanden waren. Dazu ist eine Bewertungsfunktion nötig, die einen Hinweis auf die Bearbeitungsgüte in den vergangenen Sitzungen gibt. Die Resultate sollten dabei nach ihrem Alter gewichtet werden; neuere Ergebnisse gehen stärker ein als die zu Beginn des Lehrganges, um einen aktuellen Zustand der kognitiven Struktur zu erhalten, der aber dennoch die Lerngeschichte nicht außer acht läßt. Die konkrete Berechnung hängt stark von der Strukturierung des Sachgebietes ab, prinzipiell kann jedoch gesagt werden, daß Wissenselemente, die größere Zusammenhänge repräsentieren, sehr viel stärker in die kognitive Komplexität eingehen sollten als Elemente mit eng gefaßtem Inhalt.

- **Motivdisposition**: um dieses Merkmal eines Tutanden zu bestimmen, muß im Studentenmodell vermerkt werden, wie sich der Schwierigkeitsgrad der gewählten Aufgaben zur kognitiven Komplexität des Tutanden verhält. Dabei ist natürlich Grundvoraussetzung, daß die Studentin ihre Aufgaben selbst auswählen kann. Wählt sie Aufgaben, die ihrem Wissensstand angemessen sind, so kann davon ausgegangen werden, daß es sich um eine Erfolgsmotivierte handelt, sind große Unterschiede zwischen der Aufgabenschwierigkeit und der kognitiven Komplexität festzustellen, so ist sie wohl mißerfolgsmotiviert. Da sich dieses Merkmal zu Beginn nur auf sehr wenige Daten stützt, scheint es ratsam, den Tutanden nach einer Eigeneinschätzung zu fragen, ob er einen eher geführten oder eher selbständigen Arbeitsstil vorziehen würde. Wiederum sind die Resultate jüngeren Ursprungs stärker zu beachten als die älteren Beobachtungen.

- **Beherrschung von einzelnen Wissenselementen**: um spezielle Schwachpunkte in der kognitiven Struktur des Tutanden ausfindig machen zu können, und in geeigneter Weise darauf eingehen zu können, sollte auch die Bearbeitungsgüte der einzelnen Wissenselemente im Studentenmodell festgehalten werden. Auch hier ist eine Gewichtung über die Zeit anzuraten; neuere Ergebnisse sind wichtiger als ältere. Die Beherrschung der Wissenseinheit sollte in die Auswahl des Interventionszeitpunktes einfließen, d,h. falls Probleme damit bestehen, sollte früher und evtl auch genauer interveniert werden.

Zusätzlich zu den Informationen, die langfristig abgelegt werden müssen, um geeignet aufbereitet werden zu können, sind für die Didaktikkomponente auch andere Aspekte während des Verlaufs einer Sitzung von Bedeutung:

- **Interventionszeitraum**: Wann war die letzte Intervention? Interventionen, die dem Tutanden eine Rückmeldung zu seinen Arbeitsfortschritten geben, sind nötig, selbst wenn die Arbeit ohne Probleme vor sich geht; insbesondere bei Mißerfolgsmotivierten sollte der Zeitraum ohne Rückkopplung nicht zu lange werden.

- **Schwierigkeiten mit einem bestimmten Wissenselement**: Wie lange hat der Tutand Schwierigkeiten mit einem speziellen Wissenselement?

Ist die Zeitspanne schon größer, so ist eine Hilfestellung zu geben, da der Student entweder nicht mehr weiter weiß oder nicht bemerkt hat, daß ein Fehler aufgetreten ist.

- **Typ der letzten Intervention**: Welcher Art war die letzte Intervention zu diesem Wissenselement? Diese Information dient dazu, die Genauigkeit eines Hinweises festzulegen. Haben Hilfestellungen bisher keinen Erfolg gezeigt, dann ist der nächste Hinweis genauer zu formulieren.

Umsetzung der Lehrstrategie in SYPROS

SYPROS ist ein intelligentes Lehrsystem, das als Wissensgebiet die parallele Programmierung mittels Semaphoren umfaßt. Es stellt eine Menge von Hilfs- und Korrekturmechanismen, wie z.B. Hinweise, Erklärungen, Lösungsvorschläge, Konzepterklärungen, zur Verfügung, so daß eine Abfolge von Hilfsmaßnahmen mit ansteigender Genauigkeit möglich ist. Die Wissensbasis des Expertenmoduls besteht aus der Repräsentation von Zielen und Plänen, welche in Synchronisationsaufgaben auftreten können. Ziele sind Wissenselemente, die beschreiben, wie eine Aufgabe gelöst werden kann, Pläne dienen zur Umwandlung der Ziele in konkreten Programmcode. SYPROS ist in der Lage, aus jeder formalen Aufgabenbeschreibung (siehe dazu [RH93]) einen sog. Ziel-Plan-Baum zu erstellen. Dieser Baum beinhaltet alle Bausteine, die für die jeweilige Aufgabe von Bedeutung sind und wird zur Diagnose von Studentenaktionen herangezogen (siehe Abbildung 1).

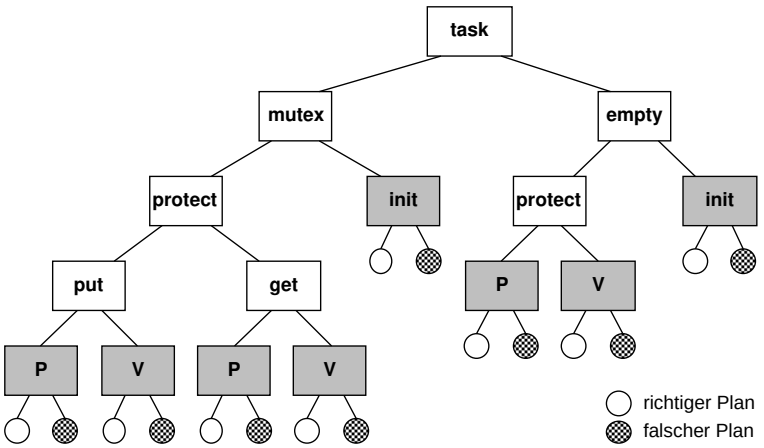


Abbildung 1: Beispiel für einen Ziel-Plan-Baum

Nach jeder Aktion des Studenten versucht das Expertenmodul die getätigte Operation dem am besten dazu passenden Plan zuzuordnen. Dies wird von einem auf unscharfer Logik basierenden Interpreter durchgeführt. Für jedes einzelne Ziel errechnet die Diagnosekomponente die dafür zutreffenden Diagnosekategorien (diese spiegeln wider, ob das Ziel korrekt oder inkorrekt erfüllt ist und um welche Fehlerart es sich im inkorrekten Fall handelt) [HG93].
Im Studentenmodell werden die Diagnosekategorien aller Ziele in den bisher bearbeiteten Aufgaben gespeichert, um dem Tutormodul die von ihm benötigten Informationen zur Verfügung zu stellen [GHK95]. Zur Berechnung der kognitiven Komplexität des Tutanden – einem wichtigen Einflußfaktor für die Auswahl von Tutorregeln – ist es notwendig, zu wissen, wie gut der Student die einzelnen Wissenselemente des Sachgebietes – in diesem Fall die Ziele der Wissensbasis

von SYPROS – beherrscht. Dazu ist das Vorhandensein einer Bewertungsfunktion für die Zielinkarnationen erforderlich; für die Zielinkarnationen und nicht die Ziele selbst, da ein Ziel in einer Aufgabe mehrmals auftreten kann und der Tutand eventuell die diversen Inkarnationen mit unterschiedlichem Erfolg bearbeitet, weshalb das Getrennthalten der Beurteilungswerte der Inkarnationen Aufschlüsse über den Aufbau der kognitiven Struktur des Tutanden liefern kann. Eine zusammenfassende Behandlung aller Zielinkarnationen könnte demnach zu einem Informationsverlust führen. Für jede Aktion wird für die betroffenen Ziele eine neue Zielbewertung errechnet und in eine Bewertungsliste eingetragen. Ausgehend vom Ziel-Plan-Baum berücksichtigt die Bewertungsfunktion

- daß die Komplexität *Komp(ziel)* der einzelnen Ziele sich stark unterscheiden kann;
- daß Unterbäume, die für einen Großteil der Ziele Bewertungen besitzen, mehr Aussagekraft haben als solche Unterbäume, die nur wenige bewertete Ziele haben (zur Lösung jenes Ziels wurde dann noch nicht viel unternommen). Der Grad der Bewertung der Unterbäume eines Ziels sei im folgenden als *Def(ziel)* bezeichnet;
- daß Diagnosen von SYPROS selbst einen Verläßlichkeitsgrad *Rel* erhalten, d.h. ist die Diagnosekomponente sich ihrer Diagnose nicht sehr sicher, so sollte diese Bewertung geringeres Gewicht erhalten;
- die Bewertung der Unterziele des zu bewertenden Zieles; dieser Wert ist natürlich der ausschlaggebende für die Bewertung.

und hat als Ergebnis , genannt *Beurteilung* den Wert

$$Beurteilung(ziel) = \frac{\sum_{uz \in Unterziele} Rel * Komp(uz) * Def(uz) * Beurteilung(uz)}{\sum_{uz \in Unterziele} Rel * Komp(uz) * Def(uz)} \in [-1, +1]$$

(Divisionen in den Formeln erfolgen aus Normierungsgründen.)

Die Bewertungen einer Zielinkarnation *zi* während der Bearbeitung einer Aufgabe werden in einem Wert, der *Bearbeitungsgüte*, zusammengefaßt. Bewertungen mit wenig bewerteten Unterbäumen gehen mit geringerem Gewicht ein als jene mit vollständiger bewerteten Unterbäumen. Somit ergibt sich als Formel für die Berechnung der *Bearbeitungsgüte* :

$$Bearbeitungsgüte(zi) = \frac{\sum_{b \in Bewertungsliste} Beurteilung(b) * Def(b)}{\sum_{b \in Bewertungsliste} Def(b)} \in [-1, +1]$$

Die Bewertungen der Zielinkarnationen in Form von *Bearbeitungsgüte(zi)* werden sodann zusammengefaßt, um einen Wert für die Beherrschung der Wissenseinheit im Laufe des gesamten Kurses, d.h. in allen Aufgaben sämtlicher Sitzungen, zu erhalten. Ziele können in verschiedener Umgebung im Ziel-Plan-Baum auftreten, d.h. Aufgaben, in denen ein Ziel in einem komplexen Zusammenhang steht, sollten einen gewichtigeren Anhaltspunkt für die Beherrschung eines Zieles liefern, als welche, die das Ziel in einfachem Zusammenhang beinhalten. Die Anzahl der

Blätter *anzBlatt(zi)* im Unterbaum der Zielinkarnation gibt an, wieviele Aktionen benötigt werden, um das Ziel zu erfüllen und damit auch, wie komplex der Zusammenhang ist, in dem das Ziel steht. Außerdem tragen kürzlich bearbeitete Aufgaben mehr zum momentan zutreffenden Wissensstand des Studenten bei als weiter zurückliegende. Deshalb wurde ein Gewichtsfaktor *Gewicht(zi)* eingeführt, der für neuere Daten höher ist als für alte Informationen. Die Formel für die *Zielbeherrschung(ziel)* ergibt sich schließlich zu:

$$Zielbeherrschung(ziel) = \frac{\sum_{zi} Gewicht(zi) * Bearbeitungsgüte(zi) * anzBlatt(zi)}{\sum_{zi} Gewicht(zi) * anzBlatt(zi)} \in [-1, +1]$$

Aus den Werten für die Beherrschung der einzelnen Ziele, kann eine Gesamtgröße für die Wissensdomäne errechnet werde, die die **kognitive Komplexität** für das Wissensgebiet annähert (für genauere Ausführungen über die diversen Berechnungen siehe [Har96]). Für SYPROS wird folgende Formel verwendet:

$$kognitiveKomplexität = \frac{\sum_{ziel \in MengeallerZiele} Zielbeherrschung(ziel) * komp(ziel)}{\sum_{ziel \in MengeallerZiele} komp(ziel)} \in [-1, +1]$$

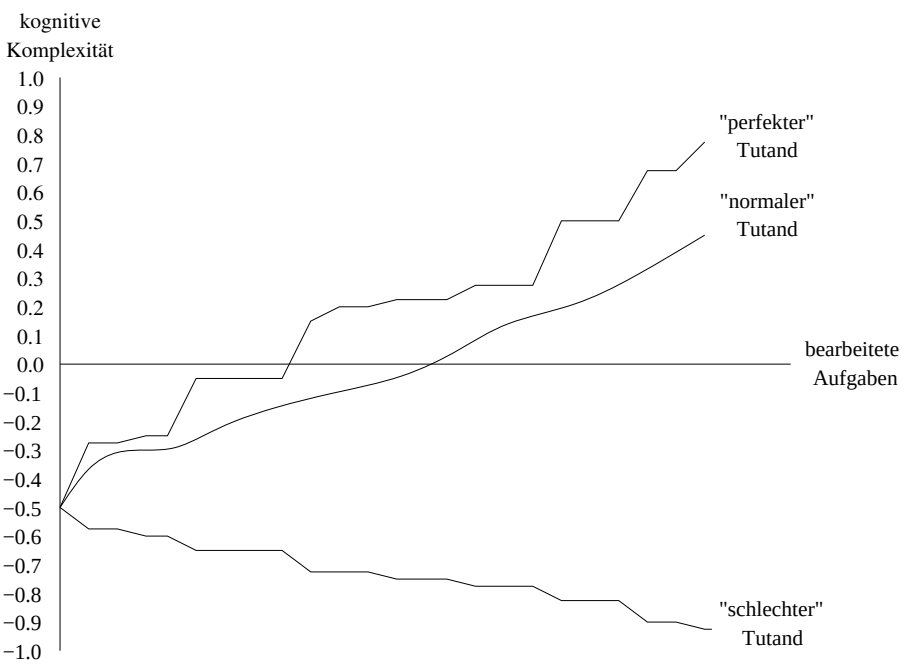


Abbildung 2: Entwicklung der kognitiven Komplexität im Verlaufe der Sitzungen mit SYPROS

In Abbildung 2 kann man verfolgen, wie sich der Wert für die kognitive Komplexität für einen Studenten , der niemals Fehler macht („perfekter Tutand"), für einen Studenten, der nur Fehler macht („schlechter Tutand") und für einen Studenten, dessen Leistung mit zunehmender Übung besser wird („normaler Tutand"), verhält. Die Plateaus in den Kurven für den „perfekten" und „schlechten" Studenten entstehen dadurch, daß Aufgaben bearbeitet wurden, die nur Ziele beinhalten, die vor der Aufgabe schon diesen Wert der *Zielbeherrschung* erreicht hatten. Der „perfekte" Student wird die Komplexität +1.0 erreichen und der „schlechte" Student den Wert -1.0 , wenn alle Ziele der Wissensbasis in Aufgaben mindestens einmal behandelt wurden (der Ausgangswert von -0.5 und nicht

etwa -1.0 wurde festgelegt, um zu verhindern, daß bei fehlender Information über den Studenten, dieser sehr schlecht eingestuft wird, obwohl dies möglicherweise nicht seinem Vermögen entspricht. Für den „schlechten“ Studenten ergibt sich also nicht wirklich eine Verschlechterung durch Lernen, sondern eine Revision der ursprünglichen - zu optimistischen - Annahme). Für den „normalen“ Studenten steigt die Zielberrschung mit zunehmender Übung an und damit wird auch innerhalb desselben Aufgabentyps (sie enthalten keine neuen Ziele) ein Anstieg zu erwarten sein.

Um die **Motivdisposition** zu erhalten, wurde eine weitere Funktion entwickelt, die aus einem Ziel-Plan-Baum einen Wert für die Aufgabenschwierigkeit errechnet. In diese geht sowohl die isoliert betrachtete Schwierigkeit der einzelnen Ziele, als auch die Position und die Zusammenhänge im Baum ein. Einen weiteren Schritt zur Adaption an die individuelle Studentin könnte in diesem Zusammenhang die Verwendung einer subjektiven Schwierigkeit für die jeweilige Studentin sein. Ziele, die der Studentin besonders schwer fallen, würden den subjektiven Schwierigkeitsgrad, der die Entscheidung, eine Aufgabe zu bearbeiten, wesentlich beeinflußt, erhöhen. Dies stellt eine Möglichkeit für weitere Arbeiten dar, wurde bisher in der Berechnung der Aufgabenschwierigkeit nicht berücksichtigt. Mit den Werten für die kognitive Komplexität und für die Aufgabenschwierigkeit, kann schließlich die Motivdisposition gefolgert werden. Ist die Differenz zwischen beiden Werten groß (die Aufgabe ist entweder zu schwer oder zu leicht), so verhält sich die Studentin mißerfolgsmotiviert, ist die Differenz klein, so wurde eine Aufgabe gewählt, die dem Wissensstand entspricht, was auf eine erfolgsmotivierte Person hindeutet. Somit beeinflußt eine große Differenz zwischen Aufgabenschwierigkeit und kognitiver Komplexität den Wert für die Motivdisposition in Richtung mißerfolgsmotiviert, eine kleine Differenz in die Gegenrichtung erfolgsmotiviert.

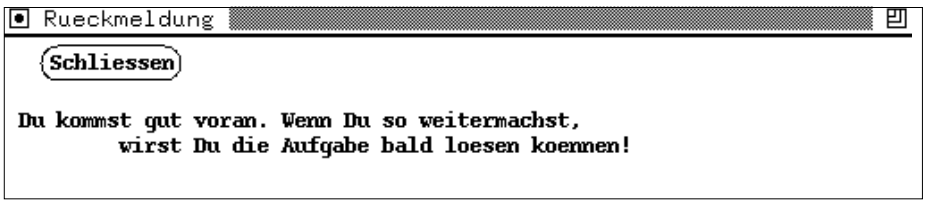
Nachdem die für die Lehrstrategie wichtigen Größen berechnet sind, können die entwickelten Prinzipien in einer Implementierung der Strategie realisiert werden. Diese Umsetzung verwendet die Attribute dazu, die vielfältigen Hilfs- und Korrekturmöglichkeiten von SYPROS in einer dem Studenten angepaßten Abfolge darzubieten. Die Tutorstrategie wurde schlußendlich als Regelkatalog, bestehend aus 66 WENN-DANN-Regeln – wie die folgende – implementiert:

Regel in der Regeldatei:

```
/*-------------------------------------------------------------------*/
Regelname         : affektiveRueckmeldung1
/*-------------------------------------------------------------------*/

Regelnummer       : 161

Voraussetzungen   : $keineSchwierigkeiten() &&
                    (($istMisserfolgsmotiviert() &&
                      $hatNiedrigeKomplexitaet())
                     ||
                     ($istMisserfolgsmotiviert() &&
                      $hatMittlereKomplexitaet()))
                    && $kurzerInterventionszeitraum()

Aktionen          : &Ausfuehrung: $Rueckmeldung(LOB);
                    &Ausfuehrung: $resetIntervention();
```

Diese Regel sagt aus, daß falls der Student momentan keine Probleme hat, mißerfolgsmotiviert ist und eine niedrige bis mittlere kognitive Komplexität besitzt und schon einige Aktionen seit der letzten Intervention getätigt wurden, die Tutorkomponente eine Intervention vom Typ `Lob` initiieren und den Zähler für die letzte Intervention auf Null zurücksetzen soll. In der jetzigen Implementierung liegen bestimmte Ausprägungswerte für die diversen Attribute vor (z.B. hohe, mittlere oder niedrige kognitive Komplexität), die Abstufungsfeinheit im Intervall [-1,+1] wird also nicht vollständig ausgenutzt (im Ausblick wird die Überlegung zur besseren Ausnutzung der zur Verfügung stehenden Genauigkeit behandelt).

Wird obige Regel angewendet, so gibt SYPROS folgendes Fenster aus:



Zusammenfassung und Ausblick

In diesem Artikel wurde eine Lehrstrategie vorgestellt, die auf kognitiven und motivationalen Gesichtspunkten beruht. Nach der Darstellung der zugrundeliegenden Prinzipien, wurden Anforderungen an die Studentenmodellierung genannt. Sind diese Anforderungen erfüllt, so kann diese Strategie umgesetzt werden, was am Beispiel des intelligenten Lehrsystems SYPROS gezeigt wird.

Durch einen Feldversuch soll im weiteren Verlauf eine Evaluierung dieser Lehrstrategie vorgenommen werden. Außerdem scheint eine Überführung der WENN-DANN-Regeln in unscharfe Regeln sinnvoll und für SYPROS gut geeignet zu sein, da somit eine feinere Abstimmung auf die - prinzipiell zur Verfügung stehende, aber noch nicht vollständig ausgenutzte - Genauigkeit der Attributsberechnungen vorgenommen werden könnte. Falls ein Lehrsystem über die Möglichkeit verfügt, Gruppenlernen [Hop95] zu unterstützen, wäre es sicherlich denkbar, nicht nur den Wissensstand als Kriterium zur Auswahl eines Partners für einen Studenten heranzuziehen, sondern auch deren motivationale Eigenschaften. Zwei mißerfolgsmotivierte Studenten, die zusammenarbeiten, können sich gegenseitig Rückmeldung und Unterstützung geben und das sicherlich in einer „echteren“ Weise als es die Rückmeldungen eines Computertutors oder eines computergenerierten „Mitschülers“ [CB88] jemals könnten.

Literatur

[And93] John R. Anderson. *Rules of the mind.* Lawrence Erlbaum Assiciates, Hillsdale, New Jersey, 1993.

[CB88] Tak-Wai Chan and Arthur B. Baskin. “studying with the prince”. In C. Frasson, G. Gauthier, and G.I. McCalla, editors, *Intelligent Tutoring Systems*, Montreal, 1988.

[Cha96] Tak-Wai Chan. Learning companion systems, social learning systems, and the global social learning club. *Journal of Artificial Intelligence in Education*, 7(2):125–159, 1996.

[Ein81] Wolfgang Einsiedler. *Lehrmethoden.* Urban & Schwarzenberg, München, Wien, Baltimore, 1981.

[GHK95] Meike Gonschorek, Christian Herzog, and Eva Kluge. Eine Didaktikkomponente für SYPROS — Studentenmodellierung, Lernzielstrukturierung und Lernerführung. In F. Huber-Wäschle, H. Schauer, and P. Widmayer, editors, *Herausforderungen eines globalen Informationsverbundes für die Informatik: 25. GI-Jahrestagung und 13. Schweizer Informatikertag, GISI 95, Zürich*, Informatik aktuell, pages 293–303. Springer, September 1995.

[Har96] Andreas Harrer. Ein didaktisches Konzept für die Lernerführung in einem intelligenten Lehrsytem. Diplomarbeit, TU München, Institut für Informatik, Februar 1996.

[Hec89] Heinz Heckhausen. *Motivation und Handeln.* Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, 1989.

[HG93] Christian Herzog and Meike Gonschorek. Generation of adaptive explanations in ITS with hierarchical knowledge domains. In Tak-Wai Chan, editor, *Proc. 1993 International Conference on Computers in Education, Taiwan*, pages 1–6, December 1993.

[Hop95] H. Ulrich Hoppe. The use of multiple student modelling to parametrize group learning. In J. Greer, editor, *Proc. of the World Conference on Artificial Intelligence in Education AI–ED 95*, Washington, 1995.

[LC88] Mark R. Lepper and Ruth W. Chabay. Socializing an intelligent tutor: Bringing empathy to computer tutors. In Heinz Mandl and Alan Lesgold, editors, *Learning Issues for Intelligent Tutoring Systems.* Springer, New York, Berlin, Heidelberg, London, Paris, Tokyo, 1988.

[RH93] Daniela Rothenhöfer and Christian Herzog. SYPROS — an intelligent tutoring system for parallel programming. In Tak-Wan Chan, editor, *Proc. 1993 International Conference on Computers in Education, Taiwan*, pages 300–305, December 1993.

[Sch74] Hans Schiefele. *Lernmotivation und Motivlernen.* Ehrenwirth, 1974.

[TH71] P.D. Tomlinson and D.E. Hunt. Differential effects of rule-example order as a function of learner conceptual level. *Canadian Journal of Behavioral Science 3*, 1971.

[VW96] Julita Vassileva and Barbara Wasson. Instructional planning approaches: from tutoring towards free learning. In *Proceedings of EuroAI-ED'96 Lisbon,Portugal*, pages 1–8, 1996.

Wissensbasierte Integration der Komponenten eines intelligenten Lehrsystems

Meike Gonschorek

Fakultät für Informatik der TU München,
D–80290 München
gonschor@informatik.tu-muenchen.de,
http://www.informatik.tu-muenchen.de/~gonschor/

1 Einleitung

Um den heutigen weltweit vernetzten Systemen gerecht zu werden, hat sich der Aufbau moderner Systeme stark verändert: immer öfter werden agentenbasierte Architekturen vorgeschlagen, die den Ressourcenaufwand verringern, die Effizienz steigern und komplexe Aufgaben besser modellieren sollen (siehe [Müll96]). Auch auf dem Gebiet der „intelligenten Lehrsysteme“ (ILS) werden diese Techniken vermehrt eingesetzt (siehe [FGL96]). Einerseits versucht man durch eine agentenbasierte Architektur in einem System verschiedene Lern– bzw. Sozialformen wie Einzel–, Partner– oder Gruppenarbeit zu verwirklichen. Andererseits werden sie dazu verwendet, verschiedene Lehrstrategien durch jeweils einen Agenten zu realisieren (siehe [FMAG96]). Schließlich möchte man den trotz eigentlich großen Bedarfs an Lernsoftware mangelnden Einsatz intelligenter Lehrsysteme durch die Modellierung mit Agenten verbessern, indem man die einzelnen Komponenten verschiedener Systeme vereinigt, um ein komplettes, in der Praxis einsetzbares System zu erhalten.

[Brus95] beschreibt dafür eine Lösungsmöglichkeit, die **Studentenmodul–zentrierter Ansatz** genannt wird. Dort wird eine integrationsorientierte Architektur mit einem zentralen Studentenmodul vorgeschlagen, das die Informationen über den Studenten von allen Komponenten sammelt und integriert. Es wird sowohl von Agenten, das sind wissensbasierte Module, als auch von Werkzeugen, das sind Schnittstellen–Module, verwendet, um sich an den Studenten anzupassen. Zur Interaktion stehen standardisierte Protokolle (Projektionen) zur Verfügung, die mit Hilfe von Projektoren erzeugt werden.

In dieser Arbeit wird ein anderer Ansatz vorgestellt, wie verschiedene Agenten, in unserem Fall sind dies Komponenten für ein ILS, in ein bestehendes Lehrsystem konsistent integriert werden können. Dazu wird erst die Architektur solch eines „integrierten Lehrsystems“ vorgestellt, in der außer den üblichen Komponenten eine Repräsentation von sogenanntem Relationswissen existiert, anschließend wird die Modellierung dieses Wissens beschrieben und Steuerungsmechanismen für die integrierten Hilfestellungen vorgeschlagen. Am Schluß wird beispielhaft die Integration der Hilfen im intelligenten Lehrsystem SYPROS durchgeführt.

2 Architektur

Auch dieser Ansatz möchte eine Integration der Komponenten eines ILS erreichen. Genau wie im Studentenmodul-zentrierten Ansatz wird auch hier die Standardarchitektur verändert (siehe

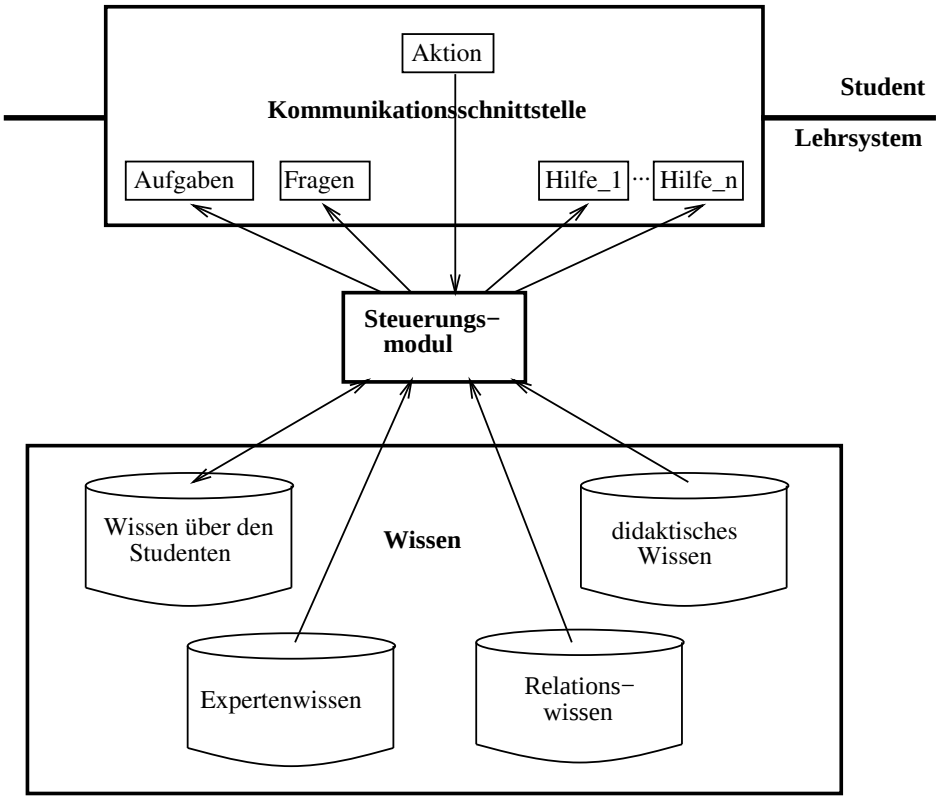


Abbildung 1: Die Architektur des wissensbasierten Ansatzes

Abbildung 1). Eine bessere Koordination der einzelnen Komponenten eines Systems wird dadurch erreicht, daß zusätzlich zu den bereits existierenden wissensbasierten Komponenten, die **Wissen über den Studenten**, **didaktisches Wissen** und **Expertenwissen** repräsentieren, die Eigenschaften der Systemkomponenten und damit auch die Zusammenhänge im System in einer Wissensbasis explizit repräsentiert werden. Dieses Wissen wird **Relationswissen** genannt (siehe auch Abschnitt 6). Dadurch ist es möglich, jede Hilfe im Dialog konsistent einzusetzen. Die Auswahl erfolgt lediglich über die im Relationswissen festgelegten Eigenschaften des jeweiligen Werkzeugs. Dieses entspricht der Vorgehensweise eines Lehrers, denn er weiß über die Vor– und Nachteile der verschiedenen Lehrmittel (z. B. kurzer Hinweis, Fehlererklärung oder Lösung) Bescheid und kann sie anhand dieser Eigenschaften bei der gewählten Unterrichtsmethode richtig einsetzen.

Dadurch kann dann auch das **didaktische Wissen** allgemeiner formuliert werden. In den Aktionen der Tutorregeln werden die einzelnen Präsentationen nicht mehr direkt benannt, sondern lediglich die Eigenschaften der gewünschten Hilfe werden festgelegt.

Ein **Steuerungsmodul** erhält die **Aktionen** des Studenten und entscheidet daraufhin anhand des Wissens aus den verschiedenen Wissensmodulen, welche konkrete **Aufgabe**, **Frage** bzw. **Hilfe** auf die Beschreibung in den Tutorregeln paßt und präsentiert diese dem Studenten.

3 Modellierung des Relationswissens

Um die Auswahl der Hilfen nur noch über ihre Eigenschaften steuern zu können, muß das Relationswissen mehrschichtig dargestellt werden. Eine obere Schicht, auf die das Steuerungsmodul zugreift, um die gewünschte Hilfe zu bestimmen, beschreibt auf einer allgemeinen Ebene die Zusammenhänge, die in jedem ILS vorhanden sein müssen. Auf der unteren Ebene werden dann dieser allgemeinen Beschreibung einer Unterweisung konkrete Realisierungen in einem bestimmten Lehrsystem zugeordnet.

3.1 Modellierung allgemeiner Zusammenhänge in einem ILS

Die allgemeinen Relationen in einem ILS werden auf einer Abstraktionsebene modelliert, auf der auch didaktische Entscheidungen getroffen werden, damit das didaktische Wissen möglichst allgemein formuliert werden kann. Wie auch die Unterrichtsplanung eines Lehrers nicht bestimmte Hilfsmittel, die ihm konkret zur Verfügung stehen, beschreibt (z. B. Lehrervortrag, Schulfernsehen oder Lernprogramm), sondern diese über ihre Eigenschaften bezeichnet (z. B. Präsentationen, bei denen etwas erklärt wird), soll auch hier von den konkreten Gegebenheiten in einem bestimmten Lehrsystem abstrahiert werden. Zur Formalisierung eines allgemeinen Lehrsystems wird auf eine Instruktionsentwurfstheorie, die „**Component Display**"–Theorie (kurz CDT) von M. D. Merrill [Merr94] aufgebaut, die einen Formalismus für eine allgemeine Unterweisung darstellt.

Merrill teilt in dieser den **Lehrstoff** in

- Fakten, das sind Zuordnungen eines Symbols zu einem Begriff,
- Konzepte, das sind Gruppen von Handlungen oder Symbolen mit gemeinsamen Eigenschaften,
- Prozeduren, das sind Schritte, um ein Ziel zu erreichen, und
- Prinzipien, das sind Erklärungen, Vorhersagen, warum etwas auf eine bestimmte Weise abläuft,

ein.

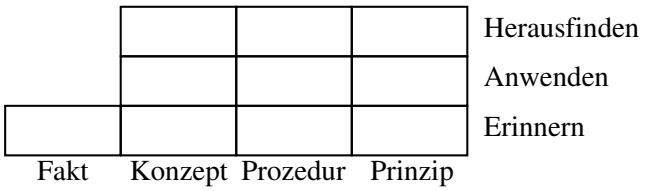


Abbildung 2: Formalisierung der Lehrzielarten

Die **Aktionen** bzw. Tätigkeiten der Lerner teilt er in die Gruppen Erinnern, Anwenden und Herausfinden ein. So entsteht eine Kategorisierung der **Lehrziele** in z. B. „Erinnerung an ein Konzept", „Anwendung eines Konzepts" oder „Herausfinden eines Konzepts" (siehe Abbildung 2). Eine Ausnahme bilden die Fakten, da es keine Lehrziele gibt, in denen das Anwenden oder Herausfinden eines Fakts verlangt wird. Nur die „Erinnerung an ein Faktum" kann zum Lernziel werden.

Zusätzlich zu den Lehrzielen formalisiert Merrill außerdem die **Präsentationsformen**, die verwendet werden können, um ein Lehrziel zu vermitteln. Diese teilt er in primäre und sekundäre Präsentationsformen ein.

	Darlegen	Erfragen
Allgemeinheit	**DA**	**FA**
Instanz	**DI**	**FI**

Abbildung 3: Primäre Präsentationsformen

Als **primäre Präsentationsformen** bezeichnet er das Darlegen einer Allgemeinheit (**DA**), die Darlegung einer Instanz (**DI**), das Erfragen einer Allgemeinheit (**FA**) und das Erfragen einer Instanz (**FI**) (siehe Abbildung 3). Unter Allgemeinheiten werden allgemeine Beschreibungen des Lehrinhalts wie z. B. Definitionen, allgemeine Beschreibungen eines Ablaufs oder Lehrsätze verstanden. Instanzen dagegen sind sind konkrete Instantiierungen eines Lehrinhalts wie Beispiele für ein Konzept, konkrete Abläufe einer Prozedur oder Situationen, in denen ein Prinzip beobachtet werden kann.

Sekundäre Präsentationsformen werden verwendet, um die Informationsaufnahme des Studenten zu erleichtern bzw. die Motivation zu steigern. Sie sind eine erweiterte Beschreibung

(sog. „Elaborationen“) der primären Präsentationsformen. Merrill unterscheidet acht verschiedene Arten der Elaboration, wobei er drei für die Beschreibung von Reaktionen des Lehrers bzw. des Lehrsystems auf Antworten des Lernenden bereithält. Intelligente Lehrsysteme, besonders jene, die das Programmieren lehren, verwenden jedoch hauptsächlich Rückmeldungen auf Studentenaktionen bei der Lösung von Aufgaben, um den Lernprozeß voranzutreiben. Hier stellen sie die verschiedensten Arten wie kurze Hinweise zur Richtigkeit bis hin zur vollständigen Überarbeitung des Problems bereit. Daher mußten die sekundären Präsentationsformen im Sinne von Merrill stärker differenziert werden (siehe Abbildung 4). Außerdem wurden auch noch rein affektive Rückmeldungen miteinbezogen, die keine Informationen liefern, sondern nur die Motivation steigern sollen. Damit die Übersichtlichkeit bei nun 15 Arten von Elaborationen nicht verloren geht, wurden sie darüberhinaus noch etwas anders strukturiert.

				DA	DI	FA	FI
Ergänzung (eg)	Information (i)	Kontext	(kt)	**DA.eg.i.kt**	**DI.eg.i.kt**	**FA.eg.i.kt**	**FI.eg.i.kt**
		Vorbedingung	(v)	**DA.eg.i.v**			
		Gedächtnishilfe	(g)	**DA.eg.i.g**	**DI.eg.i.g**		
		Hervorhebung	(h)	**DA.eg.i.h**	**DI.eg.i.h**	**FA.eg.i.h**	**FI.eg.i.h**
		Repräsentation	(r)	**DA.eg.i.r**	**DI.eg.i.r**	**FA.eg.i.r**	**FI.eg.i.r**
	Motivation (m)	positive Motivation	(p)	**DA.eg.m.p**	**DI.eg.m.p**	**FA.eg.m.p**	**FI.eg.m.p**
		negative Motivation	(n)	**DA.eg.m.n**	**DI.eg.m.n**	**FA.eg.m.n**	**FI.eg.m.n**
Rückmeldung (rm)	Information (i)	Diagnose	(d)			**FA.rm.i.d**	**FI.rm.i.d**
		Lokalisation	(lk)			**FA.rm.i.lk**	**FI.rm.i.lk**
		Beschreibung	(b)			**FA.rm.i.b**	**FI.rm.i.b**
		Teillösung	(tl)			**FA.rm.i.tl**	**FI.rm.i.tl**
		Darlegen der Teillösung	(dtl)			**FA.rm.i.dtl**	**FI.rm.i.dtl**
		Lösung	(ls)			**FA.rm.i.ls**	**FI.rm.i.ls**
		Darlegen der Lösung	(dls)			**FA.rm.i.dls**	**FI.rm.i.dls**
	Motivation (m)	Lob	(l)			**FA.rm.m.l**	**FI.rm.m.l**
		Kritik	(kr)			**FA.rm.m.kr**	**FI.rm.m.kr**

Abbildung 4: Sekundäre Präsentationsformen

Bei den darlegenden primären Präsentationsformen (DA, DI) können Ergänzungen (*.eg.*) als Elaborationen angegeben werden, die sich in informative (*.eg.i.*) und motivationale Elaborationen (*.eg.m.*) gliedern: Zum einen können kontextuelle bzw. historische Hintergründe (**DA.eg.i.kt**, **DI.eg.i.kt**) des Dargelegten hinzugefügt werden (z. B. Wer entdeckte das Prinzip, wie entstand es?). Dann sollten bei der Darlegung einer Allgemeinheit die Vorbedingungen bzw. Voraussetzungen geschildert werden, damit man den primären Inhalt verstehen kann (**DA.eg.i.v**). Manchmal ist es sinnvoll, dem Studenten eine Gedächtnishilfe oder eine Eselsbrücke zu präsentieren, die es ihm erleichtert, sich an den Lehrinhalt zu erinnern (**DA.eg.i.g**, **DI.eg.i.g**). Diese drei Arten von Elaborationen sind wieder selbst primäre Darlegungen einer Allgemeinheit bzw. Instanz und können daher wieder selbst durch sekundäre Präsentationsformen elaboriert werden.
Um Wichtiges leichter erkennen zu können, werden auch oft Hervorhebungen wie Pfeile oder Fettdruck verwendet (**DA.eg.i.h**, **DI.eg.i.h**). Eine andere Darstellung bzw. Repräsentation wie Diagramme, Formeln oder Bilder kann auch zu einem besseren Verständnis führen (**DA.eg.i.r**, **DI.eg.i.r**).
Schließlich sollte man nicht vergessen, die Motivation des Studenten aufrecht zu erhalten (*.eg.m.*). Hierzu unterscheidet man positive motivationale Kommentare (**DA.eg.m.p**,

DI.eg.m.p) wie z. B. „Das folgende wirst Du sicher schnell verstehen“ und negative Bemerkungen (**DA.eg.m.n**, **DI.eg.m.n**) wie z. B. “Jetzt mußt Du Dich einmal besonders anstrengen“.

Die Elaborationen der erfragenden Präsentationsformen (FA, FI) gliedern sich in zwei Gruppen: Zum einen können Ergänzungen der Fragestellung gegeben werden (*.eg.*), falls dies zum besseren Verständnis der Frage führt oder falls der Student die Frage bzw. die Aufgabe nicht beantworten kann. Hier sind größtenteils die gleichen Elaborationen wie bei den darlegenden Präsentationen möglich.
Falls der Student eine Antwort auf die Frage gegeben hat bzw. falls er begonnen hat, die gestellte Aufgabe zu bearbeiten, können Rückmeldungen gegeben werden (*.rm.*). Diese gliedern sich wiederum in informative (*.rm.i.*) und Rückmeldungen, die die Motivation steigern (*.rm.m.*). Eine informative Rückmeldung ist die Diagnose der Antwort, d. h. es wird nur angegeben, ob sie richtig oder falsch ist (**FA.rm.i.d**, **FI.rm.i.d**). Falls der Student einen Fehler gemacht hat, kann dieser lokalisiert (**FA.rm.i.lk**, **FI.rm.i.lk**), d. h. es wird angegeben, wo oder bei welchem Aspekt er sich befindet, oder beschrieben werden (**FA.rm.i.b**, **FI.rm.i.b**). Eine Teillösung (**FA.rm.i.tl**, **FI.rm.i.tl**) oder die ganze Lösung (**FA.rm.i.ls**, **FI.rm.i.ls**) wird präsentiert, wenn der Student die Fehler nicht alleine verbessern kann. Damit der Student die (Teil)lösung auch versteht, kann diese noch zusätzlich erklärt werden (**FA.rm.i.dtl**, **FI.rm.i.dtl**, **FA.rm.i.dls**, **FI.rm.i.dls**). Diese vier Präsentationsformen sind wiederum auch primäre darlegende Präsentationsformen. Das heißt, daß sie selbst wieder elaboriert werden können.
Motivationssteigernde Rückmeldungen sind Lob (**FA.rm.m.l**, **FI.rm.m.l**) und Kritik (**FA.rm.m.kr**, **FI.rm.m.kr**). Sie können jeweils abhängig von den Fähigkeiten und Eigenschaften des Studenten auf verschiedene Arten gegeben werden.

Um bei den sekundären Präsentationsformen, die eine andere Repräsentation verlangen (*.eg.i.r), diese ermitteln zu können, muß noch der Begriff der **Medien** eingeführt werden. Jede Präsentationsform vermittelt ihren Inhalt über ein oder mehrere Medien. Diese gliedern sich angelehnt an [Dale69] in

VB: verbale Vermittlung,
VS: visuelle Vermittlung durch formale Darstellungen, wie Karten, Tabellen, Graphen,
AK: akustische Vermittlung
BD: Bilder,
FL: Filme,
HM: Hypermedia,
SS: Simulationen, die vom System vorgeführt werden oder
SL: Simulationen, die der Lerner selbst bedienen kann.

Mit Hilfe der hier vorgestellten Begriffe kann nun jeder Lehrvorgang durch eine Folge von z. B. primären Präsentationsformen beschrieben werden. So kann z. B. auch jede Lehrmethode durch diese Sprache formalisiert werden.

- Darstellender Unterricht mit Beispielen: $(DA\ DI^*)^*$
  Bei dieser Methode wird jeder Lehrinhalt erst allgemein erklärt und dann durch Beispiele veranschaulicht.

- Lernen durch Selbsterklärung bzw. exemplarisches Lernen: $(DI^*FI)^*$
  Es werden anfangs eine Reihe von Beispielen zu einem Sachverhalt gezeigt, jedoch dieser nicht allgemein erklärt. Anschließend wird durch Fragen bzw. Aufgaben abgeprüft, ob der Lernende das Wichtige aus den Beispielen herausgezogen hat.

- Entdeckendes Lernen bzw. Lernen durch Versuch und Irrtum: $(FI^*FA)^*$
  Hier werden dem Lernenden nur Aufgaben präsentiert, bei denen er aus Fehlern, die er macht, das Konzept, den Vorgang oder ein Prinzip erlernen soll. Anschließend wird er nach der Allgemeinheit gefragt.

3.2 Modellierung konkreter Zusammenhänge

Um nun die Relationen in einem konkreten Lehrsystem zu beschreiben, müssen alle Komponenten eines Lehrsystems den in der „Component Display"–Theorie definierten Begriffen zugeordnet werden.

Das bedeutet im einzelnen: jede Wissenseinheit muß durch einen Lehrinhalt beschrieben werden, alle Lernziele werden einer Lehrzielart zugeordnet und schließlich müssen alle Werkzeuge, die in dem Lehrsystem den Dialog mit dem Lerner führen, einer primären oder sekundären Präsentationsform zugeordnet und die dabei verwendeten Medien angegeben werden. Diese Zuordnung wird in Abschnitt 5 am Beispiel des Lehrsystems SYPROS durchgeführt.

3.3 Realisierung

Das in diesem Abschnitt beschriebene Wissen über die Relationen in einem Lehrsystem ist deklarativ; daher wird es in einem KL–ONE–System repräsentiert ([BrSch85]). In einem solchen System wird das Wissen aufgeteilt in eine **TBox** (Terminology Box), die Wissen über Klassen von Individuen (z. B. konkrete Lehrziele, Präsentationen usw.) und deren Beziehungen untereinander repräsentiert. In der **ABox** (Assertion Box) wird Wissen über bestimmte Individuen dargestellt. Falls man die einzelnen Komponenten eines ILS durch Klassen in der TBox beschreibt, können so die **Relationen** in einem Lehrsystem explizit repräsentiert werden.
Alle in diesem Abschnitt eingeführten Begriffe wie Lehrinhalte, Aktionen, Lehrzielarten, primäre, sekundäre Präsentationsformen, Medien und die einzelnen Komponenten des konkreten Lehrsystems werden in der TBox beschrieben.

4 Modellierung des didaktischen Wissens

Um nun die Komponenten eines intelligenten Lehrsystems über ihre in der TBox beschriebenen Eigenschaften steuern zu können, muß es Regeln geben, die in jeder Situation des Lernprozesses eine passende Präsentationsform aussuchen. Über die gewünschte Präsentation kann dann die dafür zuständige Komponente im ILS referenziert werden.

Merrill macht hier einige Vorschläge, zu welchem Lernziel sich welche primären und sekundären Präsentationsformen eignen und wie man sie einsetzen sollte. Abbildung 5 zeigt zwei dieser Vorschläge, übertragen auf die erweiterten sekundären Präsentationsformen.

WENN	**DANN**		
Lehrzielart	**primäre Präsentations–formen**	**sekundäre Präsentations–formen**	**Beziehungen zwischen Präsentationen**
Anwendung einer Prozedur	FI	FI.eg.i.r FI.rm.i.* FI.rm.m.*	passende, unterschiedliche Instanzen versch. Schwierigkeit nachlassende Genauigkeit
Herausfinden einer Prozedur	FI FA	FA.rm.i.* FA.rm.m.*	passende, unterschiedliche Instanzen versch. Schwierigkeit

Abbildung 5: Didaktische Regeln zur Verwendung der Präsentationsformen

In der ersten didaktischen Regel wird der Vorschlag gemacht, einem Lehrziel, das die Anwendung einer Prozedur zum Inhalt hat, zuerst durch Erfragen einer Instanz (FI) einen bestimmten Anwendungsfall für die Prozedur zu verlangen. Anschließend kann die Frage durch eine

andere Darstellung (FI.eg.i.r) veranschaulicht werden und auf die Antworten durch Rückmeldungen informativer (FI.rm.i.*) und motivationssteigernder Art (FI.rm.m.*) reagiert werden. Die verschiedenen Fragen sollten zueinanderpassen und von unterschiedlicher Art sein, so daß alle wichtigen Aspekte des Lehrziels abgedeckt werden. Die Rückmeldungen sollten mit der Zeit seltener und kürzer werden. Beim Herausfinden einer Prozedur sollten erst eine Menge von Instanzen (FI) ohne Elaborationen erfragt werden. Anschließend sollte nach dem allgemeinen Vorgehen bei der Prozedur gefragt werden (FA) und die Antworten durch informative (FA.rm.i.*) und motivationssteigernde Rückmeldungen (FA.rm.m.*) bereichert werden. Auch hier gelten ähnliche Beziehungen zwischen den Präsentationen wie bei der ersten Regel.

Diese Regeln ziehen allerdings nicht die Fähigkeiten und Eigenschaften des einzelnen Studenten in Betracht. Da die Adaptivität aber eine grundlegende Eigenschaft eines intelligenten Lehrsystems ist, müssen die didaktischen Regeln konsistent zu den Regeln aus Abbildung 5 erweitert werden. Hier kann jede denkbare Lehrstrategie, wie z. B. geführtes oder entdeckendes Lernen, verwendet werden.

Um die generelle Umsetzbarkeit zu zeigen, wurde als Beispiel das didaktische Konzept, das in [Harr97] vorgeschlagen wird, realisiert. Der Student wird dort abhängig von seiner Motivdisposition (erfolgs- oder mißerfolgsmotiviert), seiner kognitiven Fähigkeit und der gegebenen Situation stärker oder weniger eng vom Tutor geführt. Durch diese Erweiterung entstehen dann Regeln, die ausschnittsweise für einen mißerfolgsmotivierten Studenten mit niedriger oder mittlerer kognitiver Fähigkeit und einem Lehrziel vom Typ „Anwenden einer Prozedur“ in Abbildung 6 dargestellt sind.

WENN						**DANN**
No.	**Lehr–ziel–art**	**Motiv–dispo–sition**	**kognitive Fähigkeit**	**Zeitdauer der Schwierigkeiten mit Lehrinhalt**	**Art der Schwierigkeiten mit Lehrinhalt**	**primäre/sekundäre Präsentationsformen**
1	Anwenden einer Prozedur	mißerfolgs–motiviert	*	keine	keine	FI
2			niedrig	keine	keine	FI.eg.i.h + FI.eg.i.r + FI.eg.i.kt FI.eg.m.p
3				sehr kurz	nicht erfüllt	FI.rm.i.tl–>FI.rm.i.dtl–> FI.rm.i.ls–>FI.rm.i.dls
4				sehr kurz	falsch erfüllt	FI.rm.i.b–> FI.rm.i.tl–>FI.rm.i.dtl–> FI.rm.i.ls–>FI.rm.i.dls
5			mittel	keine	keine	FI.eg.i.h + FI.eg.i.r FI.eg.m.p
6				kurz	nicht erfüllt	FI.eg.i.kt–> FI.rm.i.tl–>FI.rm.i.dtl–> FI.rm.i.ls–>FI.rm.i.dls
7				kurz	falsch erfüllt	FI.rm.i.lk–>FI.rm.i.b–> FI.rm.i.tl–>FI.rm.i.dtl–> FI.rm.i.ls–>FI.rm.i.dls

Abbildung 6: Erweiterte didaktische Regeln zur Verwendung der Präsentationsformen

Die erste Regel besagt, daß bei jedem Studenten in diesem Fall zuerst eine Instanz erfragt wird (FI). Die nächsten Regeln bestimmen die Abfolge der sekundären Präsentationsformen für einen bestimmten Studententyp. Bei einem Studenten mit niedriger kognitiver Fähigkeit wird die Frage mit allen möglichen ergänzenden Elaborationen nach ihrer Präsentation spezifiziert (Regeln 2). Bei einem Studenten mit mittlerer kognitiver Fähigkeit wird der Kontext

(FI.eg.i.kt) nicht mehr dargelegt. Bei Studenten mit niedriger kognitiver Fähigkeit wird bei Schwierigkeiten nahezu sofort, bei Studenten mit mittlerer kognitiver Fähigkeit wird erst nach kurzer Zeit interveniert. Falls der Student die Frage nicht beantworten kann (Regeln 3, 6), wird ihm schrittweise, bis er zu einer Antwort fähig ist, Teile der Lösung (FI.rm.i.tl), die Beschreibung dieser Teillösung (FI.rm.i.dtl), die ganze Lösung (FI.rm.i.ls) und die Nachbearbeitung der ganzen Lösung (FI.rm.i.dls) dargeboten. Bei dem Studenten mit mittlerer kognitiver Fähigkeit wird erst noch versucht, durch die noch nicht stattgefundene Präsentation des Kontextes (FI.eg.i.kt) das Verständnis der Frage zu erreichen.
Falls der Lernende bei der Antwort einen Fehler macht, bekommt er bei mittlerer kognitiver Fähigkeit erst nur die Lokalisation des Fehlers (FI.rm.i.lk), mit niedriger kognitiver Fähigkeit sofort die Beschreibung des Fehlers (FI.rm.i.b) und dann in aufsteigender Genauigkeit die Elaborationen bis hin zur Erklärung der Lösung (FI.rm.i.dls). Diese Folge wird aber nur so weit durchlaufen, bis der Student den Fehler selbst verbessern kann (Regeln 4, 7).

In der Realität ist es oft so, daß nicht jeder der in Abbildung 3 und 4 aufgeführten Präsentationsformen eine konkrete Komponente zugeordnet werden kann, bzw. eine Präsentationsform von mehreren Komponenten umgesetzt wird. Hier müssen zusätzlich Heuristiken zur **Generalisierung bzw. Diskrimination von Präsentationsformen** entworfen werden. Bei der Generalisierung von Präsentationsformen könnte man z. B. folgendermaßen vorgehen: falls eine Menge von Elaborationen verlangt wird (Regel 2, 5), sollten alle aus dieser Menge im ILS vorhandenen Präsentationsformen gezeigt werden. Ist eine Abfolge von Elaborationen bis hin zu einer bestimmten Reaktion des Studenten gefordert (Regeln 3, 4, 6, 7), sollte mit der ungenauesten im ILS vorhandenen Präsentation begonnen und in aufsteigender Genauigkeit fortgefahren werden. Für die korrekte Umsetzung einer Lehrstrategie ist es jedoch wichtig, daß möglichst viele der dort geforderten Präsentationsformen vorhanden sind. Falls einer Elaborationsart mehrere Komponenten eines ILS zugeordnet sind, könnte anhand der verwendeten Medien eine Auswahl getroffen werden. Um den Unterricht anschaulich zu machen, sollte die Verwendung der Medien möglichst abwechslungsreich sein. Falls jedoch der Student bestimmte Medien bevorzugt, können diese verstärkt verwendet werden.

5 Beispiel: Integration der Komponenten von SYPROS

In diesem Abschnitt soll nun für ein existierendes Lehrsystem die Zuordnung der Komponenten beispielhaft durchgeführt werden. Wir verwenden hier das intelligente Lehrsystem SYPROS, das die **SY**nchronisation paralleler **PRO**zesse mit **S**emaphoren lehrt (vgl. z. B. [HeGo93], [GHK95]). Es ist eines der wenigen Systeme, das eine sehr große Auswahl von sekundären Präsentationsformen zur Verfügung stellt.

Die meisten dieser Werkzeuge zur Steuerung des Lehr–/Lernprozesses verwenden als Arbeitsgrundlage den **Ziel–Plan–Baum**, der die interne Wissensdarstellung einer Problemstellung ist. Er wird mit Hilfe von Wissen in den Zielen und Plänen der Wissensbasis zu jeder denkbaren Aufgabe automatisch erstellt. Da die Synchronisation von parallelen Prozessen ein Arbeitsvorgang ist, der in einzelnen Schritten abläuft, sind die **Lehrinhalte**, die in diesem System vermittelt werden, Prozeduren. Daher wird jedes Ziel und jeder Plan in der Wissensbasis dem Lehrinhalt „Prozedur“ zugeordnet.

Der Lehrplan von SYPROS umfaßt im Moment eine nach Themen gegliederte Menge von Aufgaben, die vom Studenten in beliebiger Reihenfolge bearbeitet werden können. Voraussetzung ist, daß der Lernende mit den Grundlagen der parallelen Programmierung mit Semaphoren bereits z. B. in einer Vorlesung vertraut gemacht worden ist. Diese Kenntnisse soll er jetzt bei der Bearbeitung der Aufgaben anwenden. Daher existieren in SYPROS nur **Lehrziele** vom Typ „Anwendung einer Prozedur“.

Bei dieser Art von Lehrziel wird im Regelkatalog als **primäre Präsentationsform** das „Erfragen einer Instanz“ verlangt (siehe Regel 1 in Abbildung 6). Dies sind in SYPROS die einzelnen

Aufgaben, die dem Studenten nacheinander dargeboten werden.

Zur Steuerung des Lernprozesses bietet SYPROS folgende Werkzeuge an, die jeweils verschiedenen **sekundären Präsentationsformen** zugeordnet werden können.

- **Aufgabenpräsentation**:
  Die Aufgabe kann sowohl formal als auch verbal beschrieben werden (siehe Abbildung 7). Da hier zwei verschiedene Medien verwendet werden, visuelle (VS) bzw. verbale Vermitt-

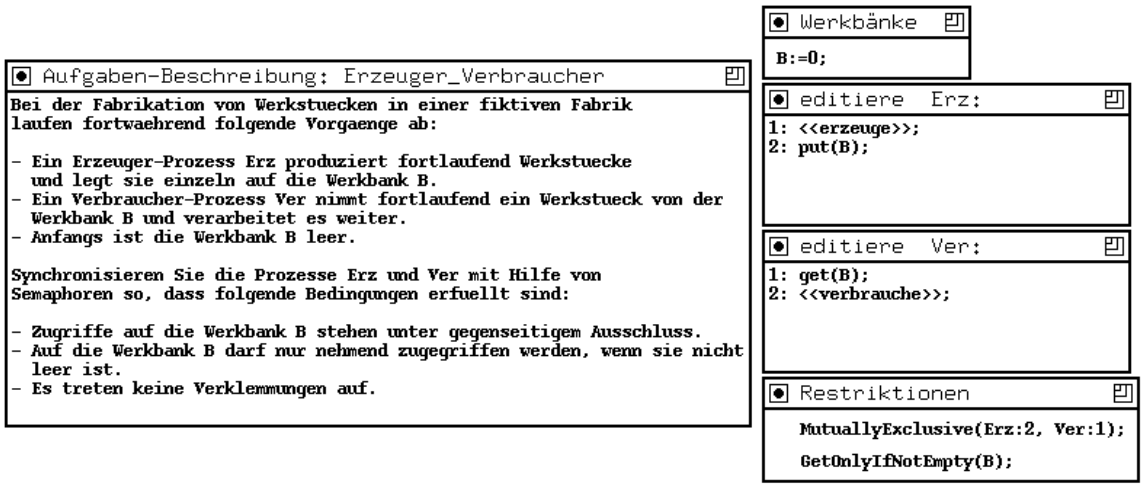


Abbildung 7: Die verschiedenen Möglichkeiten zur Präsentation einer Aufgabe

  lung (VB), ist es möglich die in den Regeln 2 und 5 vorgeschlagene Elaboration FI.eg.i.r zu verwenden.

- **Lösungsgenerator**:
  Dieser baut anhand der notwendigen Ziele im Ziel–Plan–Baum und deren richtiger Pläne eine Lösung auf. So kann sowohl für einen Teil der Aufgabe bzw. das ganze Problem eine Lösung gezeigt werden. Als Medium wird die visuelle Vermittlung (VS) gewählt. Diese Komponente kann man der sekundären Präsentationsform FI.rm.i.tl und FI.rm.i.ls zuordnen.

- **Lösungserklärung**:
  Diese nutzt ebenfalls die Struktur des Ziel–Plan–Baumes und das Wissen in den Zielen und Plänen, um die einzelnen Operationen in der Lösung zu erklären, falls eine genauere Begründung der Lösung erwünscht ist (siehe Abbildung 8). Dies geschieht mit Wor-

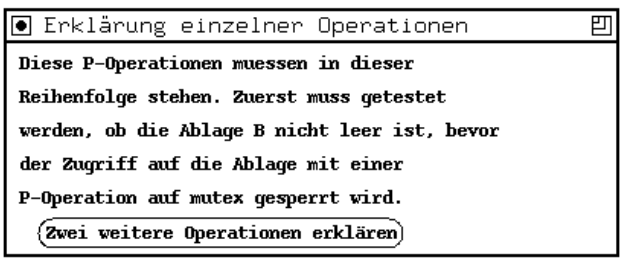


Abbildung 8: Die Erklärung eines Teils der Lösung

  ten, daher wird das Medium verbale Vermittlung (VB) verwendet. In der „Component Display“–Theorie kann man diese Komponente den Präsentationsformen FI.rm.i.dtl und FI.rm.i.dls zuordnen, je nachdem, ob ein Teil oder die ganze Lösung erklärt wird.

- **Hypothesentest**:
  Hier kann der Student einen Teil der Lösung auswählen und auf ihre Korrektheit hin überprüfen lassen (siehe Abbildung 9). Da nur über die Richtigkeit der Lösung befunden wird, kann man diese Komponente der Präsentationsform FI.rm.i.d zuordnen. Das Medium ist wieder die verbale Vermittlung (VB).

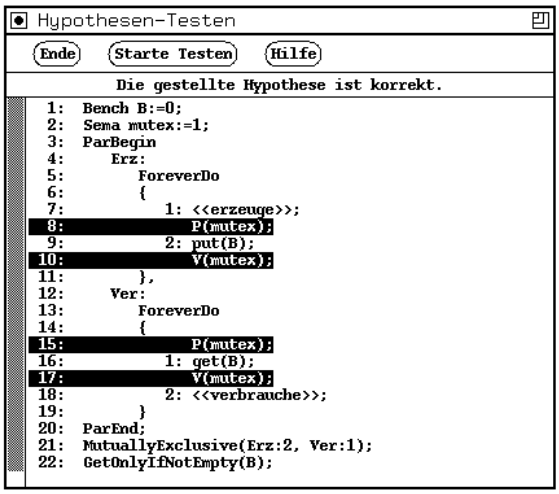


Abbildung 9: Ein Hypothesentest

- **Fehlererklärungskomponente**:
  Diese Komponente nutzt auch die Struktur des Ziel–Plan–Baumes und das in den Zielen und Plänen repräsentierte Wissen, um Fehler zu lokalisieren, zu beschreiben und einen Weg zu ihrer Behebung aufzuzeigen. Je nach Ausprägung der Fehlererklärung kann man sie einer der Präsentationsformen FI.rm.i.lk, FI.rm.i.b, FI.rm.i.tl oder FI.rm.i.ls zuordnen. Allerdings wird hier die Verbesserung bzw. Lösung nicht durch eine visuelle Darstellung, sondern mit Worten (VB) vermittelt. Obwohl man also bereits zwei Komponenten für FI.rm.i.tl und FI.rm.i.ls zur Verfügung hat, kann man sie durch das verwendete Medium unterscheiden.

- **Hypertext**:
  Für jede Wissenseinheit existiert hier ein Dokument, das das Wissen in den Zielen und Plänen verbalisiert und Hintergrundinformation zu den einzelnen Wissenseinheiten liefert. Dies ist also eine Elaboration der Lösungserklärung und kann durch die Präsentationsform FI.rm.i.dls.kt beschrieben werden, die eine Elaboration von FI.rm.i.dls ist. Das Medium ist „Hypermedia“ (HM).

- **Simulation**:
  Diese Komponente arbeitet als einzige unabhängig vom Ziel–Plan–Baum. Hier kann der Student seinen aktuellen Stand der Lösung ablaufen lassen und sehen, ob er noch Fehler enthält. Falls dies der Fall ist, wird lediglich die Fehlerart angezeigt (siehe Abbildung 10). Daher ordnet man diese Komponente der Präsentation FM.rm.i.b zu. Das Medium ist die Simulation, die der Lerner selbst bedienen kann (SL). So kann diese Komponente auch sinnvoll von der Fehlerbeschreibung der Fehlererklärungskomponente unterschieden werden.

Außerdem können in SYPROS abhängig vom Studententyp verschiedene Bemerkungen zur Motivationssteigerung generiert werden (FI.rm.m.*).

Bis auf die sekundären Präsentationsformen zur Ergänzung der Aufgabenstellung (FI.eg.*), von denen in SYPROS nur die Präsentation in anderer Repräsentation (FI.eg.i.r) vorhanden ist, konnte jeder in Abbildung 6 vorgeschlagenen Präsentationsform eine Komponente von SYPROS

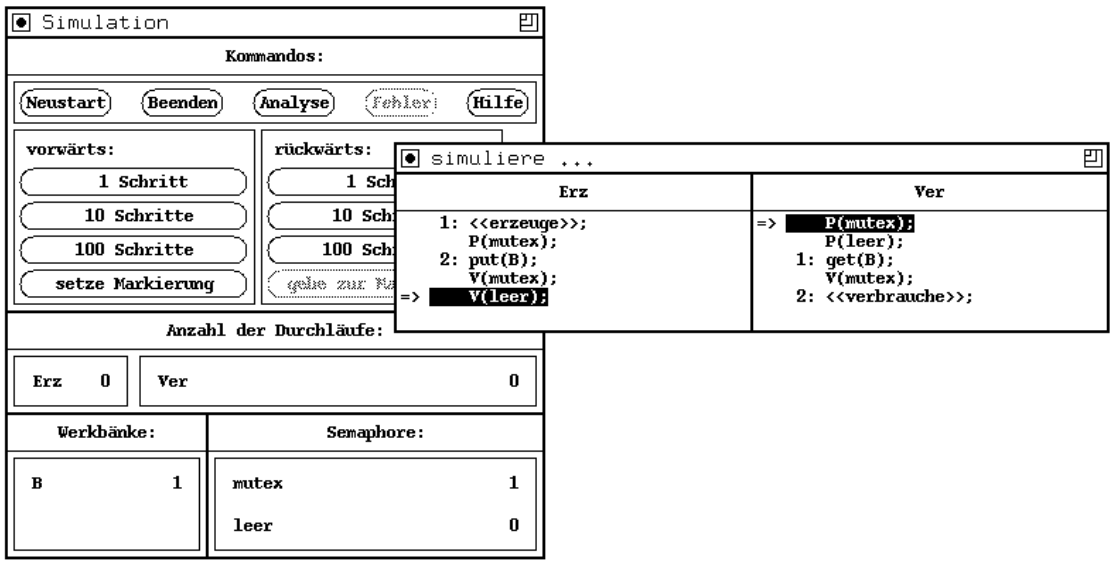


Abbildung 10: Die Simulation

zugeordnet werden. Die durch die Regeln beschriebene Lehrstrategie kann also mit minimalen Generalisierungen in SYPROS ablaufen.

6 Zusammenfassung und Ausblick

Beim wissensbasierten Ansatz kommt man durch die zusätzliche Repräsentation von Relationswissen der Integration von Komponenten eines oder durchaus auch verschiedener Lehrsysteme ein Stück näher. Bei einer solchen Architektur eines Lehrsystems ist es möglich, andere Komponenten, z. B. zusätzliche Werkzeuge zur Vermittlung des Wissens, leicht zu integrieren. Es muß lediglich ihre Funktionsweise in der TBox des Relationswissens formalisiert und einer Komponente der allgemeinen Beschreibung eines Unterrichtsprozesses zugeordnet werden. Dadurch wird dem Steuerungsmodul bekannt gemacht, um welche Komponente eines ILS es sich handelt und wie sie mit den anderen Komponenten zusammenhängt. Das Werkzeug selber muß zur Integration nicht über eine besondere Schnittstelle verfügen. Dadurch ist das System in der Lage, diese neue Komponente im Dialog entsprechend richtig einzusetzen. Im Prinzip ist es möglich, jedes System ohne größere Umstellungen um ein Steuerungsmodul, Relationswissen und das hier präsentierte didaktische Wissen zu erweitern.

Falls man die Architektur dieses Ansatzes noch um sog. Dialogwissen, das ist Wissen über den bisherigen Dialog mit dem Studenten, erweitert, kann man außerdem noch dem Problem der Adaption begegnen. Nach [Moor95] ziehen die in heutigen intelligenten Lehrsystemen eingesetzten Hilfen meist nicht den vorherigen Dialog in Betracht. Dadurch entstehen unnatürliche, sich wiederholende Äußerungen, Ähnlichkeiten und Unterschiede können nicht hervorgehoben werden. Für den Lernenden ist es sehr schwer, semantische Beziehungen zwischen den vermittelten Lehreinheiten zu bilden, da dies nicht von den Hilfekomponenten unterstützt wird.

Um Kenntnis über den bisherigen Dialog zu haben, wird dieser in der ABox des KL–ONE–Systems gespeichert (siehe Abschnitt 3.3). Bei jeder relevanten Aktion des Systems werden die entsprechenden Individuen der in der TBox definierten Präsentationen in die ABox eingefügt. Falls eine Hilfe präsentiert werden soll, muß nun vorher mittels einer Abfrage der ABox geprüft werden, ob bereits eine Hilfe zum gleichen Sachverhalt präsentiert wurde. Der Dialog kann dann an diese Präsentation angepaßt werden.

Daher hat das Relationswissen auch seinen Namen, denn indem es die Eigenschaften jeder Komponente in einem ILS beschreibt, sagt es auch gleichzeitig etwas über die Beziehungen der

Komponenten, die Relationen, untereinander aus. Nur dadurch ist eine Adaption möglich.

Ein weiterer möglicher Ansatzpunkt für Forschungen wäre die Erweiterung des didaktischen Wissens um die Fähigkeit, didaktische Regeln aufgrund von Erfolgen beim Lehren zu verstärken bzw. Regeln, die den Lernerfolg nicht gefördert haben, zu verwerfen. Da es geplant ist, die didaktischen Regeln durch Fuzzy-Methoden zu realisieren, dürfte dies durchaus realisierbar sein.

Außerdem werden im Moment noch nicht die Handlungsphasen des Rubikonmodells (siehe [Heck89]) bei der Auswahl der Hilfen und deren Medien unterstützt, da es in einem Lehrsystem sehr schwierig ist, zu erkennen, in welcher Phase sich der Lernende gerade befindet.

Literatur

[BrSch85] Brachman, R. J.; Schmolze, J. G.: An overview of the KL–ONE knowledge representation system. In: Cognitive Science 9(2), 1985.

[Brus95] Brusilovsky, P.: Intelligent learning environments: the case for integration and adaption. In: Greer, J.: *Proc. of the World Conference on Artificial Intelligence in Education AI–ED*, Washington, 1995.

[Dale69] Dale, E. A.: Audiovisual methods in teaching. Holt, Rinehart & Winston, New York, 1969.

[FGL96] Frasson, C.; Gauthier, G.; Lesgold, A. (Hrsg.): Proceedings of the International Conference on Intelligent Tutoring Systems ITS, Montréal, 1996.

[FMAG96] Frasson, C.; Mengelle, T.; Aïmeur, E.; Gouardères, G.: An actor–based architecture for intelligent tutoring systems. In: Frasson, C.; Gauthier, G.; Lesgold, A. (Hrsg.): *Proceedings of the International Conference on Intelligent Tutoring Systems ITS*, Montréal, 1996.

[GHK95] Gonschorek, Meike; Herzog, Christian; Kluge, Eva: Eine Didaktikkomponente für SYPROS — Studentenmodellierung, Lernzielstrukturierung und Lernerführung. In: Huber–Wäschle, F; Schauer, H.; Widmayer, P. (Hrsg.): GISI' 95, Springer, 1995.

[Harr97] Harrer, A.: Kognitive und motivationale Gesichtspunkte in einer Lehrstrategie. In: diesem Band, 1997.

[Heck89] Heckhausen, H.: Motivation und Handeln. Springer, Berlin, 1989.

[HeGo93] Herzog, C.; Gonschorek, M.: Generation of adaptive explanations in ITS with hierarchical knowledge domains. In: Proc. International Conference on Computers in Education, Taiwan, December 1993: S. 1–6.

[Merr94] Merrill, M. D.: Instructional Design Theory. Educational Technology Publications, Englewood Cliffs, NJ, 1994.

[Moor95] Moore, J. D.: Discourse generation for instructional applications: making computer tutors more like humans. In: Greer, J.: *Proc. of the World Conference on Artificial Intelligence in Education AI–ED*, Washington, 1995.

[Müll96] Müller, J. P. : The design of intelligent agents: a layered approach. Lecture notes in computer science, Springer , Berlin, 1996.

Adapting Knowledge Structures In Dynamic Domains

DIETRICH ALBERT, RAINER KALUSCHA [1]

Institut für Psychologie, Abteilung für Allgemeine Psychologie
Karl–Franzens–Universität Graz, Universitätsplatz 2/III, A–8010 Graz, Austria

E-Mail: dietrich.albert@kfunigraz.ac.at or rainer.kaluscha@kfunigraz.ac.at

Abstract:

For effective processes of knowledge assessment, teaching, information presentation and information use, the information should be presented on basis of a structure which is defined by prerequisite relationships *(Which information do I have to know before I'm able to understand and integrate the newly presented information ?)*. This kind of prerequisite relation for information units has to be determined and tested before using it.

A non-numerical theory for representing these structures is already worked out, and several techniques have been suggested to obtain these structures, e.g.

- mass data collection,
- analysis of didactics and curricula,
- querying experts on prerequisite relationships, competence and performance
- analysis of demands
- analysis of competence and performance,

and also techniques are available to test the obtained structures empirically, which – in principle – is a necessary precondition for using them.

However, domains and thus the related knowledge nowadays change very fast – an impressive example is the internet.

Therefore techniques for convergent structuring, for dynamic testing and for dynamic refinement have to be developed. This article will give an overview of existing techniques in the static case, i.e. the domain doesn't change at all or only very slowly, and discuss advancements towards the use of these techniques in dynamic domains.

[1] alphabetical order, send requests for reprints to either author

1. Introduction

Due to modern communication techniques, a vast amount of information and knowledge has become accessible for many people. But to find the relevant information in the floods, is still a difficult task. Search engines are not of much help, as they look only for keywords, and don't take into account the inherent structure of the information and the knowledge.

Only dividing the knowledge in small parts and structuring it leads to an easy to understand way of presentation. We'll call these units of information (sometimes also referred to as learning objects) *items*. For example, these items may be facts, concepts, problems or skills from a given domain of knowledge.

After defining a domain of knowledge and identifying the relevant items within the chosen domain, the question arises whether these items may be presented independently from each other or not. Usually, relationships exist between items, e.g. *prerequisite relationships* or *surmise relationships*. Roughly speaking, a prerequisite relationship means that people – as a necessary precondition – do have to know about one or more items in order to master another item. A surmise relationship indicates that a person who understands about a certain item will (probably) know about one or more other items.

The theory of knowledge spaces (Doignon & Falmagne, 1985) provides a formal framework (based on the theory of orders and lattices) for the representation of such relationships.

2. Knowledge Structures

Several ways for representing knowledge structures in mathematical models are known, e.g. *knowledge spaces* and *surmise systems* (Doignon & Falmagne, 1985), *and–or–graphs* (Nilsson, 1980) or *implication relations* (Dowling, 1993). All of these representations have shown to be equivalent because bijective mappings between them exist.

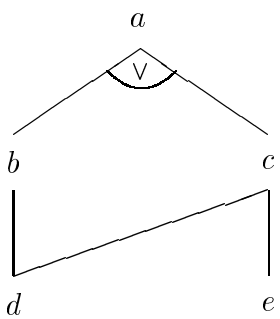


Figure 1: Example of an *and–or–graph*

Figure 1 shows an example of an *and–or–graph* of a domain $Q := \{a, b, c, d, e\}$ with five items. *Or–nodes* are marked by a curved line and a $\vee$ symbol, e.g. to master item a it is sufficient to know about item b *or* item c (or both of them). *And–nodes* don't have a special mark, e.g. a person has to understand about both items d *and* e in order to

master item c. In this sense, items d and e are *prerequisites* for c. Vice versa, knowing a person masters item c, it can be *surmised* that this person will also master items d and e.

The corresponding *knowledge space* $\mathcal{K}$ is the set of all subsets K of the domain set Q which conform to the relationships mentioned before:

$K := \{\emptyset, \{d\}, \{e\}, \{b,d\}, \{d,e\}, \{a,b,d\}, \{b,d,e\}, \{c,d,e\}, \{a,b,d,e\}, \{a,b,d,e\}, \{b,c,d,e\}, \{a,b,c,d,e\}\}$

For example, the subset $\{b\}$ is not a member of the knowledge space, since item d is a prerequisite for item b, which yields $\{b,d\}$ as the smallest (with regard to the ordering given by set inclusion) member of the knowledge space $\mathcal{K}$ which contains item b. The members of the knowledge space are called *knowledge states*.

A knowledge space $\mathcal{K}$ on a set of items Q has (by definition) the following three properties:

$$P1 : \emptyset \in \mathcal{K} \qquad P2 : Q \in \mathcal{K} \qquad P3 : \bigwedge_{K_1, K_2 \in \mathcal{K}} K_1 \cup K_2 \in \mathcal{K}$$

$P1$ and $P2$ make sure that the perfect novice, who doesn't master any item, and the perfect expert, who knows about all the items in the domain Q, are represented in the knowledge space. $P3$ means a knowledge space is closed under union. An interpretation of this could be that two persons in the knowledge states K_1 and K_2 may share their knowledge and teach each other, so that both end up in the state $K_1 \cup K_2$ which is again a member of the knowledge space.

Please note that these knowledge structures do not only give insight which paths a person can follow when moving within a domain, they also allow for the adaptive and efficient assessment of knowledge (Falmagne 1985, Dowling & Kaluscha, 1995).

Knowing that these knowledge structures are useful for knowledge assessment, teaching and information presentation, techniques are needed for establishing them. Several approaches are available and will be discussed in the following sections.

Up to now only the static case has been discussed, i.e. the domain has been fixed, and the structure of a fixed set of items had to be determined. In this paper, also the case of more dynamic domains, where

1. items [2] may become obsolete or
2. new items enter the domain from time to time

will be discussed.

3. Mass Data Collection

One approach is to collect answer patterns in a given domain from a large number of subjects. But due to the huge number of possible knowledge states which may increase

[2] We will not discuss the trivial case where the obsolete item or a new item is equivalent to other items in the domain. New or obsolete items should be considered as new or obsolete item types or item classes.

exponentially with the number of items considered, this method is rarely feasible. Furthermore, the resulting knowledge structures will be extremly dependent on the sample, and thus might be no longer valid when applied to other populations.

However, if the data can be collected without too much effort, and the domain does not consist of a large number of items, analyzing mass data might be worth trying.

For example, in the state of New York passing the Regents Competency Test in Mathematics is a requirement for graduation from high school.

Villano (1991) chose 20 items from this test and constructed a knowledge structure both by analyzing the answer patterns of 67,204 participants in the test and the judgements of experts.

However, in a dynamic domain, mass data collection has an important drawback. If an item has to be deleted, there's no problem – the answer patterns without this item may be used for a new analysis. But if a new item appears, retesting the original sample with the new item would be inevitable which often is not feasible. In the example of the New York Regents Competency Test this enterprise would mean to contact all 67,204 students that took part in the test in 1987, collect their answers to the new item and insert the result in the individuals' answer pattern. Furthermore, these answer patterns are often collected anonymously for the protection of the participants, so that it wouldn't be known which new result belongs to which answer pattern.

4. Analysis Of Didactics and Curricula

This approach hasn't been worked out scientifically yet. To our knowledge, researchers in didactics haven't yet formulated a formal model for their results suitable for deriving relationships between items.

The idea is to use knowledge from didactics and analyze existing curricula for a given domain.

Once the order in which items from the domain are taught is known, inferences concerning prerequisite relationships can be drawn.

For example, if item a is presented after item b, obviously a cannot be a prerequisite for b. On the other hand, an item taught shortly before another is liable to be (one of) its prerequisite(s).

In the dynamic case, new items have to be integrated in existing curricula. From the position where they're placed probable prerequisite relationships can be identified, as discussed above. For updating the knowledge structure the same techniques could be used as described in the next section.

5. Querying Experts

One way of determining a knowledge structure for a given domain is to query experts on prerequisite relationships (Falmagne et al., 1990; Dowling 1991, Koppen 1993, Kaluscha 1994).

In the first step, the domain of knowledge to be investigated has to be defined and the relevant items within the domain have to be identified, usually with the help of experts

in the field.

In a second step, the relationships between these items have to be determined. Please note that the expertise needed for the first and the second step may be different – defining the domain is a task for experts from the field, while determining the relationships between items usually requires additional psychological and pedagogical knowledge as well as experience in teaching the items (for an example see Held, Schrepp and Fries, 1995). For instance, if the domain chosen was elementary mathematics, mathematicians could be a help for accomplishing the first step, while experienced mathematics teachers should be preferred as experts in the second step.

A computer based querying procedure has been implemented by Kaluscha (1994) based on the algorithm due to Dowling (1993). Koch and Quante have improved the procedure and provided an enriched user interface for the experts (Dowling, Koch and Quante, 1996).

As the experts usually are not able to give the complete list of prerequisite relationships, which is due to the huge number of possible relationships even in small domains, the experts have to judge *standard form assertions*, e.g.

> `Imagine a person who does not master the items` $p_1, \ldots, p_k$.
> `Is it then (practically) certain that this person does not master item` q `?`

The items $p_1, \ldots, p_k$ are called the *premise (set)*, and the item q is called *consequence*. The index k is called the *size of the premise*. For judged assertions, the index k takes values between 1 and $N - 1$ where $N := |Q|$ is the total number of items considered.

The experts either accepts or rejects these assertions. By the rules of mathematical logic, inferences can be drawn from previous judgements, and thus the number of standard form assertions to be presented to the experts can be considerably reduced (Falmagne et al., 1990; Koppen, 1993; Dowling, 1993). If a suitable querying strategy is used which is optimized to present assertions for judgement which allow for a large number of inferences the necessary number of judgements is again considerably reduced (Dowling & Kaluscha, 1995). The querying procedure finishes when all standard form assertions have been judged by the expert or by inference from previous answers of the expert.

The first dynamic case, to remove an obsolete item i_{obs} from a knowledge structure, is a rather simple task if a knowledge space $\mathcal{K}$ is chosen as representation of the knowledge structure:

> Remove from all knowledge states $K \in \mathcal{K}$ the obsolete item i_{obs} (and eliminate duplicate knowledge states). [3]

This gives immediately a new knowledge space $\hat{\mathcal{K}}$, from which the other representations can be derived (Dowling, 1993; Koppen, 1993). Please note that the algorithm mentioned above preserves the three properties $P1$, $P2$ and $P3$ of a knowledge space, i.e. $\hat{\mathcal{K}}$ is indeed a knowledge space on the domain $Q \setminus \{i_{obs}\}$.

In the second dynamic case, adding a new item i_{new} is a more complex problem. In order to preserve property $P3$ (closedness under union) the introduction of the new item can result in two times as much knowledge states as before.

[3] This is, by the way, the trace of $\mathcal{K}$ on $Q \setminus i_{obs}$ (Falmagne & Doignon, to appear: Def. 7)

On the one hand, if we choose a simple approach and assume there are no relationships between the new item i_{new} and the other items $i_k \in Q$, for each knowledge state $K \in \mathcal{K}$ a knowledge state $\hat{K} := K \cup \{i_{new}\}$ has to be added to obtain the new knowledge space $\hat{\mathcal{K}}$. So we end up with twice as much knowledge states.

On the other hand, it wouldn't be feasible to start the querying procedure all over again, as the experts are rare and their time is too valuable.

So we choose another approach. The querying procedure virtually starts from scratch, but only those questions, i.e. standard form assertions, have to be presented to the expert for judgement which contain the new item i_{new} in the premise or consequence – the answers to the other questions are already known from the previous run of the querying procedure on the domain Q. Furthermore, not all of the standard form assertions containing the new item i_{new} have to be judged by the expert, as judgements can be inferred from the expert's answers to previous questions containing i_{new}.

Thus, already a partial run of the querying procedure yields the new knowledge space $\hat{\mathcal{K}}$ on the domain $Q \cup \{i_{new}\}$.

If several items have become obsolete, or several items have to be newly added, the procedures above can be performed for each obsolete or new item respectively in sequence.

Another possibility for adding several new items at one time could be to first determine the knowledge structure within these new items, and then use techniques for merging two or more existing knowledge structures into a new one which are under development (Falmagne & Doignon, to appear).

6. Analysis Of Demands

Another approach is the analysis of demands that an item imposes on a person (Albert & Held, 1994; Held, 1993, Lukas & Albert, 1993) [4] . For instance, demands can be applying a skill or knowing some facts. The demands can be obtained by e.g. content analysis, didactical knowledge or analysis of the underlying cognitive processes (Schrepp, 1993).

For example, if the domain chosen was computing expressions of the form $a + b \times c$ with $a, b, c \in \mathbb{N}_0$, an example for an item would be the task "compute $1 + 2 \times 3$".

Relevant demands of this kind of tasks are

1. A: adding natural numbers
2. M: multiplying natural numbers
3. P: knowing that multiplication takes precedence over addition if no parentheses are specified

These demands are called *(problem) components*. A component may have several *attributes* (values) from a *component set*. These attributes can be interpreted as a kind of difficulty level.

If component A (adding natural numbers) is considered the following cases may occur:

[4] These papers deal with a subclass of knowledge structures called the quasiordinal spaces which correspond to surmise relations.

1. a_0: $a = 0$, i.e. no addition is required
2. a_1: $a + b \times c < 10$, i.e. only one digit numbers have to be added without considering a carry
3. a_2: $a + b \times c \geq 10$, i.e. only one digit numbers have to be added with considering a carry
4. a_3: two digit numbers are involved
5. a_4: three and more digit numbers are involved

Similiar considerations can be made for component M (multiplying natural numbers) and for component P (knowledge on precedence rule), where the precedence rule may have to be applied or not. Thus each problem can be characterized by a vector of components and their attributes, e.g. the problem "compute $1 + 2 \times 3$" has the following attributes:

- a_1: $1 + 6$ requires no carry operation
- m_1: 2×3 is a multiplication of one digit numbers, resulting in a one digit number
- p_1: the precedence rule has to be applied

Thus the problem is characterized by the vector (a_1, m_1, p_1) which is a member of the Cartesian product of the component sets $A \times M \times P$.

If a partial order (with respect to "difficulty" or prerequisite relations between attributes) exists on each of the component sets, a partial order on the items can easily be derived. An item a is considered a prerequisite for another item b if and only if the attribute value of each of its components is a prerequisite for the corresponding attribute value of the components of the other item (*coordinatewise order*, see Davey and Priestley, 1990, p. 18). For example, let the three partial orders on the components addition, multiplication and precedence be given by

$$a_0 \preceq a_1 \preceq a_2 \preceq a_3 \preceq a_4 \qquad m_0 \preceq m_1 \preceq m_2 \preceq m_3 \preceq m_4 \qquad p_0 \preceq p_1$$

Then a problem of type $\{a_2, m_1, p_1\}$ will be a prerequisite for a problem of type $\{a_3, m_3, p_1\}$ because the prerequisite relationship holds for each component. However, two problems of the types $\{a_0, m_2, p_0\}$ and $\{a_1, m_1, p_1\}$ have no relationship at all, they cannot be compared.

Alternatively, if the components are not of equal importance, a *lexicographical order* can be used (see Davey and Priestley, 1990, p. 19).

For other domains, the ordering principles of *set inclusion* or *sequence inclusion* may be more appropriate (see Albert & Held, 1994 for examples).

The order on the items gives the relationships between items, which can be used the same way as those obtained by querying experts.

What might happen in the dynamic case ? If an item disappears, the task is simple. But if a new item i_{new} has to be added several cases have to be considered:

1. The new item can be characterized by the existing components and attributes. No special action has to be taken, the new item is simply integrated into the ordering of the other items according to its characterizing vector of components and attributes and the ordering principles.

2. The new item induces new attributes on some components, i.e. the component sets grow for some components. For these components, the partial order has to be redefined. While the basic structure of the order of attributes may be preserved, just the new attribute values have to be inserted in the correct position. After that, while the order on the old items stays the same, the relationships between the new and the old items can be derived by the ordering principles mentioned above.

3. The new item i_{new} requires adding a new component C to the structure. Even in this case there are no real difficulties. First all old items have to be analyzed what attribute values they show with respect to the new component. Then two possible cases have to be discussed:

   - For all of the old items, the attribute value of the new component C is the same, e.g. the component isn't present. Then the component set is defined to be $\{c_0, c_1\}$ where c_0 means the component is not present, and c_1 is the attribute value of the new item with respect to the new component. Then the partial order on C is given by $c_0 \preceq c_1$, and again, the order on the items can be derived by the algorithms mentioned above.
   - If the new item does not only induce a new component C, but also gives rise to the insight that the new component has been neglected on the other items, i.e. the old items have different attributes c_i on the component C an appropriate partial order has to be defined on the component set C, and then the relations between the items can be generated by the ordering principles mentioned above.

7. Competence and Performance

Korossy (1993, 1997) extended the theory of knowledge structures by separating *competence* and *performance*. Competence means skills or abilities that enable a persons to solve a problem, and cannot be observed directly. Performance is the behaviour, e.g. the answer that is given, and can be observed.

Of course, competence, demands and performance are related. However, competences are properties of persons, while demands are properties of problems. While a demand requires a competence to fulfill it, the relationship is not a one to one relation. For example, the demand "add two natural numbers" may be met by adding mentally, by adding with the help of paper and pencil or by operating a pocket calculator.

Similarly, when a performance of a person is observed, it is not obvious what underlying competences have contributed to the solution.

Thus Korossy introduced two spaces: a competence space on a set of (elementary) competences, and a performance space on a set of items. These spaces have the same properties as knowledge spaces, and also prerequisite and surmise relations exist. A state

in the competence space describes the competences a person has, while a state in the performance space is given by the set of items that the person can master.

By identifying the relationships between competences and performance on the items, an *interpretation function* and a *representation function* can be defined. The interpretation function maps a problem or an item to the set of all competence states that allow for solving the problem. Vice versa, the representation function maps a competence state to the set of problems which can be mastered with the competences of this state (Korossy 1993, 1997). Please note that it is sufficient to determine one of these functions as the other one can be derived (Korossy, 1996).

In any case, the interpretation function (or the representation function) has to be adjusted when items become obsolete or are inserted into the performance space. The same holds for new competences that might be added to the set of competences, enlarging the competence space. This is simply due to the fact that the domain of the functions changes.

If a new competence is added, the procedure and the problems for determining the new competence space are similiar to that for knowledge spaces as relationships between the new and the old competences should be used to avoid a large increase of the number of competence states. Furthermore, for each of the items has to be checked whether the new competence opens up new ways of solving the item. If so, the interpretation function (or the representation function) has to be updated because now there are more competence states which allow for solving this item. For example, in the elementary school pupils are forbidden to use pocket calculators. Later in high school the competence "using a pocket calculator" comes into play. This competence enables the pupils to solve items of the type "addition of natural numbers", but also "addition of real numbers". This means the interpretation function has to be changed for both item types.

For deleting a competence from the competence space, the same procedure as deleting an item from a knowledge space is used (see section 5).

If a new item is added to the domain, the following cases may occur:

1. The new item can be solved with the existing competences. In this case, the value of representation function has to be augmented by the new item for all competence states that allow for solving the new item. This means that the result of the interpretation function for the new item will be the set of all competence states that allow for solving the new item, as expected.

2. Mastering the new item requires new competences. Now both the steps for adding new competences (see above) and a new item have to be performed.

Schrepp (1993) goes one step further than Korossy and investigates not only the competences of a person, but also the underlying cognitive processes.

A model for these processes also allows to derive the competence structure, and by means of the representation function the performance structure can be obtained (see Schrepp, 1993).

The neccessary actions when a new item arises depend on whether the model of the cognitive processes can explain how a person masters the new item. If so, the new structures can be derived easily. Otherwise, the cognitive model has to be extended which may require further research.

8. Conclusion

Several approaches for determining knowledge structures are known. Though up to now they've been designed for and applied to static domains only they (except for mass data collection) also can be used in domains that are changing dynamically.

The extent to which old results can be reused and the amount of reanalysis needed varies between the different techniques, also depending on the properties of the new item types that occur.

Most of the necessary work can be done by algorithms which have been sketched in this paper. However, they have to be worked out in detail and implemented on computers. Then pilot studies, applying the techniques in dynamic domains, have to be carried out to gather more experience and prove the usefulness of the dynamic approach.

References

Albert, Dietrich and Held, Theo (1994). Establishing knowledge spaces by systematical problem construction. In Dietrich Albert (Ed.), *Knowledge Structures* (pp. 78-112). New York: Springer Verlag .

Albert, Dietrich; Schrepp, Martin and Held, Theo (1994). Construction of knowledge spaces for problem solving in chess. In Fisher, G.H. & Laming, D. (Ed.), *Contributions to Mathematical Psychology, Psychometrics, and Methodology* (pp. 123–135). New York: Springer-Verlag .

Davey, B. and Priestley, H. (1990). *Introduction to lattices and orders.* Cambridge: Cambridge University Press .

Doignon, Jean–Paul, & Falmagne, Jean–Claude (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, **23**, 175–196.

Dowling, Cornelia (1991). *Constructing knowledge structures from the judgements of experts.* Habilitationsschrift, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Dowling, Cornelia (1993). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, **37**, 21–48 .

Dowling, Cornelia and Kaluscha, Rainer (1995). Prerequisite relationships for the adaptive assessment of knowledge. In Jim Greer (Ed.), *Proceedings of the AI–ED 95*, 43–50 .

Dowling, Cornelia; Koch, Uwe and Quante, Kai A. (1996). A new interface for querying experts on prerequisite relationships. *Proceedings of the OzCHI'96*,

Falmagne, Jean–Claude & Doignon, Jean–Paul (to appear). Meshing knowledge structures. In Dowling, Cornelia; Roberts, Fred and Theuns, Peter (Eds.), *Recent Progress in Mathematical Psychology* Hillsdale, USA: Lawrence Earlbaum Associates Ltd. .

Falmagne, Jean–Claude; Koppen, Mathieu; Villano, Michael; Doignon, Jean-Paul and Johannesen, Leila (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, **97**, 201–224 .

Held, Theo (1993). *Establishment and empirical validation of problem structures based on domain specific skills and textual properties.* Doctoral Dissertation, Universität Heidelberg, Germany .

Held, Theo; Schrepp, Martin and Fries, Stefan (1995). Methoden zur Bestimmung von Wissensstrukturen — Eine Vergleichsstudie (Methods For Determining Knowledge Structures — A Comparison Study). *Zeitschrift für Experimentelle Psychologie*, **XLII, 2**, 205–236 .

Kaluscha, Rainer (1994). *Ein effizienter Algorithmus zur Expertenbefragung (An efficient algorithm for querying experts).* Diplomarbeit, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Koppen, Mathieu (1993). Extracting human expertise for constructing knowledge spaces: an algorithm. *Journal of Mathematical Psychology*, **37**, 1–20.

Korossy, Klaus (1993). *Modellierung von Wissen als Kompetenz und Performanz (Modeling Knowledge as Competence and Performance).* Doctoral Dissertation, Universität Heidelberg, Germany .

Korossy, Klaus (1996). *A qualitative–structural approach to the modeling of knowledge.* Report of the Institute of Psychology, University of Heidelberg, Germany .

Korossy, Klaus (1997). Extending the theory of knowledge spaces: a competence–performance approach. *Zeitschrift für Psychologie*, **205**, 53–82 .

Lukas, Josef and Albert, Dietrich (1993). Knowledge assessment based on skill assignment and psychological task analysis. In Strube, Gerhard and Wender, Karl (Eds.), *The Cognitive Psychology of Knowledge, volume 101 of Advances in Psychology.* (pp. 139–160). Amsterdam: North–Holland .

Nilsson, Nils J. (1980). *Principles of artificial intelligence.* Palo Alto: Tioga Publishing Co.

Schrepp, Martin (1993). *Über die Beziehung zwischen kognitiven Prozessen und Wissensräumen beim Problemlösen (On the relatinship between cognitive processes and knowledge spaces in problem solving).* Doctoral Dissertation, Universität Heidelberg, Germany .

Villano, Michael (1991). *Computerized knowledge assessment: Building the knowledge structure and calibrating the assessment routine.* Doctoral Dissertation, New York University, New York. *Dissertation Abstracts International*, **552**, 12B.

Eine generische Erklärungskomponente für Autorensysteme

Ute Trapp, Markus Lusti
Institut für Informatik / WWZ
Petersgraben 51, CH-4051 Basel, Schweiz
e-mail trapp@ifi.wwz.unibas.ch
lusti@ubaclu.unibas.ch

Abstrakt: Es wird untersucht, wie ein bestehendes Multimedia-Autorensystem um eine generische Erklärungskomponente erweitert werden kann. Eine Analyse verschiedener Aufgaben aus dem betriebswirtschaftlichen Bereich zeigt, daß die Generierung von Erklärungen regelbasiertes und prozedurales Wissen erfordert.

Der Beitrag diskutiert eine Wissenserwerbskomponente, mit der das Wissen in einer einheitlichen formalen Sprache eingegeben und automatisch nach Prolog übersetzt wird. Ein Meta-Interpreter konsultiert die resultierende Wissensbasis und erzeugt ein traversierbares Hypernetz, das als Grundlage für Erklärungen dient.

1 Einleitung

Multimediale Autorensysteme (MM-Systeme) bieten gute Möglichkeiten, Lerneinheiten ansprechend und effizient zu gestalten. Die Autorin[1] kann insbesondere Mehrfachwahl-, Zuordnungsaufgaben und Lückentexte erstellen. Kann eine Aufgabe nicht gelöst werden, so wird häufig auf eine zurückliegende Lerneinheit verwiesen.

Die Studentin gelangt so in eine Schleife, die keine neuen Ansätze enthält. Besser ist es, passende und neue Erklärungen zu bieten. Diese Lücke soll unsere Erklärungskomponente teilweise füllen. Bei der Entwicklung von Intelligenten Tutoriellen Systemen (ITS) sind zwei Tendenzen zu beobachten: Die Erstellung von Autorenkomponenten für die Endbenutzerin und die Entwicklung von Modulen für die Softwareentwicklerin. In beiden Fällen steht die Enwicklung von domänenunabhängigen Systemen oder Komponenten noch am Anfang.

Das zur Generierung von Erklärungen benötigte Domänenwissen wird von der Autorin in einer einheitlichen formalen Sprache eingegeben und automatisch

[1] Bei Ausdrücken wie "Autorin" oder "Studentenmodell" ist immer auch die Rede vom anderen Geschlecht.

nach Prolog übersetzt. Ein Metainterpreter erzeugt daraus einen Lösungsbaum, der als Grundlage für Erklärungen dient.

Das Projekt befindet sich noch im Anfangsstadium. Es konzentriert sich zur Zeit auf den Wissenserwerb, die Klassifikation von Aufgabentypen sowie die prinzipielle Architektur des Systems.

1.1 Einführendes Beispiel

Ein einfaches Beispiel aus der Finanzmathematik veranschaulicht das Zusammenspiel eines MM-Systems und der zu entwickelnden Erklärunskomponente. Es werde folgende Aufgabe gestellt [KÖHL92]:

Ein Darlehen von 50000.- wird mit 8% verzinst und mit 4% zuzüglich eingesparter Zinsen getilgt. Stellen Sie den Tilgungsplan auf.

Abbildung 1: Aufgabe

Der Lösungsteil des Lehrbuches enthält dazu folgende Erläuterung:

Da eine Annuitätentilgung vorliegt, kann die Annuität im ersten Jahr als Prozentsatz der Gesamtschuld bestimmt werden. Die Zinsen im 1. Jahr betragen $8\% \cdot 50000 = 4000$. Die Tilgungsrate im 1. Jahr ist $4\% \cdot 50000 = 2000$. Die Annuität beläuft sich damit auf 6000. Nach dem 1. Jahr besteht eine Restschuld von 50000 - 2000 = 48000. Es folgt die Abbildung des vollständigen Tilgungsplanes.

Abbildung 2: Textbaustein

In einem Beispielszenario erstellt die Autorin eine Lektion mit der Aufgabe in einem MM-System. Dann konfiguriert sie die Erklärungskomponente für ihre Aufgabe. Kann die Studentin bei der Bearbeitung der Lerneinheit die Aufgabe nicht lösen, so fordert sie Erklärungen an. Abbildung 3 veranschaulicht die zugehörige Sitzung. Die linke Seite repräsentiert die Aktionen im MM-System, und die rechte Seite zeigt die Aktionen der Erklärungskomponente.

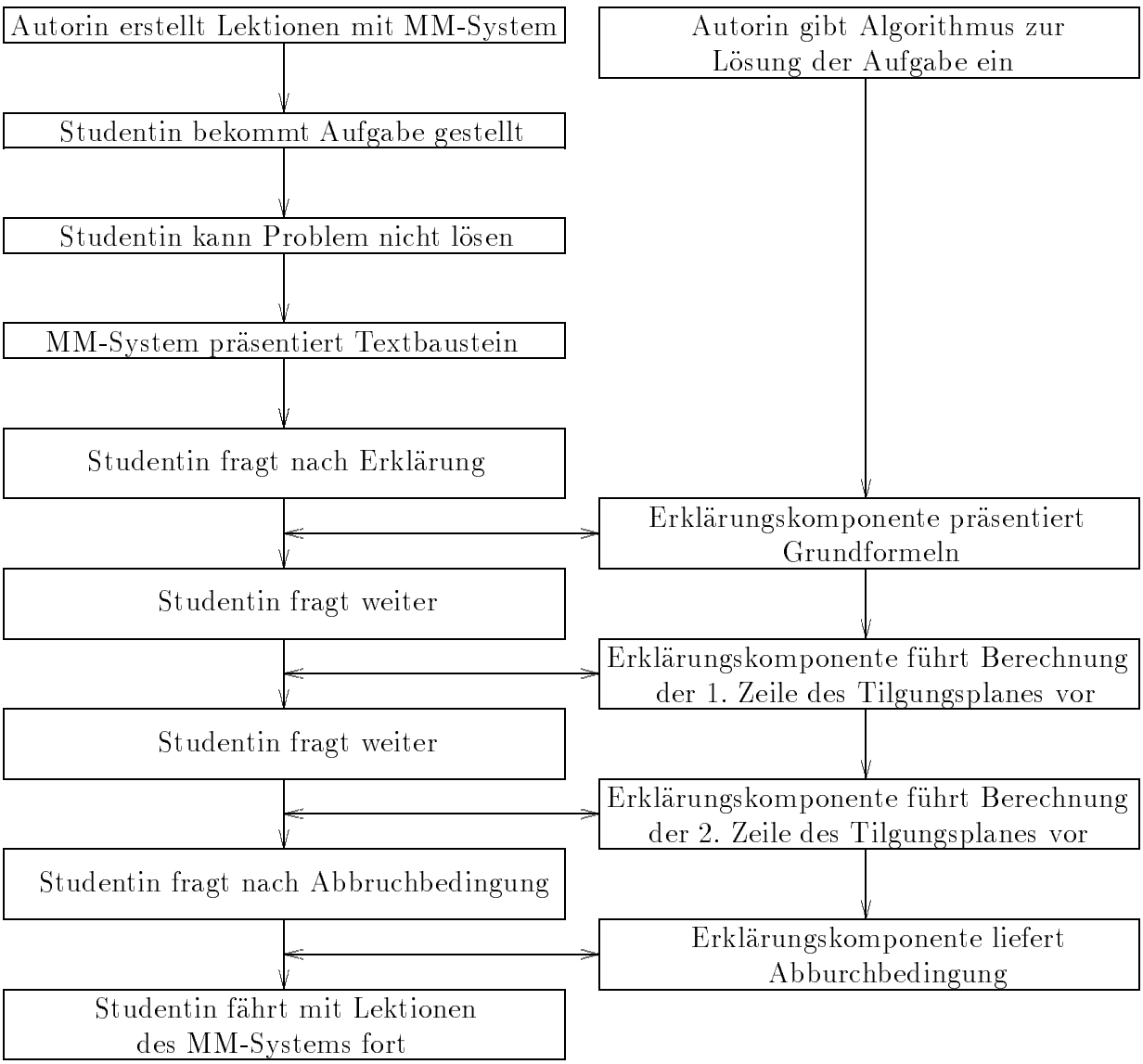


Abbildung 3: Sitzung

2 Erklärungen

2.1 Aus der Perspektive der Lernpsychologie

Lernziele, Aufgaben und Erklärungen stehen in engem Zusammenhang. In einem klassischen ITS übernimmt das Unterrichtsmodul die Aufgabenauswahl. Bei dem hier diskutierten Ansatz legt die Autorin die Aufgaben fest. Entweder konfiguriert sie das Lernsystem derart, daß die Studentin durch die Aufgaben geführt

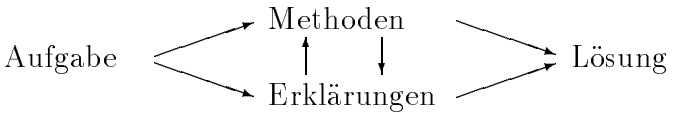


Abbildung 4: Erklärungen

wird oder die Studentin wählt die Aufgaben, die sie bearbeiten möchte, eigenverantwortlich aus einer Gesamtheit von Aufgaben aus.

Für jede Aufgabe gibt es unterschiedliche Methoden, um zur Lösung zu gelangen [EDEL94]. Eine sinnvolle Erklärung unterstützt die gewählte Methode der Studentin oder argumentiert, warum die Methode nicht zur Lösung führt. Erklärungen und Methoden beeinflussen sich also gegenseitig. Die Erklärungskomponente muß somit aufgrund der eingegebenen Methoden geeignete Erklärungen generieren.

2.2 Aus der Perspektive der Implementation

Auf der Implementationsseite werden Erklärungen nach ihrem Funktionsumfang charakterisiert, insbesondere sind hierbei Ausführlichkeit und Tiefe der Erklärung von Interesse [LUST92]. Prinzipiell unterscheidet man folgende Generierungsprozesse:

- *protokollabhängige* Erklärungen werden zur Laufzeit generiert und geben häufig einen Teil des Lösungsbaumes aus.
- *protokollunabhängige* Erklärungen sind in ihrer Struktur und ihrem Inhalt fest, sie bestehen oft aus Textkonserven.

Beide Generierungsverfahren können kombiniert werden: Knoten des Lösungsbaumes können auf Textkonserven zeigen. Insgesamt erhält man ein traversierbares Hypernetz. Der Fragetyp (How, Why, Why not, What if) bestimmt die Art des Traversierens [LUST92].

Die Erklärungen von JA-Tutor [LUST95] basieren auf einem traversierbaren Hypernetz, das von der Autorin über eine Autorenkomponente modifiziert werden kann. Dieses Konzept zur Generierung von Erklärungen für Probleme der Jahresabschlußrechnung wird mit der generischen Erklärungskomponente auf eine größere Klasse von Aufgabentypen übertragen.

3 Aufgabentypen

Tabelle 1 listet exemplarisch verschiedene Aufgabentypen aus dem betriebswirtschaftlichen Bereich auf. Die Klassen ergeben sich aufgrund folgender Kriterien:

- Anzahl der Arbeitsschritte, die bis zur Lösung erforderlich sind (Erklärungstiefe)

- Methoden, die zur Lösung führen (Algorithmen (A) oder Heuristiken (H))
- Wissensdarstellung (deklarativ (d) oder prozedural (p))

Typ	Kurzbeschreibung	Beispiel	Tiefe	A/H	d/p
1	mehrere Formeln berechnen	Zinseszins, Preiselasitizität	$\geq$ Anzahl Formeln	A	d/p
2	lineares Gleichungssystem lösen	Rentenplan	$\geq$ Dimension des Gleichungssystems	A/H	d/p
3	Prozedur ausführen	Tilgungsplan	$\leq$ Anzahl Anweisungen	A	p
4	Textverständnis	Buchungen	≥ 1	A	d
5	Interpretation	Aussagen von Kennzahlen	≥ 1	H	d

Tabelle 1: Charakteristische Aufgaben

Eine Erklärung der zugehörigen Lösungen kann wie folgt aussehen:

- Typ 1: Alle verwendeten Formeln werden angegeben und die zugehörigen Rechnungen ausgeführt.
- Typ 2: Die Lösungsschritte werden vorgeführt. Es wird begründet, warum genau dieses Lösungsverfahren gewählt wurde.
- Typ 3: vgl. Abbildung 3.
- Typ 4: Der Text wird in die Aufgabenstellung übersetzt, dies wird mit den entsprechenden Regeln begründet.
- Typ 5: Es wird dargelegt, welche Regeln aufgrund von welchen Fakten zu welchen Schlüssen führen.

Folgende Informationen benötigt die Erklärungskomponente von der Autorin zur Generierung von Erklärungen:

- prozedurales Wissen:
  - Formeln
  - Prozeduren
- deklaratives Wissen:
  - Formeln
  - Regeln, welche die Auswahl der Formeln bestimmen
  - Regeln zur Termumformung, Differentation, ...

- nicht weiter zu begründendes Wissen:
  - Verweise auf Lektionen des MM-Systems

4 Gewählter Ansatz

Ziel ist, mit minimaler Eingabe der Autorin eine möglichst flexible und sinnvolle Erklärung zu erzeugen.

4.1 Architektur

Die klassische ITS-Architektur mit Unterrichts-, Studenten-, Kommunikations- und Expertenmodul wird folgendermaßen umgesetzt (Abbildung 5): Das Unterrichtsmodul konzentriert sich auf die Auswahl der Aufgaben und wird vom MM-System übernommen. Ein ausgearbeitetes Studentenmodul entfällt. Das Expertenmodul löst Aufgaben und begründet die Lösungen. Es greift zur Generierung von Erklärungen und Lösungen auf die von der Autorin eingegebene Wissensbasis zu.

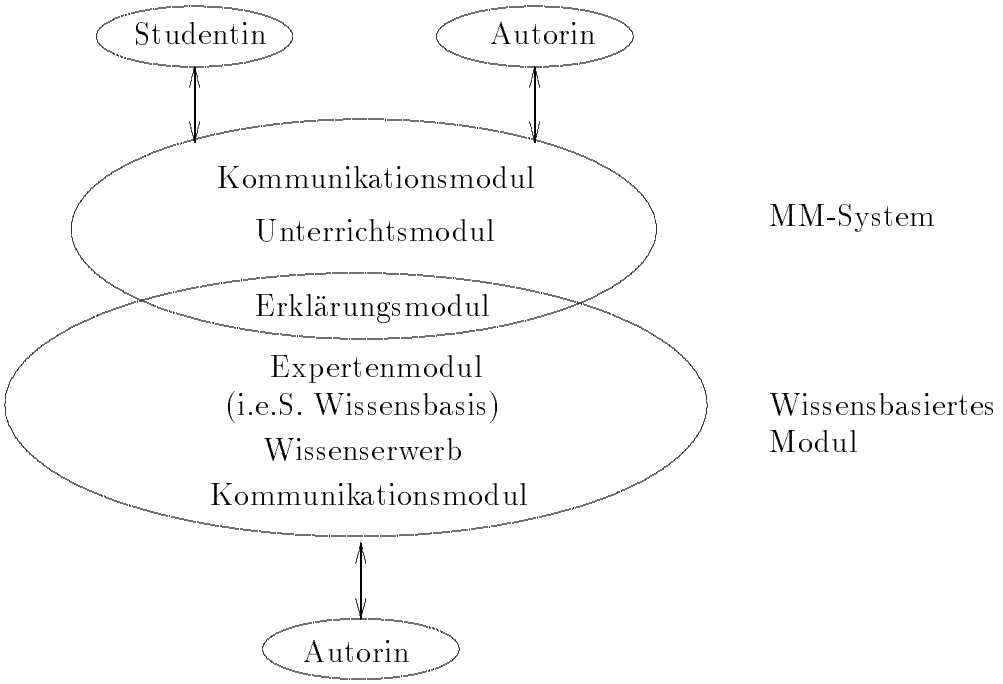


Abbildung 5: Architektur

4.2 Wissenserwerb

Wie in Abschnitt 4 gezeigt wurde, benötigt die Wissensbasis prozedurales und regelbasiertes Wissen. Dieses Wissen wird über eine Pseudosprache eingegeben und nach Prolog übersetzt.

Anhand des einführenden Beispiels von Abschnitt 1.1 wird die Arbeitsweise des Übersetzers veranschaulicht. Abbildung 6 zeigt den zugehörigen Pseudocode, und Abbildung 7 stellt die Übersetzung dar. Der Übersetzer arbeitet wie folgt:

- Das Programm wird als Klausel `start` übersetzt.
- Variablennamen werden `__Name` benannt, wobei `Name` der von der Autorin eingegebene Name ist.
- Jede Wertzuweisung wird als destruktive Wertzuweisung betrachtet. Somit bewirkt jede Wertzuweisung eine Änderung des Variablennamens, `__Name` wird zu `_1Name`, `_1Name` wird zu `_2Name`, ...
- Für jedes Prädikat wird eine Liste von Eingangsparametern und eine Liste von Ausgangsparametern erzeugt. Eingangsparameter sind alle Variablen, die innerhalb des Prädikats benötigt werden, Ausgangsparameter sind alle Variablen, die innerhalb des Prädikats verändert werden.
- Eine `while`-Anweisung wird einerseits als Bedingung im aufrufenden Prädikat und andererseits als rekursives Prädikat mit Abbruchbedingung und Schleifenkörper und als Fakt übersetzt.

```
/*          S: Schuld, t: Tilgungssatz, z: Zinssatz          */
program tilgung([z, t, S])
begin
      a = (z + t) * S /* Annuität */
      j = 0
      while S > a
      begin
            j = j + 1
            zins = S * z
            t = a - zins
            s = s - t
      end
      a = s * (1 + z)
end
```

Abbildung 6: Pseudocode des einführenden Beispiels (Abschnitt 1.1)

```
start([__z,__t,__s]):-
      _1a is ( __z + __t ) * __s,
      _1j is 0,
      p_2while([_t,__z,__zins,_1j,_1a,__s], [_1s,_1t,_1zins,_2j]),
      _2a is _1s * ( 1 + __z )
.
/* ----- While predicat --------- */
p_2while([_t,__z,__zins,__j,__a,__s], [_2s,_2t,_2zins,_2j]):-
      __s > __a,
      _1j is __j + 1,
      _1zins is __s * __z,
      _1t is __a - _1zins,
      _1s is __s - _1t,
      p_2while([_1t,__z,_1zins,_1j,__a,_1s], [_2s,_2t,_2zins,_2j])
.
/* ------ While fact ------------- */
p_2while([_2t,__z,_2zins,_2j,__a,_2s], [_2s,_2t,_2zins,_2j]).
```

Abbildung 7: Übersetzung von Abbildung 6

Die Möglichkeiten und Grenzen des momentanen Übersetzers faßt Tabelle 2 zusammen.

Positiv	*Negativ*
Die Pseudosprache wird in editierbaren Prolog-Quellcode übersetzt.	Zum Übersetzen werden dynamische Fakten benötigt.
Es werden hilfreiche Syntax-Fehlermeldungen ausgegeben.	
Schachtelung von Schleifen und Verzweigungen sind möglich.	Der Code ist nicht immer übersichtlich, besonders bei geschachtelten Schleifen und Verzweigungen.
Die Autorin kann eigene Prozeduren definieren.	Beim Aufrufen einer Prozedur wird nicht überprüft, ob die Argumente vollständig übergeben werden.
Trigonometrische Funktionen, Potenzen und Prolog-Stringprädikate können verwendet werden.	Es werden keine Listen oder Bäume unterstützt.

Tabelle 2: Bewertung des Übersetzers

Der Übersetzer unterstützt erst die Darstellung von Aufgaben mit prozeduralem Charakter, also die Aufgabentypen 3 (vollständig), 1 und 2 (teilweise). Für

die Eingabe eines Regelsystems ist die gewählte Pseudosprache umständlich. Daher sind für den Erwerb von deklarativem Wissen andere Sprachkonstrukte oder Eingabemöglichkeiten zu entwickeln. Dies ist jedoch relativ einfach, denn Prolog ist selbst eine deklarative Sprache. Zusätzlich ist eine Bibliothek mit domänenabhängigem, aber dennoch allgemeinem Wissen erforderlich – zum Beispiel Regeln zur Differentation oder Regeln zum Lösen linearer Gleichungssysteme.

4.3 Erklärung

Ein Lösungsprotokoll macht die Herleitung der Lösung für die Benutzerin transparent und ist somit eine gute Basis für eine Erklärung. Es gibt verschiedene Möglichkeiten, Lösungsprotokolle zu generieren. Die Realisierung durch Metainterpreter hat die Vorteile, Problemlösung und Erklärung voneinander zu trennen und alle vier Grunderklärungsarten (How, Why, Why not und What if) zu ermöglichen [RÄTZ93].

In Prolog ist es einfach, Metainterpreter zu schreiben, also Prolog-Programme zu entwickeln, die Prolog-Programme interpretieren [NEUM88]. Das gesamte Wissen liegt nach der Übersetzung als Prologklauseln vor und kann von einem Metainterpreter zur Erzeugung eines Lösungsbaumes interpretiert werden. Dieser Lösungsbaum kann mit Teilen des MM-Systems verbunden werden, so daß insgesamt ein traversierbares Hypernetz entsteht.

An jedem Knoten des Netzes kann die Studentin nachfragen. Die Art der Traversierung des Hypernetzes liefert die gewünschte Erklärung.

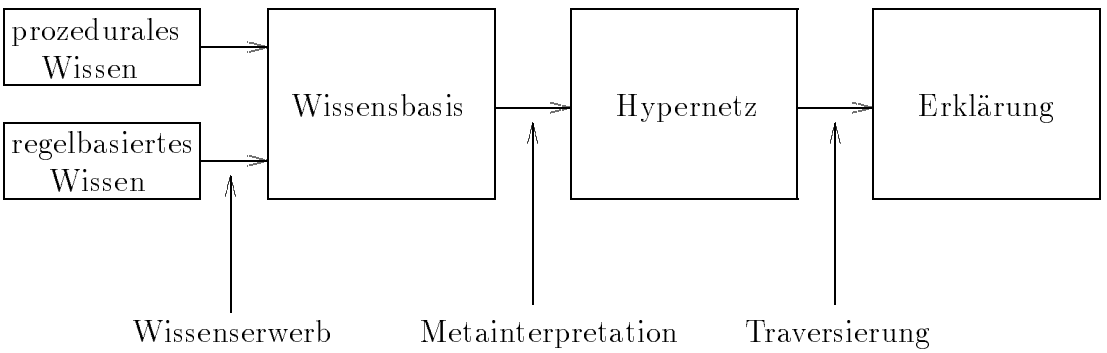


Abbildung 8: Generierung einer Erklärung

5 Zusammenfassung und Ausblick

Zur Generierung von Erklärungen für verschiedene Aufgabentypen werden deklaratives und prozedurales Wissen sowie Verweise auf Fakten (Textbausteine)

benötigt. Dieses Wissen wird über eine formale Sprache eingegeben und automatisch nach Prolog übersetzt.

Folgende Fragen sind dabei zu diskutieren:

- Ist das Ergebnis der Übersetzung, also das resultierende Prologprogramm, eine gute Basis für einen Metainterpreter?
- Wie können Verweise auf Teile des MM-Systems in einer formalen Sprache eingegeben werden und wie soll die Übersetzung nach Prolog aussehen?
- Ist es sinnvoll, für den Erwerb von prozeduralem Wissen und deklarativem Wissen, eine einheitliche formale Sprache zu entwickeln oder sind verschiedene Eingabemöglichkeiten in Betracht zu ziehen?
- Welche Möglichkeiten müssen der Autorin gegeben werden, die Struktur der generierten Erklärungen zu ändern?
- Wie müssen die generierten Erklärungen aussehen, so daß die Studentin in ihrem Problemlösungsprozeß unterstützt wird?

Mit der Entwicklung geeigneter Metainterpreter versuchen wir diese Fragen zu beantworten.

Literatur

[EDEL94] Walter Edelmann, *Lernpsychologie - Eine Einführung*, 4. Auflage, Weinheim, 1994, Beltz Psychologie Verlags Union

[KÖHL92] Harald Köhler, *Finanzmathematik*, 3. Auflage, München, Wien, 1992, Carl Hanser Verlag

[LUST92] Markus Lusti, *Intelligente tutorielle Systeme - Einführung in wissensbasierte Systeme*, Band 15.4 des Handbuchs der Informatik, München, 1992, Oldenbourg-Verlag

[LUST95] Markus Lusti, (1995) *Protocol Driven Hypertext Explanations in a Tutor for Financial Analysis*, Proceedings of AI-ED 95 - 7th World Conference on Artificial Intelligence in Education, 1995, Charlottesville, VA, Association For The Advancement of Computing In Education (AACE)

[NEUM88] G. Neumann, *Metaprogrammierung und Prolog*, 1988, Bonn, Addison-Wesley

[RÄTZ93] Thomas Rätz, *Erklärungen in wissensbasierten Lernsystemen am Beispiel eines Tutors zur Normalisierung von Datenbanken*, Frankfurt/Main, 1993, Peter Lang Verlag

Abstraktionsbasierte Erklärungen für Prognose und Diagnose mit Bayes-Netzen

Andreas Lüdtke
Fachbereich Informatik der Universität Oldenburg - Abteilung Lehr- /Lernsysteme
Andreas.Luedtke@informatik.uni-oldenburg.de

1. Einleitung

Bayes-Netze [Pearl 1988, Neapolitan 1990] sind heute eine weitverbreitete Methode zur Repräsentation und Verarbeitung von Wissen in unsicheren Domänen. Die Konstruktion und die Nutzung von Bayes-Netzen ist in erster Linie ein Wissensakquisitions- und Wissensvermittlungsproblem. Bei der Konstruktion steht die Wissensakquisition im Vordergrund. Domainenexperten sind es gewohnt, die Unsicherheit von Zusammenhängen des relevanten Weltausschnitts mit linguistischen Termen zu beschreiben. Ordnet man diesen Termen eine probabilistische Semantik zu, läßt sich aus der linguistischen Beschreibung ein Bayes-Netz generieren [Schröder, Möbus, Thole-1996]. Bei der Nutzung (Diagnose, Prognose) von Bayes-Netzen steht die Wissensvermittlung im Vordergrund. Mit Bayes-Netzen wird unsicheres Wissen normativ verarbeitet, d. h., durch Bayes-Netze berechnete Ergebnisse sind konsistent mit einer fundierten Theorie, der Wahrscheinlichkeitstheorie. Daraus ergeben sich typische Probleme, die auftreten, wenn eine mathematisch fundierte Methode in Computersystemen verwendet wird, um normative Ergebnisse zu erzielen: Der Gewinn an Konsistenz geht zu Lasten der intuitiven Erklärbarkeit der Ergebnisse. Ein Benutzer wird Ergebnisse nicht akzeptieren, wenn es dem System nicht gelingt, diese in verständlicher und vertrauter Form zu rechtfertigen. Die Vorteile eines Werkzeugs, das auf einer normativen Theorie beruht, zahlen sich erst dann aus, wenn es um eine adäquate Erklärungskomponente ergänzt wird.
Dieser Beitrag soll einen Schritt in Richtung einer solchen Erklärungskomponente für Bayes-Netze liefern. Betrachtet werden Erklärungen von Wahrscheinlichkeitsänderungen, die durch die Eingabe von Evidenz in das Bayes-Netz ausgelöst wurden. Dabei werden die exakten probabilistischen Schlußmethoden auf Strukturen plausiblen Schließens übersetzt. Von Poyla [Poyla-1954]und Pearl [Pearl-1988] wurde gezeigt, daß probabilistsiche Schlußmethoden bestimmten Strukturen menschlichen Schließens sehr ähnlich sind. Diese Erkenntnis wird in der vorgestellten Methode genutzt. Angeknüpft wird an bereits bestehende Ansätze von Druzdzel und Henrion [Druzdzel, Henrion-1993 (a), (b), (c)] und Elsaesser [Elsaesser-1989].
Druzdzel und Henrion abstrahieren die Berechnungen von Wahrscheinlichkeitsänderungen auf Vorzeichen von Wahrscheinlichkeitseinflüssen zwischen Variablen. Durch das Vorzeichen eines Wahrscheinlichkeitseinflusses zwischen zwei Variablen wird angegeben, ob die Wahrscheinlichkeit der einen Variablen steigt, sinkt bzw. unverändert bleibt, wenn die Wahrscheinlichkeit der anderen Variablen steigt bzw. sinkt. In den Erklärungen von Druzdzel und Henrion werden Vorzeichen von Wahrscheinlichkeitsänderungen auf Vorzeichen von Wahrscheinlichkeitseinflüssen zurückgeführt. Der Nachteil dieses Ansatzes besteht darin, daß bzgl. der bedingten Wahrscheinlichkeiten zwischen den Variablen bestimmte restriktive Annahmen gemacht werden, die in der Praxis häufig nicht gelten.
Elsaesser abstrahiert die Wahrscheinlichkeitsberechnungen auf eine Ebene, auf der Aussagen über die Höhe bzw. das Ausmaß der Wahrscheinlichkeitsänderungen möglich sind. Er schlägt vor, Strukturen plausiblen Schließens, wie sie von Polya [Polya-1954] herausgestellt wurden, für die Abstraktion zu verwenden. Exemplarisch führt er vor, wie sich die Beziehung zwischen zwei Variablen auf eine bestimmte Struktur plausiblen Schließens übersetzen läßt.

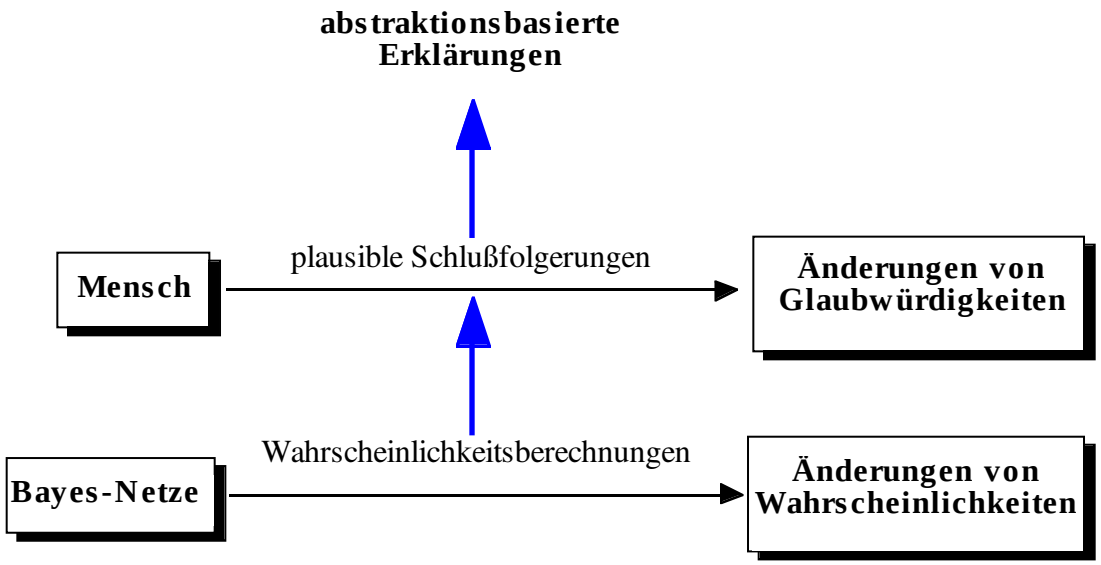


Abb. 1: Generierung von abstraktionsbasierten Erklärungen

Der in diesem Beitrag vorgestellte Ansatz integriert und modifiziert bzw. erweitert die Ansätze von Druzdzel und Henrion und von Elsaesser. Die Abstraktion wird auf drei Ebenen vorgenommen: Auf der abstraktesten Ebene werden Änderungen von Wahrscheinlichkeiten auf Abhängigkeiten zwischen Variablen zurückgeführt. Eine Abhängigkeit zwischen zwei Variablen ist die Voraussetzung dafür, daß es zwischen den beiden Variablen einen Wahrscheinlichkeitseinfluß gibt. Auf der nächsten Ebene werden Vorzeichen von Wahrscheinlichkeitsänderungen durch Vorzeichen von Wahrscheinlichkeitseinflüssen begründet. Auf der untersten Ebene wird das Ausmaß einer Wahrscheinlichkeitsänderung auf das Ausmaß von Wahrscheinlichkeitseinflüssen zurückgeführt.

Im Bereich des plausiblen Schließens steht im Gegensatz zu den Berechnungen in Bayes-Netzen nicht die Generierung von korrekten Ergebnissen im Vordergrund, sondern die Beschreibung menschlicher Schlußmechanismen. Wahrscheinlichkeiten, die von Menschen geschätzt werden, sind häufig fehlerhaft [Kahnemann, Slovic, Tversky-1982]. Die Wahrscheinlichkeiten, die in den Erklärungen dieses Beitrags präsentiert werden, sind hingegen korrekt, da sie auf Basis der Wahrscheinlichkeitstheorie berechnet werden. Die Strukturen plausiblen Schließens werden erst nach der korrekten Berechnung der Wahrscheinlichkeiten verwendet, um die Veränderung der Wahrscheinlichkeiten für den Benutzer verständlich zu machen.

2. Abstraktionsbasierte Erklärungen in elementaren Bayes-Netz-Strukturen

Die erklärungssuchende Frage, auf die eine Antwort generiert werden soll, lautet:

> „Warum ändert sich die Wahrscheinlichkeit von *X*, wenn für *Y* Evidenz eingegeben wird?“

Die Berechnungen der Wahrscheinlichkeiten bilden den Ausgangspunkt der Abstraktion. Zentrales Abstraktionsmittel sind Wahrscheinlichkeitseinflüsse zwischen Variablen. Informell besteht ein Wahrscheinlichkeitseinfluß der Variablen *Y* auf die Variable *X*, wenn aufgrund einer Wahrscheinlichkeitsänderung der Ausprägungen von *Y* die Wahrscheinlichkeiten der Ausprägungen von *X* neu berechnet werden. Bei der Argumentation in den Erklärungen wird die Auswirkung von Evidenz zunächst auf die direkten Nachbarn des entsprechenden Knotens im Bayes-.Netz betrachtet. Jeder der direkten Nachbarn übt wiederum einen Einfluß auf seine eigenen direkten Nachbarn aus. So pflanzt sich die Evidenz durch das Netz fort. Jeder einzelne Schritt läßt sich auf eine bestimmte Struktur plausiblen Schließens zurückführen.

2.1 Erklärungen auf der Abstraktionsebene der Abhängigkeiten zwischen Variablen

Wenn zwei Variablen voneinander abhängig sind, bewirkt die Änderung der Wahrscheinlichkeit einer der Variablen eine Neuberechnung der Wahrscheinlichkeit der anderen Variablen. Abhängigkeiten lassen sich aus der Netzstruktur, in der die Variablen miteinander verbunden sind, ablesen (d-Separation [Neapolitan-1990]). Bei den Beziehungen direkt benachbarter Variablen sollen zwei elementare Netzstrukturen (Abb. 2) unterschieden werden.

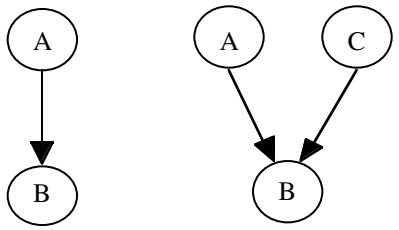


Abb. 2: Zwei elementare Netzstrukturen - die lineare (a) und die konvergente (b) Netzstruktur

In der linearen Netzstruktur (Abb. 2 (a)) ist *B* abhängig von *A*, und *A* ist abhängig von *B*. Mathematisch ausgedrückt bedeutet dies:

$$P(A|B) \neq P(A) \qquad P(B|A) \neq P(B)$$

Da *A* und *B* voneinander abhängig sind, besteht, wenn sich die Wahrscheinlichkeit von *A* oder *B* ändert, ein direkter Wahrscheinlichkeitseinfluß zwischen *A* und *B*. Wahrscheinlichkeitseinflüsse in Pfeilrichtung, wie der von *A* nach *B*, sollen als *prognostische Wahrscheinlichkeitseinflüsse* und Wahrscheinlichkeitseinflüsse entgegen der Pfeilrichtung, wie der von *B* nach *A*, als *diagnostische Wahrscheinlichkeitseinflüsse* bezeichnet werden.
Ein prognostischer Wahrscheinlichkeitseinfluß ähnelt der Struktur plausiblen Schließens *'Untersuchung eines Beweisgrundes'* [Polya-1954]. Das Prinzip dieser Struktur und des Wahrscheinlichkeitseinflusses ist gleich: Der Beweis des Sachverhaltes *A* läßt eine Aussage über die Glaubwürdigkeit bzw. die Wahrscheinlichkeit eines anderen Sachverhaltes *B* zu, weil der Sachverhalt *A* den Sachverhalt *B* zu einem bestimmten Grad impliziert. Bei der Struktur plausiblen Schließens impliziert *A* sicher *B*. Bei dem prognostischen Wahrscheinlichkeitseinfluß wird der Grad, mit dem *A B* impliziert, durch die bedingte Wahrscheinlichkeit $P(B|A)$ angegeben. Dabei fällt oder sinkt die Wahrscheinlichkeit von *B*.
Ein diagnostischer Wahrscheinlichkeitseinfluß ähnelt der Struktur plausiblen Schließens *'Untersuchung einer Konsequenz'* [Polya-1954]. Bei beiden Schlußformen wird von den Folgen eines Sachverhaltes auf die Glaubwürdigkeit bzw. Wahrscheinlichkeit des Sachverhaltes selbst geschlossen. Der Unterschied liegt wieder darin, daß bei der plausiblen Struktur *B* sicher aus *A* folgt und bei dem diagnostischen Wahrscheinlichkeitseinfluß der Grad der Implikation durch die bedingte Wahrscheinlichkeit $P(B|A)$ angegeben wird. Weiterhin wird bei der Struktur plausiblen Schließens ausgesagt, daß die Glaubwürdigkeit von *A* steigt. Während bei dem Wahrscheinlichkeitseinfluß die Glaubwürdigkeit von *A* je nach dem Wert von $P(A|B)$ steigen oder sinken kann.

In konvergenten Netzstrukturen (Abb. 2 (b)) existiert ein anderer Einfluß als in der linearen Struktur. Es handelt sich um den Einfluß zwischen *A* und *C*. Es gelten folgende (Un-)Abhängigkeiten: *A* und *C* sind unabhängig, wenn *B* nicht evident ist, und *A* und *C* sind abhängig, wenn *B evident* ist. Ein Wahrscheinlichkeitseinfluß zwischen *A* und *C* besteht also nur unter der Voraussetzung, daß für *B* Evidenz vorliegt. Im Gegensatz zu den prognostischen und diagnostischen Wahrscheinlichkeitseinflüssen handelt es sich hier um einen indirekten Wahrscheinlichkeitseinfluß, denn er ist in der Netzstruktur nicht über eine direkte Kante zwischen *A* und *C* repräsentiert. Da Evidenz für *C* die Wahrscheinlichkeit von *A* beeinflussen kann, ohne die Wahrscheinlichkeit von *B* zu verändern, soll der Wahrscheinlichkeitseinfluß zwischen den Vorgängern in einer konvergenten Netzstruktur als elementarer Wahrscheinlichkeitseinfluß aufgefaßt werden. Er wird als *interkausaler Wahrscheinlichkeitseinfluß* bezeichnet.
Der interkausale Wahrscheinlichkeitseinfluß läßt sich mit der Struktur plausiblen Schließens *'Untersuchung einer Konkurrenzvermutung'* [Polya-1954] oder auch 'explaining away' [Pearl-1988] vergleichen. Das gemeinsame Prinzip lautet: Der Beweis des Sachverhaltes *C* läßt eine

Aussage über die Glaubwürdigkeit bzw. die Wahrscheinlichkeit des Sachverhaltes *A* zu, weil *A* und *C* einen gemeinsamen Sachverhalt *B* jeweils zu einem bestimmten Grad implizieren und weil *B* bekannt ist. In der Struktur plausiblen Schließens wird über den Grad der Implikation von *B* durch *A* bzw. *C*, nur ausgesagt, daß sowohl *A* als auch *C* eine Erklärung für *B* darstellen. Bei dem interkausalen Wahrscheinlichkeitseinfluß in einer konvergenten Netzstruktur wird der Grad, mit dem *A B* impliziert, durch die bedingte Wahrscheinlichkeit $P(B|A)$ und der Grad, mit dem *C B* impliziert, durch die bedingte Wahrscheinlichkeit $P(B|C)$ angegeben. Zusätzlich wird der Grad, mit dem *A* und *C* gemeinsam *B* implizieren, durch die bedingte Wahrscheinlichkeit $P(B|A,C)$ angegeben. Durch die Struktur plausiblen Schließens sinkt die Wahrscheinlichkeit von *A*, wenn *B* bereits bekannt ist und *C* nachgewiesen wird. Die resultierende Glaubwürdigkeit des interkausalen Wahrscheinlichkeitseinflusses wird durch die bedingte Wahrscheinlichkeit $P(A|B,C)$ angegeben, d. h., die Wahrscheinlichkeit von A kann sinken oder steigen.

In der folgenden Definition wird angegeben, was hier unter einem Wahrscheinlichkeitseinfluß zwischen zwei Variablen verstanden wird.

Definition 1 - Wahrscheinlichkeitseinfluß zwischen zwei Variablen in einem Bayes-Netz:
Y und *X* seien propositionale Variablen in einem Bayes-Netz. Es besteht ein Wahrscheinlichkeitseinfluß von *Y* nach *X*, wenn die Wahrscheinlichkeitsänderung von *Y* eine Neuberechung der Wahrscheinlichkeit von *X* bewirkt hat. Insofern kann ein Wahrscheinlichkeitseinfluß von *Y* nach *X* nur bestehen, wenn *Y* und *X* abhängig voneinander sind.

Aufgrund der Ähnlichkeiten zwischen probabilistischen Abhängigkeiten und den genannten Strukturen plausiblen Schließens, liefern die Wahrscheinlichkeitseinflüsse intuitive Antworten auf die Ausgangsfrage: „Warum ändert sich die Wahrscheinlichkeit von *X*, wenn für *Y* Evidenz eingegeben wird ?“:

- Antwort bei einem prognostischen Wahrscheinlichkeitseinfluß von *Y* nach *X*:
  „Der Verdacht auf *X* hat sich geändert, weil sich der Verdacht auf *Y* geändert hat. *X* ist abhängig von *Y*, denn *Y* ist eine Mögliche Erklärung für *X*.“
- Antwort bei einem diagnostischen Wahrscheinlichkeitseinfluß von *Y* nach *X*:
  „Der Verdacht auf *X* hat sich geändert, weil sich der Verdacht auf *Y* geändert hat. *X* ist abhängig von *Y*, denn *X* ist eine mögliche Erklärung für *Y*.“
- Antwort bei einem interkausalen Wahrscheinlichkeitseinfluß von *Y* nach *X* über *Z*:
  „Der Verdacht auf *X* hat sich geändert, weil sich der Verdacht auf *Y* geändert hat. *X* ist abhängig von *Y*, denn sowohl *Y* als auch *X* haben die Erklärung von *Z* zum Ziel, und es ist bekannt, daß *Z* vorliegt.“

Auf dieser Ebene kann der Benutzer die Auswirkungen der Evidenz sehr abstrakt nur auf Basis der Richtungen der Kanten aufzeigen lassen. In vielen Fällen reicht dies schon aus, und es besteht keine Notwendigkeit, detaillierter zu erklären. Diese Abstraktionsebene ist eine Erweiterung der Arbeiten von Druzdel und Henrion und Elsaesser.

2.2 Erklärungen auf der Abstraktionsebene der Vorzeichen von Wahrscheinlichkeitseinflüssen

Durch das Vorzeichen wird angegeben, ob die Wahrscheinlichkeit durch den Wahrscheinlichkeitseinfluß steigt, sinkt oder unverändert bleibt. In diesem Sinne wird zwischen positiven, negativen und neutralen Wahrscheinlichkeitseinflüssen unterschieden. In der folgenden Definition werden die Voraussetzungen angegeben, die erfüllt sein müssen, damit zwischen zwei Variablen ein positiver, negativer oder neutraler Wahrscheinlichkeitseinfluß besteht?

Definition 2 - Wahrscheinlichkeitseinfluß zwischen Ausprägungen von propositionalen Variablen unter Berücksichtigung des Kontextes:
Y und *X* seien propositionale Variablen. *Y* habe *m* und *X* habe *n* Ausprägungen, y_j beeinflußt x_i in dem Kontext *K* positiv, wenn gilt:

$$P(x_i|K) < P(x_i|y_j, K), \qquad (1)$$

wobei *K*, falls *Y* und *X* in einer konvergenten Struktur verbunden sind, Evidenz für die übrigen Vorgänger der konvergenten Variable repräsentiert.

Negative bzw. neutrale Wahrscheinlichkeitseinflüsse zwischen Ausprägungen werden analog definiert, indem in (1) '<' durch '>' bzw. '=' ersetzt wird.

Die Abstraktionsebene des Vorzeichens von Wahrscheinlichkeitseinflüssen ähnelt der Abstraktionsebene von Druzdzel und Henrion [Druzdzel, Henrion-1993(b)]. Druzdzel und Henrion definieren Wahrscheinlichkeitseinflüsse wie folgt:

Definition von Druzdzel und Henrion - qualitativer Wahrscheinlichkeitseinfluß:[1]
Wir sagen Y beeinflußt X positiv, geschrieben $S^+(Y, X)$, genau dann, wenn für alle Ausprägungen $y_i > y_j$ von Y und x_0 von X und K, gilt:

$$P(X \geq x_0 | y_i, K) \geq P(X \geq x_0 | y_j, K), \qquad (2)$$

wobei K die Menge aller Vorgänger von X bis auf Y ist.
Negative bzw. neutrale qualitative Wahrscheinlichkeitseinflüsse (geschrieben $S^-(Y, X)$ bzw. $S^0(Y, X)$) werden analog definiert, indem in (2) '≥' durch '≤' bzw. '=' ersetzt wird.

Diese Definition basiert auf einer Ordnung der Ausprägungen der Variablen. Ein Beispiel für eine Ordnung von Ausprägungen liefert die Variable A = 'Cholesterinspiegel im Blut' mit den Ausprägungen 'x<200 mg/dl', '200≤x<300 mg/dl' und 'x≥300 mg/dl'. Diese Ausprägungen geben die Menge des Cholesterins im Blut an und lassen sich dementsprechend ordnen:

a_1 = 'x≥300 mg/dl',
a_2 = '200≤x<300 mg/dl',
a_3 = 'x<200 mg/dl'.

Die Ausprägungen werden von der größten zur kleinsten Ausprägung absteigend geordnet. Es gilt also:

$$\forall i < j : [a_i \geq a_j].$$

Ein Nachteil dieser Definition ist, daß in der Praxis Wahrscheinlichkeitseinflüsse zwischen Variablen auftreten, die mit dieser Definition nicht erfaßt werden können. Hierzu gehören beispielsweise alle nicht monotonen Wahrscheinlichkeitseinflüsse.
Die Definition 2 und die von Druzdzel und Henrion unterscheiden sich in folgenden Punkten:

- Bei Druzdzel und Henrion bedeutet ein Wahrscheinlichkeitsanstieg von X, daß die Wahrscheinlichkeit von höheren Ausprägungen von X angestiegen ist. Lassen sich die Ausprägungen von X nicht ordnen, macht diese Aussage keinen Sinn. Deswegen wird in Definition 2 die *Wahrscheinlichkeitsänderung jeder Ausprägung* von X einzeln betrachtet.
- Y hat einen positiven Wahrscheinlichkeitseinfluß auf X bedeutet bei Druzdzel und Henrion, daß höhere Ausprägungen von Y höhere Übergangswahrscheinlichkeiten für höhere Ausprägungen von X (ab x_0) haben. Auch diese Aussage macht keinen Sinn, wenn sich keine Ordnung auf die Ausprägungen von Y und X festlegen läßt. Um in diesem Fall den Wahrscheinlichkeitseinfluß von Y zu bestimmen, werden in Definition 2 die *Wahrscheinlichkeitseinflüsse der einzelnen Ausprägungen* von Y betrachtet.
- In der Definition 2 wird ein Wahrscheinlichkeitseinfluß zwischen zwei Ausprägungen immer in Bezug auf einen bestimmten *Kontext* definiert. Durch den Kontext wird in einer konvergenten Struktur Evidenz für die anderen Vorgänger berücksichtigt. In Abb. 3. soll beispielsweise der Einfluß von A nach B betrachtet werden. Der Kontext beinhaltet hier die Evidenz für D. In bestimmten Kontexten kann ein Wahrscheinlichkeitseinfluß stärker bzw. schwächer als in anderen Kontexten sein (additive Synergie [Druzdzel, Henrion-1993(b)]). Weiterhin wird dadurch berücksichtigt, daß das Vorzeichen von Wahr-scheinlichkeitseinflüssen je nach Kontext unterschiedlich sein kann. Bei Druzdzel und Henrion muß das Vorzeichen eines Wahrscheinlichkeitseinflusses von Y nach X in jedem Kontext gleich sein.

[1] Die Definiton wurde bzgl. der Variablenbezeichnung angepaßt.

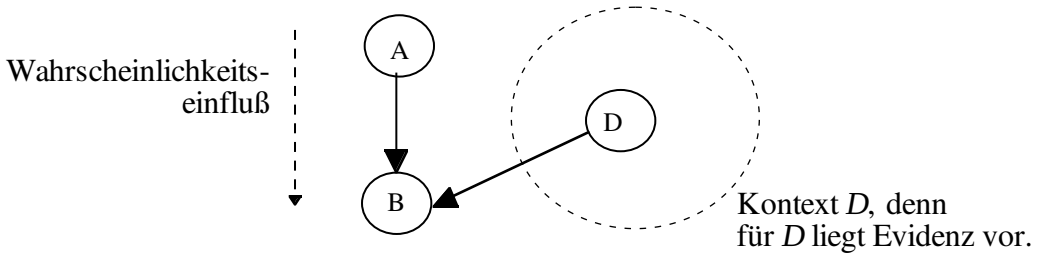


Abb. 3: Wahrscheinlichkeitseinfluß von *A* nach *B* in dem Kontext Evidenz für *D*.

Wie sieht nun die Antwort auf die Ausgangsfrage aus? Angenommen, die Wahrscheinlichkeit von x_i ist gestiegen. Die Frage lautet dann: „Warum steigt die Wahrscheinlichkeit von x_i?“ Zur Beantwortung der Frage wird zunächst der Wahrscheinlichkeitseinfluß von *Y* angegeben:

„Der Verdacht auf x_i steigt, weil *Y* einen positiven Einfluß auf x_i hat.“

Zur Erklärung des Wahrscheinlichkeitseinflusses von *Y* werden die Wahrscheinlichkeitseinflüsse der einzelnen Ausprägungen von *Y* als Argumente angegeben. Die Ausprägungen werden in zwei Mengen eingeteilt, Ausprägungen mit einem positiven und Ausprägungen mit einem negativen Wahrscheinlichkeitseinfluß auf x_i. Folgende Wahrscheinlichkeitseinflüsse werden als Argumente angeführt:
Bei einem positiven Wahrscheinlichkeitseinfluß von *Y* auf x_i:

- Positive Wahrscheinlichkeitseinflüsse von Ausprägungen, deren Wahrscheinlichkeit gestiegen ist.
- Negative Wahrscheinlichkeitseinflüsse von Ausprägungen, deren Wahrscheinlichkeit gesunken ist.

Bei einem negativen Wahrscheinlichkeitseinfluß von *Y* auf x_i:

- Negative Wahrscheinlichkeitseinflüsse von Ausprägungen, deren Wahrscheinlichkeit gestiegen ist.
- Positive Wahrscheinlichkeitseinflüsse von Ausprägungen, deren Wahrscheinlichkeit gesunken ist.

2.3 Erklärungen auf der Abstraktionsebene des Ausmaßes von Wahrscheinlichkeitseinflüssen

Durch das Ausmaß wird angegeben, wie hoch die Wahrscheinlichkeit durch den Wahrscheinlichkeitseinfluß steigt oder sinkt. Es wird nun gezeigt, daß das Ausmaß eines Wahrscheinlichkeitseinflusses von bestimmten Parametern abhängig ist, die den Prämissen in bestimmten Strukturen plausiblen Schließens sehr ähnlich sind.

Prognostische Wahrscheinlichkeitseinflüsse

Y sei der direkte Vorgänger von *X*. Der Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf x_i im Kontext *K* wird durch die bedingte Wahrscheinlichkeit $P(x_i|y_j,K)$ angegeben. Wenn *Y* der einzige direkte Vorgänger von *X* ist, gilt immer $K = \emptyset$. In diesem Fall ist die bedingte Wahrscheinlichkeit $P(x_i|y_j)$ als Übergangswahrscheinlichkeit in der Wahrscheinlichkeitsmatrix von *X* gespeichert und muß nicht berechnet werden. Ansonsten kann $P(x_i|y_j,K)$ mit Hilfe des Satzes von der totalen Wahrscheinlichkeit berechnet werden. $P(x_i|y_j,K)$ ist der einzige Parameter, von dem das Ausmaß des Wahrscheinlichkeitseinflusses von y_j auf x_i abhängt. Unter Berücksichtigung des Kontextes gelten somit folgende Zusammenhänge:

- Der *positive* prognostische Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i im Kontext *K* ist um so *höher*, je *wahrscheinlicher* es ist, daß x_i im Kontext *K* aus y_j folgt.
- Der *negative* prognostische Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i im Kontext *K* ist um so *höher*, je *unwahrscheinlicher* es ist, daß x_i im Kontext *K* aus y_j folgt.

Diagnostische Wahrscheinlichkeitseinflüsse

Als nächstes sei angenommen, Y sei direkter Nachfolger von X und für Y liege Evidenz vor. Der diagnostische Wahrscheinlichkeitseinfluß wurde weiter oben mit der Struktur plausiblen Schließens 'Untersuchung einer Konsequenz' verglichen. Y wird dabei als Konsequenz von X interpretiert. Polya hat eine Erweiterung dieser Struktur vorgestellt: 'Untersuchung einer Konsequenz mit Zusatzbedingung' [Polya-1954 Seite 21]:

Aus X folgt Y
Y an sich sehr unwahrscheinlich
Y wahr

X sehr viel glaubwürdiger

In der Struktur plausiblen Schließens werden nur boolesche Variablen betrachtet. Aus 'X folgt Y' bedeutet: $P(y^+|x^+) = 1$. Die zweite Prämisse der Struktur 'Y an sich sehr unwahrscheinlich' soll als Bedingung für die Wahrscheinlichkeit von $P(y^+)$ interpretiert werden. 'Y an sich sehr unwahrscheinlich' bedeutet dann, daß $P(y^+)$ einen sehr kleinen Wert hat. Die Konklusion wird als Aussage über die Wahrscheinlichkeit, die X durch den Wahrscheinlichkeitseinfluß von Y erhält, interpretiert. 'X sehr viel glaubwürdiger' bedeutet dann, daß $P(x^+|y^+)$ einen hohen Wert hat.

Bei der Bayes-Netzstruktur kann es sich bei Y und X um multiple Variablen handeln. Die Wahrscheinlichkeit, mit der y_j aus x_i folgt, wird durch $P(y_j|x_i)$ angegeben, 'y_j an sich sehr unwahrscheinlich' wird durch einen kleinen Wert für $P(y_j)$ angegeben. Die Wahrscheinlichkeit, die x_i nach der Verifikation von y_j erhält, ist gegeben durch $P(x_i|\ y_j)$. Diese Interpretation der Struktur plausiblen Schließens bildet die Grundlage der folgenden Überlegungen.

Der diagnostische Wahrscheinlichkeitseinfluß von y_j auf x_i im Kontext K wird durch die bedingte Wahrscheinlichkeit $P(x_i\ |y_j,K)$ repräsentiert. Es soll untersucht werden, ob sich bei der Berechnung dieser bedingten Wahrscheinlichkeit ein ähnlicher Zusammenhang wie in der plausiblen Struktur 'Untersuchung einer Konsequenz' herausstellen läßt. Zu diesem Zweck wird die $P(x_i\ |y_j,K)$ nach der Definition bedingter Wahrscheinlichkeiten folgendermaßen berechnet:

$$P(x_i|y_j,K) = \frac{P(y_j|x_i,K)P(x_i)}{P(y_j|K)} \quad . \tag{3}$$

Wenn X der einzige direkte Vorgänger von Y ist, dann ist die Wahrscheinlichkeit $P(x_i\ |y_j,K)$ als Übergangswahrscheinlichkeit in der Wahrscheinlichkeitsmatrix von Y repräsentiert.

In der Struktur 'Untersuchung einer Konsequenz mit Zusatzbedingung' gilt (siehe [Polya-1954] Seite 21]):

> Die Verifizierung einer Konsequenz zählt mehr oder weniger, je nachdem, ob die Konsequenz an sich mehr oder weniger unwahrscheinlich ist.

Mit der eingangs eingeführten Interpretation der Prämissen und der Konklusion der Struktur plausiblen Schließens gilt bei der Berechnung des Wahrscheinlichkeitseinflusses von y_j auf x_i ein analoger Zusammenhang, der sich aus der Formel (3) ergibt:

> Der Wahrscheinlichkeitseinfluß von y_j auf x_i ist um so größer, je geringer die Wahrscheinlichkeit von y_j im Kontext K ist, und je wahrscheinlicher es ist, daß y_j im Kontext K aus x_i folgt.

Es gelten also folgende Zusammenhänge:

- Der *positive* diagnostische Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i im Kontext K ist um so höher, je *wahrscheinlicher* es ist, daß y_j im Kontext K aus x_i folgt, und je *unwahrscheinlicher* y_j in dem Kontext K ist. Weiterhin ist der Wahrscheinlichkeitseinfluß um so höher, je *wahrscheinlicher* x_i vor dem Wahrscheinlichkeitseinfluß von y_j bereits war.
- Der *negative* diagnostische Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i im Kontext K ist um so *höher*, je *unwahrscheinlicher* es ist, daß y_j im Kontext K aus x_i folgt, und je *wahrscheinlicher* y_j in dem Kontext K ist. Weiterhin ist der Wahrscheinlichkeitseinfluß um so höher, je *unwahrscheinlicher* x_i vor dem Wahrscheinlichkeitseinfluß von y_j bereits war.

Interkausale Wahrscheinlichkeitseinflüsse

Y und *X* seien direkte Vorgänger von *Z* (konvergente Struktur). Weiter oben wurde geschildert, daß der interkausale Wahrscheinlichkeitseinfluß der Struktur plausiblen Schließens 'Untersuchung einer Konkurrenzvermutung' sehr ähnlich ist, wenn *Y* und *X* als Konkurrenzvermutungen interpretiert werden. Sowohl *Y* als auch *X* sind in diesem Fall Vermutungen, die der Erklärung des Sachverhaltes *Z* dienen. Die Struktur plausiblen Schließens hat folgende Gestalt:

Y starker Konkurrent von *X*
Y wahr

X viel weniger glaubwürdig

Das Ausmaß der Konklusion hängt in dieser Struktur davon ab, ob es sich bei *Y* um einen starken bzw. schwachen Konkurrenten oder um einen starken bzw. schwachen Verbündeten von *X* handelt. Durch die Analyse der Berechnung eines interkausalen Wahrscheinlichkeitseinflusses wird im folgenden eine probabilistische Interpretation der Prämisse '*Y* starker Konkurrent von *X*' ermöglicht.

Ein interkausaler Wahrscheinlichkeitseinfluß von y_j über z_k auf x_i im Kontext *K* wird durch die bedingte Wahrscheinlichkeit $P(x_i|y_j,z_k,K)$ repräsentiert. Diese Wahrscheinlichkeit läßt sich nach der Definition von bedingten Wahrscheinlichkeiten wie folgt berechnen:

$$P(x_i|y_j,z_k,K) = \frac{P(z_k|x_i,y_j,K)P(x_i)}{P(z_k|y_j,K)}. \qquad (4)$$

In der Struktur plausiblen Schließens 'Untersuchung einer Konkurrenzvermutung' gilt (siehe [Polya-1954] Seite 52, 53):

> Die Verifizierung einer konkurrierenden Vermutung zählt mehr oder weniger, je nachdem, ob es sich bei der Konkurrenzvermutung um einen mehr oder weniger starken Konkurrenten handelt.

Das bedeutet, daß die Glaubwürdigkeit von *X* um so stärker sinkt, je stärker der Konkurrent *Y* ist, und sie sinkt um so weniger, je schwächer der Konkurrent *Y* ist.

Aufgrund der Interpretation der Formel (4) soll die Stärke des Konkurrenten/Verbündeten *Y* von *X* bzgl. *Z* für die einzelnen Ausprägungen von *Y*, *X* und *Z* durch zwei probabilistische Parameter bestimmt werden: Durch die Übergangswahrscheinlichkeit $P(z_k \mid x_i, y_j)$ und durch die bedingte Wahrscheinlichkeit $P(z_k|y_j)$. Somit gilt:

> Die Ausprägung y_j ist ein um so stärkerer Konkurrent von x_i bzgl. z_k, je wahrscheinlicher es ist, daß z_k aus y_j allein folgt, und je unwahrscheinlicher es ist, daß z_k aus y_j und x_i gemeinsam folgt.
>
> Die Ausprägung y_j ist ein um so schwächerer Konkurrent von x_i bzgl. z_k, je unwahrscheinlicher es ist, daß z_k aus y_j allein folgt und je wahrscheinlicher es ist, daß z_k aus y_j und x_i gemeinsam folgt.
>
> Analog ist der Zusammenhang für den Fall, daß y_j und x_i Verbündete sind.

Bei dieser Interpretation gelten in der Formel (4) die gleichen Zusammenhänge wie in der Struktur plausiblen Schließens 'Untersuchung einer Konkurrenzvermutung'.

Unter Berücksichtigung des Kontextes *K* lassen sich also folgende Zusammenhänge festhalten:

- Der *positive* interkausale Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i über eine Ausprägung z_k im Kontext *K* ist um so *höher*, je *wahrscheinlicher* es ist, daß z_k im Kontext *K* aus y_j und x_i gemeinsam folgt, und je *unwahrscheinlicher* es ist, daß z_k im Kontext *K* aus y_j allein folgt. Weiterhin ist der positive Wahrscheinlichkeitseinfluß um so höher, je *wahrscheinlicher* x_i vor dem Wahrscheinlichkeitseinfluß von y_j bereits war.
- Der *negative* interkausale Wahrscheinlichkeitseinfluß einer Ausprägung y_j auf eine Ausprägung x_i über eine Ausprägung z_k im Kontext *K* ist um so *höher*, je *unwahrscheinlicher* es ist, daß z_k im Kontext *K* aus y_j und x_i gemeinsam folgt, und je *wahrscheinlicher* es ist, daß z_k im Kontext *K* aus y_j allein folgt. Weiterhin ist der negative Wahrscheinlichkeitseinfluß um so höher, je *unwahrscheinlicher* x_i vor dem Wahrscheinlichkeitseinfluß von y_j bereits war.

Die Interpretation des diagnostischen Einflusses ist der von Elsaesser ähnlich. Die Interpretation des interkausalen Einflusses ist bei Elsaesser nicht vorgenommen worden. Interkausale Einflüsse sind aber eine wichtige Eigenschaft von Bayes-Netzen.

3. Beispiel einer Erklärung

Um ein Beispiel einer abstraktionsbasierten Erklärung zu zeigen, soll das einfache Bayes-Netz in Abb. 4 betrachtet werden.

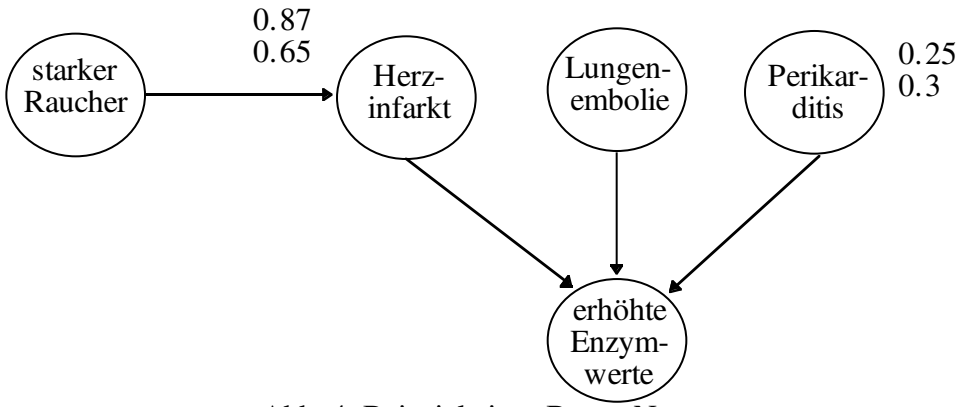


Abb. 4: Beispiel eines Bayes-Netzes

Es sei angenommen, der Patient hat erhöhte Blutenzymwerte. Nun wird bekannt, daß er starker Raucher ist. Die Wahrscheinlichkeiten für "Herzinfarkt" und "Perikarditis" vor und nach dieser Beobachtung sind in Abb. 4 angegeben. Die Frage lautet: "Warum ändert sich die Wahrscheinlichkeit, daß der Patient eine Perikarditis[2] hat?"
Die Wahrscheinlichkeitsberechnungen werden auf Grundlage des 'message-passing'-Algorithmus von Pearl [Neapolitan-1990] generiert. In diesem Algorithmus wird die Wahrscheinlichkeitsänderung von z. B. "Perikarditis" durch zwei probabilistische Schlußfolgerungen berechnet: Von "starker Raucher" nach "Herzinfarkt" und von "Herzinfarkt" nach "Perikarditis". Der Benutzer kann sich nun beide Schlußfolgerungsschritte auf allen drei Abstraktionsebenen ansehen. Bis zu welcher Ebene ein Schritt erklärt wird, bleibt dem Benutzer überlassen. Im folgenden wird der entsprechende Erklärungsdialog angegeben. Dabei wird angenommen, daß der Benutzer den ersten Schritt nur auf der ersten Ebene und den zweiten Schritt auf allen drei Ebenen erklärt haben möchte. In dem Erklärungstext wird die Höhe von Wahrscheinlichkeiten durch linguistische Terme angegeben. Zur Übersetzung von numerischen Wahrscheinlichkeiten in linguistische Terme wurde eine Intervallzuordnung von Lichtenstein und Newmann [Lichtenstein, Newman-1967] angewendet. Für MEDIKUS ist weiterführend geplant, eine probabilistische Semantik von linguistischen Termen zu ermitteln, um auf Basis dieser Semantik eine entsprechende Übersetzung vornehmen zu können [Schröder, Möbus, Thole-1996].

> „*Der Verdacht auf "Herzinfarkt ändert sich, da bekannt "starker Raucher" bekannt wurde. "Herzinfarkt" und "starker Raucher" sind abhängig, da "Herzinfarkt" eine Folge von "starker Raucher" sein kann.*
> *Der Verdacht auf "Perikarditis" hat sich geändert, weil sich der Verdacht auf "Herzinfarkt" geändert hat. "Herzinfarkt" und "Perikarditis" sind abhängig, da beide die Erklärung von "erhöhten Enzymwerten" zum Ziel haben und "erhöhte Enzymwerte" bekannt sind.*
> *Der Verdacht auf "Perikarditis" ist gesunken, weil "Herzinfarkt" eine zu "Perikarditis" alternative Erklärung für "erhöhte Enzymwerte" ist.* "

P(Perikarditis | erhöhte Enzymwerte,Herzinfarkt)=0.22 < 0.3= P(Perikarditis | erhöhte Enzymwerte). $K = \varnothing$, da für "Lungenembolie" keine Evidenz vorliegt,

> „*Für die Höhe der negativen Verdachtsänderung gilt allgemien: Der negative Einfluß einer sicher beobachteten Erkrankung auf den Verdacht, daß eine weitere Erkrankung vorliegt, ist um so höher, je unwahrscheinlicher es ist, daß beide Erkrankungen gemeinsam zu dem gemeinsamen Symptom führen und je wahrscheinlicher es ist, daß die erste Erkrankung ohne die weitere Erkrankung zu dem Symptom führt. Generell ist der Verdacht auf eine Erkrankung nach einem negativen Einfluß immer noch um so höher, je*

[2] Herzbeutelentzündung

stärker der Verdacht vor dem Einfluß bereits war. Im vorliegenden Fall gilt: Es ist ziemlich wahrscheinlich, daß "Herzinfarkt" und "Perikarditis" gemeinsam eine Erklärung für "erhöhten Enzymwerten" liefern, weiterhin ist es sehr wahrscheinlich, daß "Herzinfarkt" auch ohne "Perikarditis" zu "erhöhte Enzymwerte" führt. "Perikarditis" war vor dem negativen Einfluß durch "Herzinfarkt" wahrscheinlich."

P(erhöhte Enzymwerte | Herzinfarkt) = 0.76, P(erhöhte Enzymwerte | Herzinfarkt, Perikarditis) = 0.78

4. Zusammenfassung und Erweiterungen

Es wurde ein abstraktionsbasierter Ansatz zur Erklärung von Wahrscheinlichkeitsänderungen vorgestellt. Dieser Ansatz integriert und erweitert bereits existierende Ansätze und wurde bereits zwecks Evaluation in das System MEDIKUS, einem Modellierungs- und Diagnosesystem auf Basis von Bayes-Netzen, integriert. Die Evaluation soll zusammen mit Medizinern durchgeführt werden. Es werden bestimmte Diagnosen durchgespielt, wobei die Mediziner gebeten werden, Wahrscheinlichkeitsänderungen zu begründen. Diese Begründungen werden mit denen des Erklärungsansatzes verglichen.

In dem Erklärungsansatz werden Wahrscheinlichkeitsänderungen über Schlußfolgerungsschritte erklärt. Es ist sinnvoll, für die Erklärung von Wahrscheinlichkeitsänderungen weitere Erklärungsarten in die Erklärungskomponente aufzunehmen. Dadurch erhält der Benutzer die Möglichkeit, aus den angebotenen Erklärungsarten seine bevorzugte auszuwählen. Beispielsweise könnten fallbasierte Erklärungen [Schank-1986] von Wahrscheinlichkeitsänderungen entwickelt werden. Der Wahrscheinlichkeitsanstieg einer Erkrankung, der durch die Eingabe von Evidenz für ein bestimmtes Symptom bewirkt wurde, könnte dabei gerechtfertigt werden, indem in der Erklärung typische Fälle genannt werden, in denen die Erkrankung zu dem Symptom geführt hat. Diese Art der Erklärung entspricht der Struktur plausiblen Schließens 'Folgerung aufgrund von Analogie', die in [Polya-1954] beschrieben wird.

Weiterhin können Szenario-Erklärungen nach dem Muster von Druzdzel und Henrion [Druzdzel, Henrion-1990] implementiert werden. In Szenario-Erklärungen werden Wahrscheinlichkeiten boolescher Variablen gerechtfertigt, indem zwei Gruppen von Szenarien aufgezählt werden: Szenarien, in denen die Variable wahr ist, und Szenarien, in denen die Variable nicht gilt. Durch Abwägen der Wahrscheinlichkeiten der beiden Szenario-Gruppen, ergibt sich die Wahrscheinlichkeit der Variable.

5. Literaturverzeichnis

Druzdzel, M. J. und Henrion, M. 1990. Using scenarios to explain probabilistic inference, in *Working notes of the AAAI-90 Workshop on Explanation*, Boston, MA: American Association for Artificial Intelligence, S. 133-141.

Druzdzel, M. J. und Henrion, M. 1993(a). Efficient reasoning in qualitative probabilistic networks, in *Proceedings of the Eleventh National Conference on Artificial Intelligence* (AAAI-93), Washington, D.C., S. 548-553.

Druzdzel, M. J. und Henrion, M. 1993(b). Belief Propagation in Qualitative Probabilistic Networks, in N. Pipra Carrete, M.G. Singh (eds*), Qualitative Reasoning and Decision Technologies*, CIMNE, Barcelona, S. 451-460.

Druzdzel, M. J. und Henrion, M. 1993(c). Intercausal reasoning with uninstantiated ancestor nodes, in *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence* (UAI-93), Washington, D.C., S.317-325.

Druzdzel, M. J. und Van der Gaag, L. C. 1995. Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information, in *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence* (UAI-9), Montreal, Quebec, Canada, S. 141-148.

Elsaesser, C. 1989. Explanation of probabilistic inference, in L.N. Kanal, T.S. Levitt, and J.F. Lemmer (eds), *Uncertainty in Artificial Intelligence 3*, North Holland: Elsevier Science Publishers B.V, S. 387-400.

Folckers, J., Möbus, C., Schröder, O. und Thole, H. J. 1996(a). An Intelligent Problem Solving Environment for Designing Explanation Models and for Diagnostic Reasoning in Probabilistic Domains, in C. Frasson, G. Gauthier, A. Lesgold (eds), *Intelligent Tutoring Systems, Proceedings of the Third International Conference* (ITS), Montreal, Berlin: Springer (LNCS 1086).

Folckers, J., Möbus, C., Schröder, O. und Thole, H. J. 1996(b). *Supporting the Construction of Explanation Models and Diagnostic Reasoning in Probabilistic Domains*, paper accepted for the Second International Conference on the Learning Sciences (ICLS 96), (http://lls.informatik.uni-oldenburg.de/Project/Abteilung/Publikationen.html).

Henrion, M. 1989. Some Practical Issues in Constructing Belief Networks, in L.N. Kanal, T.S. Levitt, and J.F. Lemmer, (eds), *Uncertainty in Artificial Intelligence 3*, North Holland: Elsevier Science Publishers B.V, S. 161-173.

Kahnemann, D., Slovic, P. und Tversky, A., (eds) 1982. *Judgement under Uncertainty: Heuristics and Biases*, Cambridge: Cambridge University Press.

Lichtenstein, S. und Newman, J. R. 1967. Empirical scaling of common verbal phrases associated with numerical probabilities, in *Psychon. Sci.*, Vol. 9 (10).

Neapolitan, R.E. 1990. *Probabilistic reasoning in Expert Systems*, New York: Wiley.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo: Morgan Kaufman (second ed.).

Polya, G. 1954. *Mathematics and Plausible Reasoning, Vol II: Patterns of Plausible Inference*, Princeton, New Jersey: Princeton University Press.

Schank, R. C. 1986. *Explanation Patterns: Understanding Mechanically and Creatively*, Hillsdale, New Jersey: Erlbaum Publishers.

Schröder, O., Möbus, C. und Thole, H. J. 1996. *Acquiring Knowledge from Linguistic Models in Complex Probabilistic Domains*, paper accepted für the European Conference on AI in Education (EuroAIED-96), Lissabon, Portugal (http://lls.informatik.uni-oldenburg.de/Project/Abteilung/Publikationen.html).

Die Integration von dynamischen und statischen Analysemethoden in einer intelligenten Lern- und Problemlöseumgebung

Janine Willms, Hermann Göhler, Claus Möbus

Fachbereich Informatik, Abteilung Lehr-und Lernsysteme,
C.v.O. Universität Oldenburg, D-26111 Oldenburg, und
OFFIS Institut, D-26121 Oldenburg,
E-Mail: {Janine.Willms,Hermann.Goehler,Claus.Moebus}@informatik.uni-oldenburg.de

Der Hypothesentest-Ansatz ist ein zentraler Aspekt der ISP-DL-Theorie . Er ermöglicht individuelles Feedback auf Basis des Lösungsentwurfes des Lernenden, ohne ein explizites Schülermodell aufbauen zu müssen. Die Umsetzung dieser kognitionswissenschaftlichen Theorie in reale Problemlöseumgebungen (IPSE = Intelligent Problem Solving Environment) wurde bereits früher an Beispielen unterschiedlichster Domänen gezeigt (Möbus 1995, Möbus 1996). Dabei wurde bisher die Prüfung der dynamischen Aspekte eines Systems in den Vordergrund gestellt, da diese eine vollständige Methode zum Testen beliebiger Schülerlösungen bietet. Ein System, das auf dynamischem Model-Checking beruht, kann dadurch zu jedem Designentwurf eines Schülers korrektes Feedback geben. Allerdings bietet das dynamische Model-Checking sehr fein granulierte Basis-Informationen, die in frühen Entwicklungsphasen des Modells, d.h. Abwäge- und Planungsphasen, nicht sinnvoll genutzt werden können. Eine zweite, auf der Statik eines Entwurfs beruhende, konzeptbasierte Analyse soll helfen, den Lernenden in diesen frühen Entwicklungsphasen zu unterstützen. In diesem Beitrag wird PULSE vorgestellt, eine IPSE zur Umsetzung pneumatischer Schaltpläne aus vorgegebenen Spezifikationen, in der beide Analysemethoden realisiert wurden, so daß die Vorteile beider Methoden nutzbar sind.

1 Statische and Dynamische Analyse

Es gibt verschiedene Arten, einen Lernenden durch ein computergestütztes System zu unterstützen. Unabhängig davon, ob diese nun intelligentes Tutorsystem, explorative Lernumgebung oder intelligente Designumgebung genannt werden, ist die Analyse des Entwurfs eines Schülers essentiell, um dessen Ziele zu erkennen und Fehler zu beseitigen. Systeme wie SPADE (Miller & Goldstein, 1976) oder PROUST (Johnson, 1986) verwenden eine statische, planbasierte Analyse mit korrekten Plänen und Fehlerplänen, um den Entwurf des Lernenden zu untersuchen und zu bewerten. Korrektur- und Ergänzungsvorschläge werden auf Basis der dem System bekannten Pläne generiert. Andere Systeme, wie z.B. das System PETRI-HELP (Möbus, Pitschke, Schröder, 1992), bewerten den dynamischen Aspekt eines Entwurfes. Nur selten wird die Kombination von dynamischer und statische Analyse zur Bewertung von Schülerlösungen herangezogen. In dem intelligenten Tutorsystem SYPROS (Henning, 1994; Herzog, 1996) wird der statische Ansatz momentan durch eine dynamische Analysekomponente erweitert. Aus unseren Erfahrungen mit PETRI-HELP wissen wir die Vorzüge einer dynamischen Analyse, nämlich deren Vollständigkeit und Korrektheit, zu schätzen. Vollständigkeit bedeutet hierbei, daß der gesamte Problemraum durch die Analysemethode abgedeckt wird. Korrektheit bezieht sich auf das Ergebnis der Analyse. Das dynamische Model-Checking liefert immer korrektes Feedback. Für frühe Problemlösephasen, wie z.B. der Abwäge- oder Planungsphase sind die Rückmeldungen einer dynamischen Analyse jedoch nur eingeschränkt geeignet. Teilentwürfe können oft nicht auf Basis einer dynamischen Analyse getestet werden. Auch sind gerade in frühen Entwurfsphasen höhere Konzepte notwendig, um einen generellen Entwurfsablauf zu planen. Die fein granulierten Rückmeldungen einer dynamischen Analyse sind in dieser Planungsphase auf einer zu niedrigen Entwurfsebene. Das System PULSE vereint beide Analyse-Methoden. Es handelt sich bei PULSE um eine IPSE (Möbus, 1996), die auf der ISP-DL-Theorie ("Impasse-Success-Problem-Solving-Driven-Learning", Möbus, 1995, 1996), einer kognitiven Theorie des Wissenserwerbs beruht.

2 Die ISP-DL-Theorie

Die ISP-DL-Theorie verbindet verschiedene Formen menschlicher Wissensakquisition zu einer zusammenhängenden kognitiven Theorie. Dabei wird von einem induktiven Erwerb neuen Wissens nach Überwindung einer Stocksituation (Impasse) und einer deduktiven Optimierung bisher bereits vorhandenen Wissens nach der erfolgreichen Lösung eines Problems ausgegangen. Der Problemlöseprozeß umfaßt vier verschiedene Phasen: die Abwäge-, Planungs-, Aktions- und Bewertungsphase. Um intelligente Problemlöseumgebungen oder intelligente Designumgebungen zu realisieren, die effektiv 'Learning by doing' und die Akquisition neuen Wissens *während des Problemlösens* unterstützen, wurden aus der ISP-DL-Theorie Design-Kriterien für solche Systeme abgeleitet:

- Das System sollte einen Lernenden niemals unterbrechen. Neues Wissen wird nur akzeptiert, wenn sich der Schüler in einer selbstverschuldeten Stocksituation befindet. Eine Unterbrechung des Problemlöseprozesses durch das System würde störend wirken.
- Der Lernende soll sein Vorwissen in den Problemlöseprozeß so weit als möglich einbringen können. Das System muß daher in der Lage sein, die Aktionen des Schülers zu interpretieren und zielgerichtete Hilfen und Erklärungen anzubieten. Diese sollten auf den bisherigen Entwurf des Lernenden abgestimmt sein.

- Der Problemlöseprozeß des Lernenden sollte nicht durch das System eingeschränkt werden. Aktives Lernen, uneingeschränktes Arbeiten und exploratives Erforschen der Domäne sollte dem Lernenden ermöglicht werden.
- Das System sollte auf Anfrage passende Informationen und Hilfestellungen in Form von Korrektur- und Ergänzungsvorschlägen liefern können. Diese sollten **in jeder Phase** des Problemlöseprozesses sinnvoll nutzbar sein.

Insbesondere die letzte Anforderung wird erst durch die Verbindung von dynamischer und statischer Analysemechanismen realisiert.

3 Die intelligente Problemlöseumgebung PULSE

PULSE (pneumatic learning and simulation environment) ist eine wissensbasierte Design- und Problemlöseumgebung, welche die Konstruktion pneumatischer Schaltungen unterstützt. PULSE wurde in Zusammenarbeit mit der DIHT-Gesellschaft für Berufliche Bildung, Organisation zur Förderung der IHK-Weiterbildung mbH (Deutscher Industrie- und Handelstag in Bonn) sowie den regionalen Industrie- und Handelskammern entwickelt. Die Problemlöseumgebung PULSE soll in der Ausbildung zum Industriemeister "Metall" eingesetzt werden. Den Auszubildenden steht eine Sequenz von 30 Aufgaben der Prüfungs-, Aufgaben- und Lernmittelstelle (PAL) zur Verfügung. Sie können in PULSE mit einem CAD-Editor pneumatische Schaltungen frei konstruieren. In jeder Phase des Entwerfens ist es ihnen dabei möglich, ihren Lösungsentwurf als Hypothese an das System zu formulieren. Das System untersucht diese Hypothese und gibt dem Auszubildenden differenzierte Fehlerrückmeldungen und Erklärungen.

PULSE beinhaltet die folgenden Komponenten:

- Eine *Sequenz von Pneumatik- Aufgaben* mit aufsteigendem Schwierigkeitsgrad. Aufgaben sind durch textuelle Beschreibungen und Funktionsdiagramme spezifiziert. Ein Funktionsdiagramm beschreibt dabei nur das Verhalten von elementaren Aktoren (z.B. Zylinder), Stellgliedern und Schaltern oder Tastern (z.B. manuell betätigte Ventile).
- Ein *CAD-Editor* erlaubt die Konstruktion und Simulation von Pneumatikschaltungen. Wichtige CAD-Funktionen wie Verschieben, Ausrichten, Gummibandeffekt usw. stehen dabei als Werkzeuge zur Verfügung.
- Lösungsentwürfe werden durch einen *dynamischen Modelchecker* geprüft. Er zeigt an, an welcher Stelle ein Lösungsentwurf das im Funktionsdiagramm dargestellte gewünschte Verhalten einer Pneumatikschaltung verletzt und wo korrektes Verhalten vorliegt.
- *Konzeptbasierte Hilfen und Heuristiken* unterstützen den Entwurfsprozeß in frühen Problemlösephasen.
- Eine *Simulationsumgebung* ermöglicht die Visualisierung des dynamischen Schaltungsverhaltens des realisierten Entwurfes.
- Ein *Dozentenmodus* ermöglicht die interaktive Erstellung neuer Aufgaben und die Einbindung in das System durch einen Dozenten.

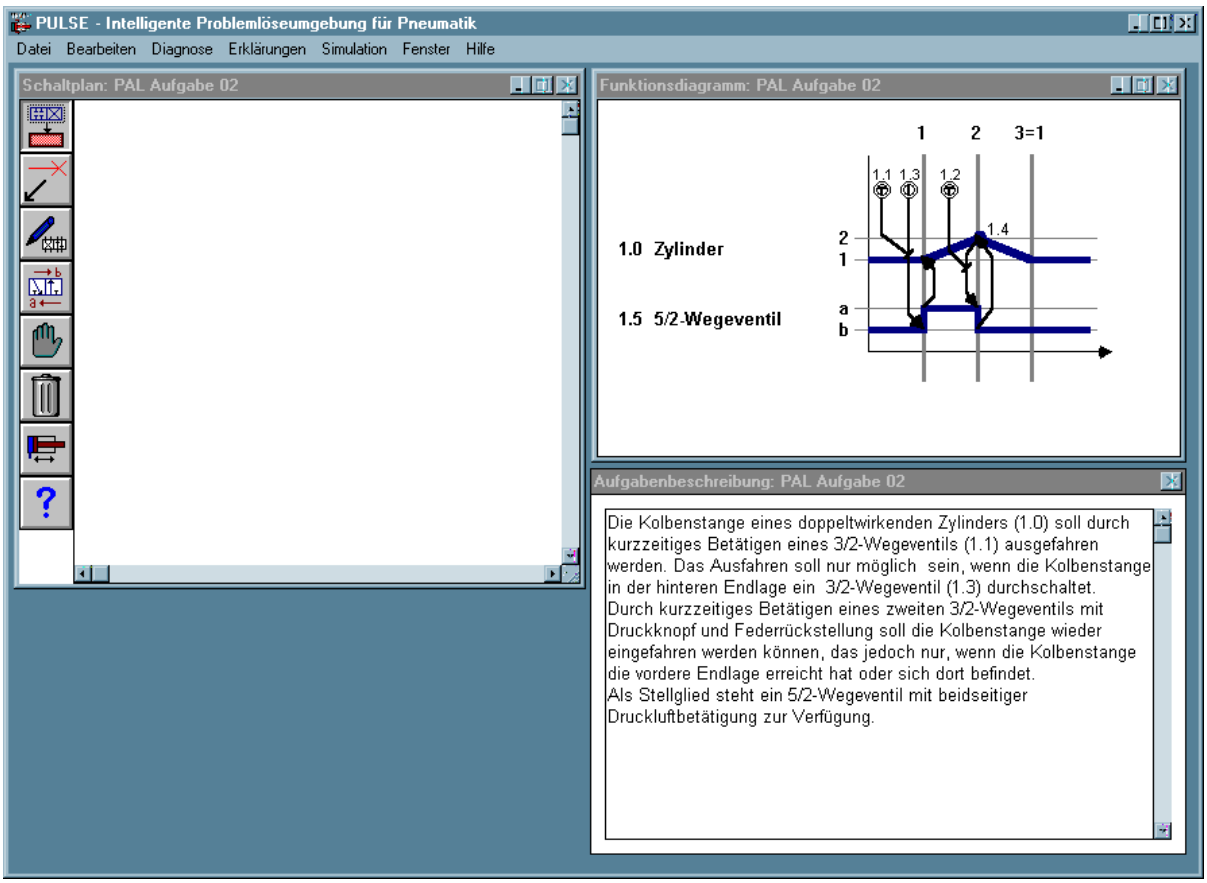


Abb. 1: Aufgabe PAL 02

3.1 Der Hypothesentest-Ansatz: Model-Checking und statische Analyse

Nach Auswahl einer Aufgabe präsentiert sich dem Lernenden die Aufgabenstellung in Form einer textuellen Beschreibung sowie eines Funktionsdiagramms. Ein leerer Editor steht zur Konstruktion des Lösungsentwurfes zur Verfügung (siehe Abb. 1). Pneumatische Bauglieder nach DIN-ISO 1219 können in beliebiger Weise auf der Entwurfsfläche plaziert und durch Leitungen miteinander verbunden werden. Durch Benennung der einzelnen Bauglieder mit den Bezeichnungen aus der Aufgabenstellung wird der Entwurf für das System überprüfbar.

Wenn der Lernende sich bereits in der Domäne auskennt, kann er, ohne durch das System unterbrochen zu werden, eine Lösung generieren und diese im Anschluß durch den in PULSE integrierten, dynamischen Model-Checker (Clarke, Emerson & Sistla, 1986; Damm, Döhmen, Gerstner, Josko, 1990; Josko 1990) verifizieren lassen. Eventuelle Vorkenntnisse des Lernenden werden optimal unterstützt. Die Dynamik eines Schaltplans kann *jederzeit* durch den dynamischen Model-Checker analysiert werden. Dafür markiert der Lernende im Funktionsdiagramm eine Testhypothese, von der er annimmt, daß sie bereits korrekt realisiert wurde. Er hat die Möglichkeit nur einzelne Bauteile oder bestimmte Taktschritte, zum Beispiel bis zu einer bestimmten Schaltungsstellung, prüfen zu lassen. Die Korrektheit der Hypothese wird durch Erzeugung eines Fallgraphen überprüft, der sämtliche von einem Initialzustand erreichbaren Zustände der pneumatischen Steuerung und deren Übergange enthält. Aus der Testhypothese können temporallogische Formeln abgeleitet werden, die anhand des Fallgraphen auf Gültigkeit überprüft werden können. Fehler im Verhalten der Schaltung werden durch entsprechende Markierung im Funktionsdiagramm visualisiert.

Diese Analysemethode hat den Vorteil, zu jedem Entwurf und jeder Testhypothese eine korrekte Antwort generieren zu können. Daher kann sie von einem erfahrenen Lerner genutzt werden, um den Lösungsraum einer Aufgabenstellung explorativ zu erforschen, d.h. verschiedene Lösungsalternativen aufzubauen und schnelle Antworten über die Korrektheit seiner Entwurfsideen zu erhalten. Der dynamische Model-Checker bezieht sich bei seiner Rückmeldung jedoch immer nur auf die dynamischen Aspekte des Funktionsdiagramms. Bedingungen, die zum Beispiel aus dem Text der Aufgabenstellung hervorgehen, z.B. daß ein bestimmtes Ventil zur Lösung der Aufgabe genutzt werden soll, können nicht überprüft werden. Es muß dafür auf die statische Analyse zurückgegriffen werden.

Die Rückmeldungen basieren immer auf dem Funktionsdiagramm, daher sind die Informationen sehr fein-granuliert und beziehen sich immer auf einzelne Bauteile und Taktschritte. Die Erklärung von Fehlern ist auf Grundlage des dynamischen Model-Checkers entsprechend schwierig. Der Lernende muß sich selbst erklären, wie Abweichungen von dem erwarteten Verhalten entstanden sind. Ihm können keine abstrakteren Erklärungen auf einer höheren Konzeptebene angeboten werden, wie sie von Lehrern oder Lehrbüchern genutzt werden.

Es ist anzunehmen, daß ein unsicherer Lerner jedoch bereits in früheren Entwurfsphasen Feedback durch das System benötigt. So kann bereits das Lesen der Aufgabenstellung Stocksituationen auslösen, beispielsweise durch unbekannte Konzepte (3/2-Wegeventil, Stellglied), oder fehlende oder unvollständige Pläne zur Umsetzung der Aufgabenstellung.

Grundlegende Begriffsdefinitionen, auf die sich die Aufgabenstellung und die konzeptbasierten Erklärungen beziehen, werden deshalb durch PULSE *just in time* zur Verfügung gestellt. Es werden *Objektkonzepte* und *Funktionskonzepte* unterschieden. Bei Objektkonzepten handelt es sich um die Bauteile der Domäne, die sich in einer Hierarchie (siehe Abb. 2) organisieren lassen. Durch Verbindung der Bauteile mit Leitungen im Editor werden Funktionen realisiert. Auch können einzelne Bauteile innerhalb der Schaltung bestimmte Rollen übernehmen. Ein Ventil übernimmt beispielsweise die Rolle eines *Stellglieds*, wenn ein Zylinder direkt und nur durch dieses Ventil angesteuert wird. Diese Funktionskonzepte bilden ebenfalls eine Hierarchie, an dessen Spitze das Funktionskonzept der *Indirekten Steuerung* steht. Diese Hierarchien können direkt von dem Lernenden eingesehen werden und durch Anklicken des gewünschten Bauteils oder der gewünschten Funktion werden ihm detailliertere Informationen, teilweise durch Grafiken und Animationen unterstützt, präsentiert.

Der Lernende hat darüberhinaus die Möglichkeit, *aufgabenspezifische Erklärungen* anzufordern, die beispielsweise beinhalten, welche Objekte und Funktionen zur Lösung der speziellen Aufgabe zu realisieren sind. Diese Erklärungen sind nur von der Aufgabenstellung abhängig. Die dafür benötigten Daten werden beim Starten der Aufgabe durch PULSE aus der Aufgabenstellung mittels Heuristiken extrahiert. Daher ist PULSE in der Lage, konzeptbasierte Erklärungen zur Aufgabenstellung auch zu begründen. Ein eventuell bereits vorgenommener Lösungsansatz des Lernenden in Form eines Schaltplans wird dabei noch nicht berücksichtigt.

PULSE stellt weiterhin Erklärungen zur Verfügung, die sich auf die *Umsetzung der Aufgabenstellung* in einen Schaltplan beziehen. Dabei handelt es sich um domänenspezifische Regeln und Heuristiken, die dem Lernenden auf Anfrage zur Verfügung gestellt werden.

Zur Lösung einer Aufgabenstellung reichen aufgabenspezifische Erklärungen allein aber im Allgemeinen nicht aus. Normalerweise baut der Lernende einen eventuell unvollständigen Schaltplan auf und möchte diesen durch das System bewerten lassen. Eventuell fehlt ihm Wissen, wie er mit der Lösung der Aufgabe fortfahren soll. Er benötigt also zum einen

schaltplanbezogene Erklärungen, für die sein Entwurf analysiert und bewertet werden muß, und Erklärungen zur Korrektur bzw. Ergänzung seines Entwurfes.

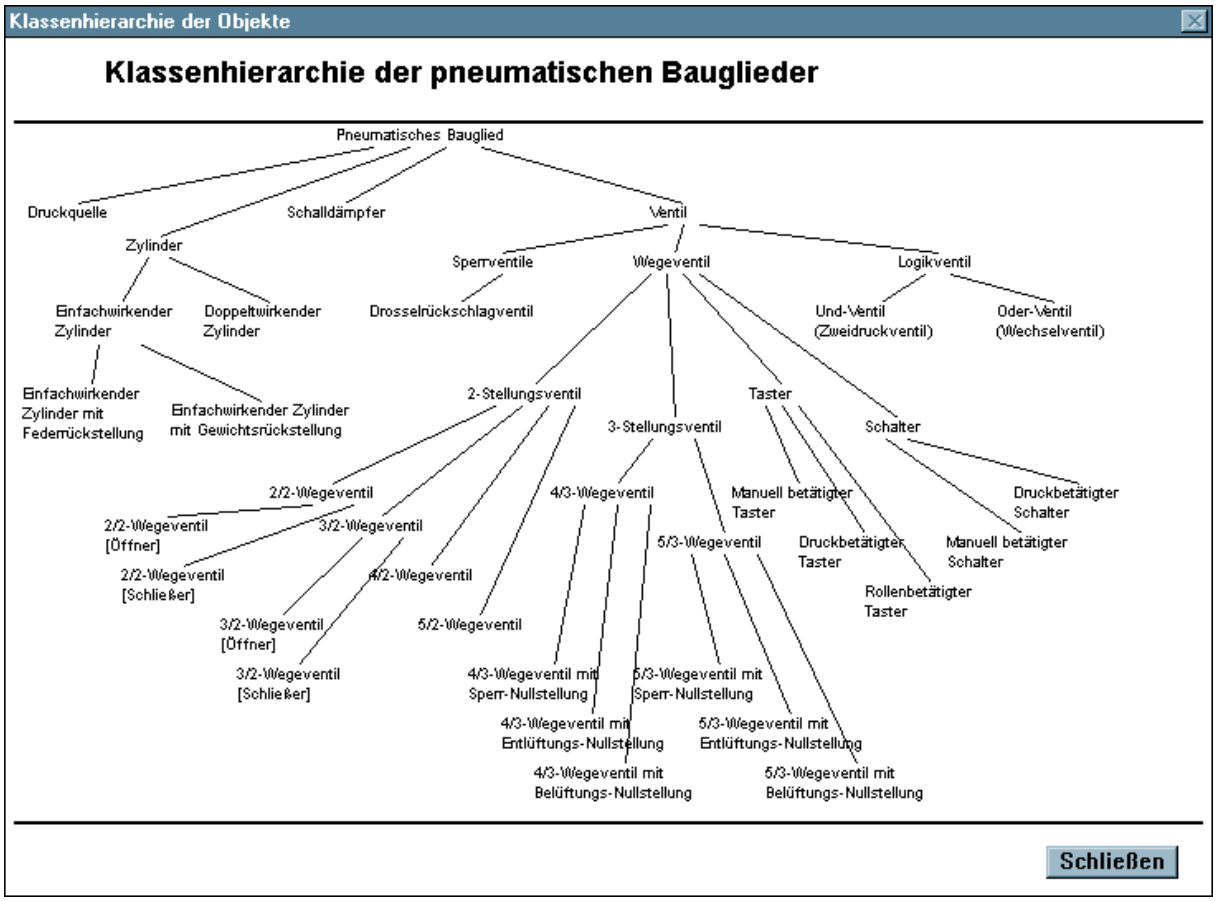


Abb. 2: Die Hierarchie der pneumatischen Bauteile

Bei der statischen Analyse des Schaltplans handelt es sich um eine planbasierte Analyse des Schaltplans mit allen Vor- und Nachteilen, die bereits in anderen Projekten identifiziert wurden. Die planbasierte Analyse ist eine unvollständige Analysemethode, da eventuell nicht der gesamte Problemraum durch die statische Analyse abgedeckt wird. Der Schüler kann möglicherweise Schaltungen entwerfen, die PULSE aufgrund der beschränkten Menge integrierter Pläne nicht identifizieren kann. Dadurch kann es zu Fehlinterpretationen durch das System und sogar zu fehlerhaften Rückmeldungen kommen. Im Gegensatz zum dynamischen Model-Checking ist die statische Analyse also nicht inhärent vollständig und korrekt. Die Qualität der statischen Analyse ist abhängig von der Vielfalt der dem System bekannten Pläne. Zu jedem Konzept der Aufgabenstellung muß PULSE mindestens einen Plan kennen, wie dieses Konzept auch im Schaltplan umgesetzt werden kann. Durch die Existenz verschiedener Pläne zur Realisierung eines Konzeptes wächst die Anzahl der erkannten korrekten Lösungen. Die statische Analyse ist erfolgreich, wenn sehr viele korrekte Pläne integriert sind. Es muß jedoch immer davon ausgegangen werden, daß nicht alle korrekten Pläne identifiziert wurden bzw. werden können. Auch steigt mit der Anzahl der Pläne natürlich auch die Dauer der Analyse, so daß zwischen Zeitbedarf und 'Erfolgsquote' abgewogen werden muß.

Es ist sinnvoll, sich den dynamischen Model-Checker zur Identifikation noch nicht bekannter Pläne nutzbar zu machen. Kann die statische Analyse einen Entwurf nicht identifizieren, weil beispielsweise ein komplexerer Plan fehlt, so wäre es mit Hilfe des dynamischen Model-

Checkers möglich, zu erkennen, ob sich der Schaltungsentwurf dennoch korrekt verhält. Ist das der Fall, so kann davon ausgegangen werden, daß ein vermeintlich fehlendes Konzept durch eine Ersatzschaltung realisiert wurde. Durch Analyse der relevanten Druckluftströmungen, die aus den Daten der Fallgrapherzeugung des dynamischen Model-Checkings entnommen werden können, kann ein Modul isoliert werden, das dasselbe Ein-/Ausgabeverhalten aufweist, wie das fehlende Konzept. Dieses Ersatzmodul kann dann als neue Realisationsmöglichkeit des Konzeptes in das System integriert werden.

Schaltplanbezogene Erklärungen können sich ebenfalls auf Objekt- und Funktionskonzepte beziehen. Der Entwurf des Lernenden wird zunächst unabhängig von der Aufgabenstellung analysiert. Dabei werden die Objekte untersucht und ihre spezifischen Eigenschaften festgehalten. Diese Daten werden dem Lernenden auf Wunsch zur Verfügung gestellt. Auch wird festgehalten, wie die Objekte zu Funktionen verknüpft worden sind.

Der Vergleich der Konzepte des Entwurfs mit denen der Spezifikation deckt Fehler im Entwurf auf. So können fehlerhafte, fehlende oder zusätzliche Objekte und Funktionen identifiziert und dem Lernenden als schaltplanbezogene Informationen zur Verfügung gestellt werden.

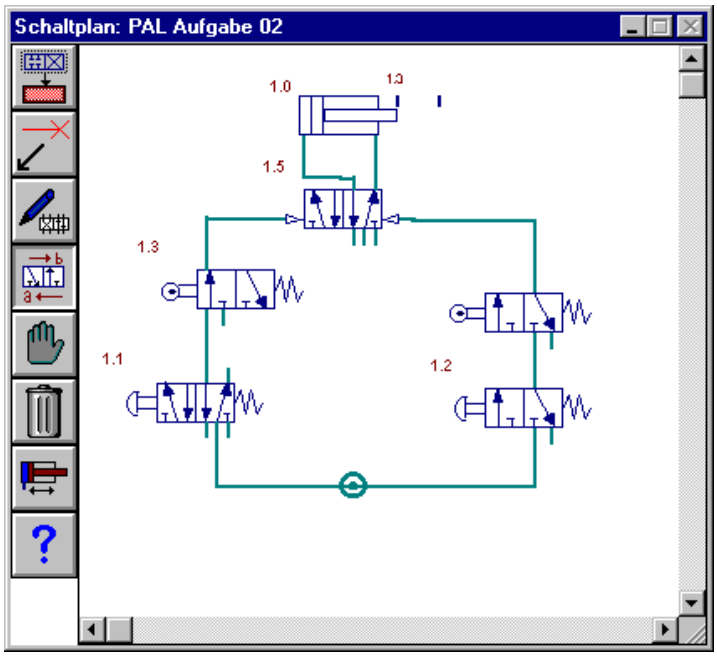


Abb. 3: Fehlerhafter Entwurf eines Lernenden

In Abb. 3 wird der Entwurf eines Lernenden zur Lösung von Aufgabe 2 (siehe Abb. 1) dargestellt. Der Aufruf der Diagnose bewirkt zunächst einen Test auf Basis der statischen Analyse, um fehlerhafte und unbenannte Objekte zu identifizieren und auf inkorrekte Ausgangsstellungen der Bauglieder hinzuweisen. Eine Meldung gibt dem Lernenden entsprechende Hinweise (siehe Abb. 4), damit er sich eventuelle Fehler in der Dynamik besser erklären kann. Genauere Details werden an dieser Stelle nicht zur Verfügung gestellt. Der Lernende kann nun den Test seines Entwurfs abbrechen und selbständig die Fehler identifizieren und korrigieren, oder sich durch die Erklärungskomponente gezielt Hinweise geben lassen. Fährt er mit dem Test seiner Lösung fort, so erhält er die in Abb. 5 dargestellte Rückmeldung auf Basis des Funktionsdiagramms.

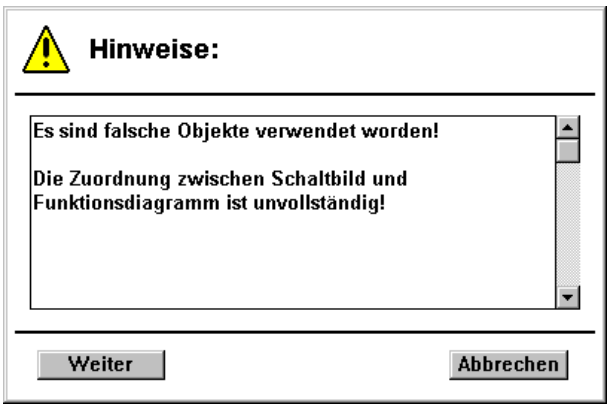


Abb. 4: Erste Rückmeldungen

Er steht nun vor der Aufgabe, die dynamischen Effekte seines Schaltungsentwurfes zu interpretieren und Fehler in der Dynamik durch Fehler in der Statik zu erklären. Diese Aufgabe ist für Novizen häufig nicht allein durchführbar. Ohne Hilfe durch das System befände sich der Lernende in einer Stocksituation, die er eventuell nicht selbständig überbrücken kann. Es ist zu vermuten, daß dies seine Motivation, mit dem System zu lernen, stark negativ beeinflussen würde. Die Erklärungskomponente auf Basis statischer Daten kann nun den Lernenden in seinem Problemlöseprozeß unterstützen. Er kann entweder die Daten der Analyse seines Schaltplanes betrachten, wobei ihm fehlerhafte, fehlende und unbenannte Bauteile und Funktionen genannt werden. Dabei würde ihm an dieser Stelle überlassen bleiben, wie er die konkreten Probleme lösen kann. Oder aber, wenn er sich nicht in der Lage fühlt, in seinem Entwurf fortzufahren, kann er sich Tips von der Erklärungskomponente geben lassen, die ihm auf Wunsch detailliert erläutert, was zu tun ist, und wie ein Entwurf zu einer korrekten Lösung transformiert werden kann.

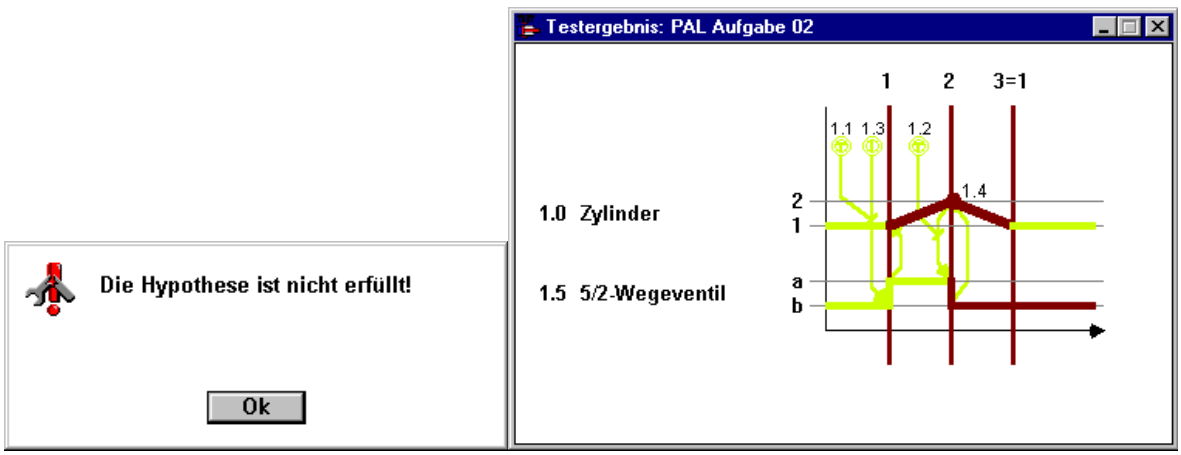


Abb. 5: Feedback des dynamischen Model-Checkers

Für *Korrektur- und Ergänzungsvorschläge* durch das System mußte eine Synthesekomponente implementiert werden, die ausgehend von einem Entwurf eine korrekte Lösung generiert, die möglichst wenige Änderungen an dem bestehenden Entwurf notwendig macht. Der Lernende wird auf diese Weise nicht mit einer Musterlösung des Systems konfrontiert, die nichts mit seinem Entwurf gemein hat, sondern erkennt korrekte Teile seines Entwurfs wieder und wie sie in eine korrekte Lösung integriert werden können. Die Hinweise werden zunächst auf abstrakter

Ebene gegeben, d.h. es wird zum Beispiel zuerst die Reparatur des obersten Konzeptes gefordert, in dem ein Fehler vorliegt. Reicht diese Information nicht aus, so kann durch Nachfragen die Position des Fehlers immer mehr eingeschränkt werden. Auch bei Ergänzungvorschlägen wird nach dieser *Methode der minimalen Information* vorgegangen. Sie soll bewirken, daß der Lernende sich selbst Gedanken über die Lösung macht und Fehler möglichst selbst entdecken und korrigieren kann. Es wird angenommen, daß Informationen, die sich der Lernende selbst erarbeitet hat, qualitativ hochwertiger sind, d.h., daß der Lernende sich länger daran erinnern kann und sie häufiger anwenden wird. PULSE ist ebenfalls in der Lage auf Wusch auch konkrete Hinweise zu geben, die sich direkt auf das Ändern oder Hinzufügen einzelner Bauteile oder Leitungen beziehen.

In dem gegebenen Beispiel sind mehrere Fehler kombiniert aufgetreten, die dem Lernenden erklärt werden müssen. a) Bei dem gemeldeten falschen Objekt handelt es sich um das Ventil 1.1, das laut Aufgabenstellung ein 3/2-Wegeventil sein soll. Im Entwurf wurde statt dessen ein 5/2-Wegeventil gewählt, das durchaus das korrekte Verhalten aufweisen kann und daher im Feedback der dynamischen Analyse auch als korrekt dargestellt wurde.

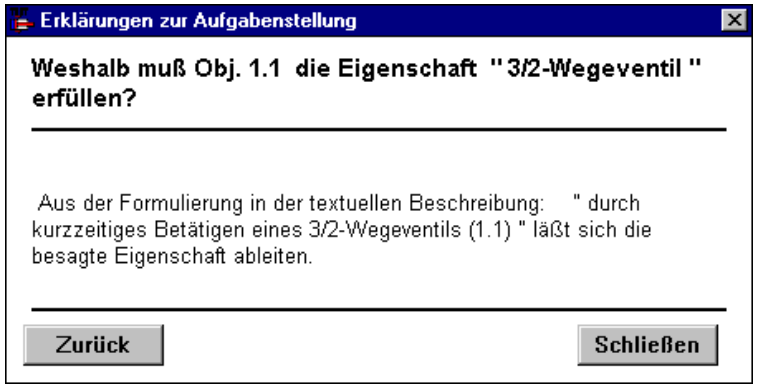


Abb. 6: Aufgabenspezifische Erklärung zu einem Bauteil

b) Eine fehlende Zuordnung bewirkt, daß die Dynamik des unbenannten Ventils durch den Model-Checker nicht geprüft werden kann. Das entsprechende Ventil 1.4 ist daher in der Rückmeldung als falsch markiert, ob wohl es, allerdings unbenannt, bereits korrekt realisiert wurde. c) Die Ursache dafür, daß sich der Zylinder nicht korrekt verhält, liegt in einer fehlenden Leitung beim Druckanschluß. Da Ventil 1.5 nicht an die Druckquelle angeschlossen ist, kann der Druck nicht zum Zylinder weitergegeben werden. Dies zeigt deutlich ein Manko der dynamischen Analyse. Obwohl der Schaltplan bereits korrekt aufgebaut sein kann, liefert sie zwar korrekte, aber unpassende Rückmeldungen, wenn die Schaltung nicht ablauffähig ist, zum Beispiel, wenn die notwendigen Anschlüsse an die Druckquelle fehlen. Dadurch ist der dynamische Model-Checker für das Testen von Teilentwürfen nur bedingt geeignet.

4 Zusammenfassung

Der Hypothesentest-Ansatz ist eine Methode, einem Lernenden individuelles Feedback zu liefern, ohne die Probleme des Aufbaus eines Schülermodells (Korrektheit, Adäquatheit, Datenschutz, etc.) in Kauf nehmen zu müssen. Die statische Analyse ist eine wichtige Informationsquelle, um dem Lernenden auf Basis abstrakterer Informationen und höherer Konzepte Fehler und weitere Vorgehensweisen zu erläutern. Sie sind daher insbesondere in den

frühen Phasen eines Entwurfsprozesses, den Abwäge- und den Planungsphasen, nützlich. Solche Erklärungen stehen auf Basis des dynamischen Model-Checkings nicht zur Verfügung. Fehlende Pläne zur Realisierung von Konzepten können jedoch fehlerhafte Erklärungen generieren. Das dynamische Model-Checking bietet dafür eine vollständige und korrekte Methode zur Analyse beliebiger Schaltpläne. Eine Kombination beider Analysemethoden bietet dem Lernenden sowohl die fein-granulierten Erklärungen des dynamischen Model-Checkers als auch die höheren Konzepte, wie sie in früheren Entwurfsphasen benötigt werden. Fehlende Pläne der statischen Analysemethode können mit Hilfe des dynamischen Model-Checkers ergänzt werden. Durch Integration beider Analysemethoden in dem System PULSE hoffen wir, die Vorteile der Methoden zu nutzen und die Nachteile ausgleichen zu können.

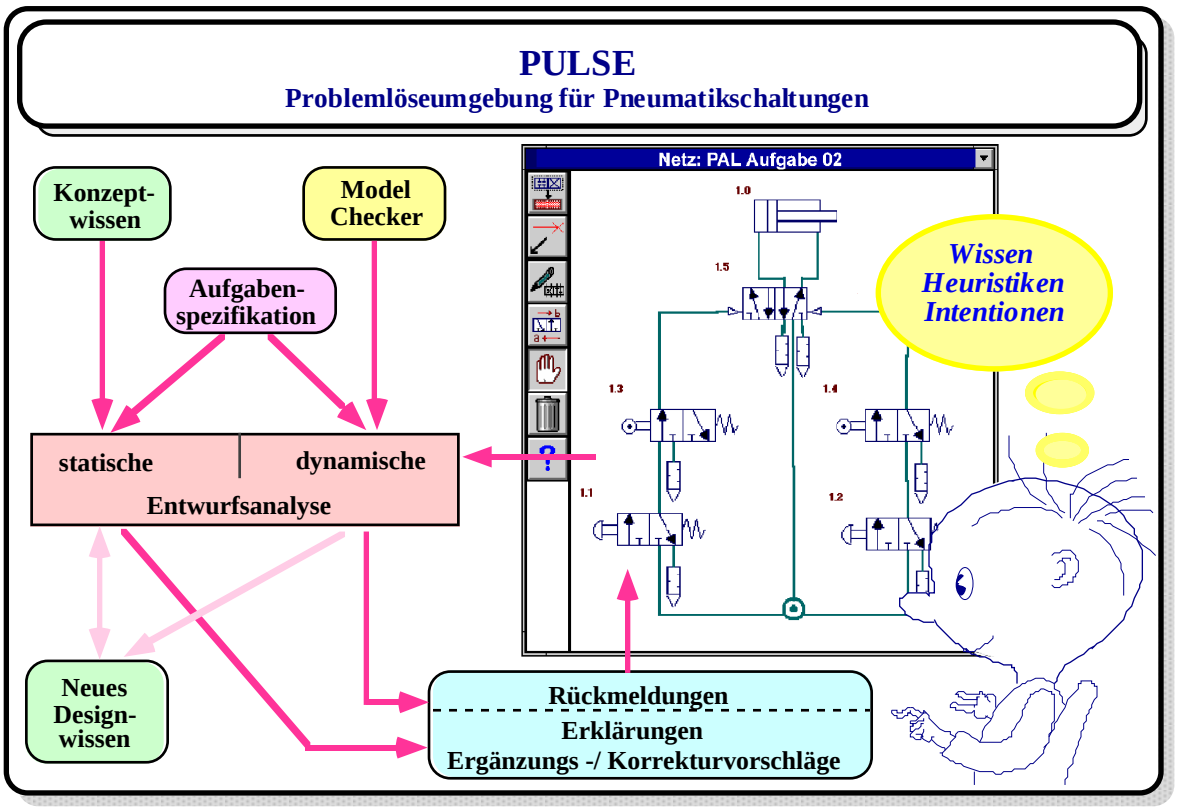


Abb. 7: Die IPSE PULSE im Überblick

Literatur

Clarke, E.M., Emerson, F.A. & Sistla, A.P. (1986). Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications. ACM Transactions on Programming Languages and Systems, Vol. 8, No. 2, 244 - 263

Damm, W., Döhmen, G., Gerstner, V., Josko, B. (1990). Modular Verification of Petri Nets. The Temporal Logic Approach, in: J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.), Proceedings REX-Workshop on stepwise refinement of distributed systems: models, formalisms, correctness. Berlin: Springer, LNCS 430.

Henning, U. (1994). Analyse eines Modells paralleler Prozesse für die Anwendung in einem intelligenten Lehrsystem, Dissertation, TU München, Institut für Informatik.

Herzog, C. (1996). Syntaxorientierte vs. ablauforientierte Diagnose in intelligenten Programmierumgebungen als Beispiel für den Einsatz konkurrierender Problemlöser. in: M. Thielscher, S.-E. Bornscheuer (eds.), Fortschritte der Künstlichen Intelligenz (KI 96), Dresden: Dresden University Press.

Johnson, W.L. (1986). Intention-based Diagnosis of Novice Programming Errors. Los Altos: Morgan Kaufmann.

Josko, B. (1990). Veryfying the correctness of AADL modules using model checking, in: J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.), Proceedings REX-Workshop on stepwise refinement of distributed systems: models, formalisms, correctness. Berlin: Springer, LNCS 430, 386-400.

Miller, M.L., Goldstein, I.P. (1976). SPADE: a grammar bases editor for planning and debugging programs. AI Lab Memo 386. Massachusetts Institute of Technology, Cambridge, Massachusetts.

Möbus, C., Pitschke, K., Schröder, O. (1992). Towards the Theory-Guided Design of Help Systems for Programming and Modelling Tasks, in C. Frasson, G. Gauthier, G.I. McCalla (eds), Intelligent Tutoring Systems, Proceedings of the Second International Conference ITS 92, Montreal, Berlin: Springer, LNCS 608, 294 - 301

Möbus, C. (1995). Towards an Epistemology of Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in: J. Greer (ed.), Artificial Intelligence in Education, Proceedings of AI-ED 95, Washington, D.C., August 16-19, 1995, Charlottesville: AACE, 138-145

Möbus, C., (1996). Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in: Proceedings of EuroAIED 96, Lisbon, Portugal, Sept. 30 - Oct. 2, 1996

ILSE: Ein Interaktives Lehr-/Lernsystem für Spätertaubte

Claudia Wans

Gerhard-Mercator-Universität – GH Duisburg
Fachbereich 11, Fachgebiet Informatik
Lotharstr. 65, D-47048 Duisburg
wans@informatik.uni-duisburg.de

Zusammenfassung

In diesem Beitrag stellen wir unser Forschungsvorhaben ILSE vor, in dem es um die Entwicklung, Implementation und Evaluation eines interaktiven Lehr-/Lernsystems für das Training der Sprechfertigkeit Spätertaubter geht. Nach Beschreibung der Zielgruppe nennen wir die grundlegenden Inhalte der Rehabilitation Spätertaubter. Dabei erläutern wir die Auswirkungen der Ertaubung auf die Sprechfertigkeit und geben einen Überblick über die bereits vorhandenen Computersysteme für die Rehabilitation Hörgeschädigter. Vor diesem Hintergrund zeigen wir, daß es zum gegenwärtigen Zeitpunkt kein Computerprogramm für das Sprechtraining gibt, das den Bedürfnissen Spätertaubter gerecht wird. Davon ausgehend beschreiben wir den Aufbau und die Lerninhalte des Systems ILSE, die geplante Form der Evaluation und die verschiedenen Einsatzmöglichkeiten von ILSE. Abschließend geben wir einen Ausblick auf weiterführende Aspekte, die unmittelbar mit unserem Forschungsvorhaben in Zusammenhang stehen.

1 Einleitung

Ziel des Forschungsvorhabens ist es, mit Hilfe moderner Multimediatechnik ein interaktives Lehr-/Lernsystem für Spätertaubte zu entwickeln, um ein selbständiges, eigenverantwortliches Training der Sprechfertigkeit zu ermöglichen. Zunächst erläutern wir einige der für unser Forschungsvorhaben wichtigen Begriffe.

In [23] werden Hörschädigungen folgendermaßen definiert: „Gehörlosigkeit liegt vor, wenn lautsprachliche Informationen der Umwelt nicht über das Gehör aufgenommen werden können. Schwerhörigkeit liegt vor, wenn trotz apparativer Versorgung lautsprachliche Informationen der Umwelt nur begrenzt aufgenommen werden können und wenn erhebliche Beeinträchtigungen in der Entwicklung des Sprechens und der Sprache oder im kommunikativen Verhalten oder im Lernverhalten auftreten oder wenn eine erhebliche Störung der zentralen Verarbeitung der Höreindrücke besteht."

Für die sprachliche Entwicklung ist es von großer Bedeutung, wann die Hörschädigung aufgetreten ist. So unterscheiden wir in Abhängigkeit davon, ob und wie weit der Betreffende beim Eintritt der Ertaubung über die Lautsprache verfügen kann, zwischen Gehörlosen und Spätertaubten. In Anlehnung an [25] verwenden wir folgende Definitionen:

- Als gehörlos bezeichnen wir eine Person, die
  - von Geburt an gehörlos oder
  - prälingual, d.h. vor dem Lautspracherwerb, ertaubt oder
  - perilingual, d.h. während des Lautspracherwerbs, ertaubt ist.
- Dagegen bezeichnen wir eine Person als spätertaubt, wenn sie
  - postlingual, d.h. nach dem Lautspracherwerb, ertaubt ist.
- Von Alterstaubheit sprechen wir, wenn die Person
  - erst im späteren Leben ertaubt ist.

Zielgruppe unseres Forschungsvorhabens sind Spätertaubte, d.h. Personen, die bis zu ihrem Kindes-, Jugend- oder frühen Erwachsenenalter normalhörend waren und im Besitz der Lautsprache das Gehör verloren haben.

Die häufigste Ursache einer solchen Ertaubung nach dem Lautspracherwerb ist Meningitis. Aber auch andere Krankheiten wie Hörsturz und Neurofibromatose, ototoxische Medikamente und Unfälle (Schädel-Hirn-Trauma) können den vollständigen Verlust des Gehörs zur Folge haben.

In Deutschland ertauben etwa 3.000 Personen pro Jahr; insgesamt gibt es hier rund 150.000 Spätertaubte.

2 Grundlagen der Rehabilitation Spätertaubter

In der Rehabilitation Gehörloser geht es in erster Linie um den Sprachaufbau, für den im Laufe der Zeit sehr verschiedenartige Methoden entwickelt wurden (vgl. [19, 42]). Da Spätertaubte im Besitz der vollen Lautsprache sind, jedoch diese nicht mehr auditiv kontrollieren können, geht es in deren Rehabilitation nicht um den Sprachaufbau, sondern um den Erhalt der Sprechfertigkeit [16]. Spätertaubte haben die Lautsprache als Normalhörende erlernt und verfügen nach der Ertaubung über diese in Wort und Schrift. Im Laufe der Zeit kann sich jedoch das Sprechen zunehmend verschlechtern, da die Rückmeldung über das Gehör fehlt.

Die Verschlechterung der lautsprachlichen Kommunikationsfähigkeit hat große Auswirkungen auf den sozialen Bereich. Werden Hörende mit der Sprache von Ertaubten konfrontiert, so schätzen sie in Abhängigkeit von der Sprechfertigkeit soziales Milieu, Alter, Bildung und Intelligenz in der Regel niedrig ein. In [11] wird belegt, daß Hörende die spätertaubten Sprecher für dumm oder sogar geistig behindert, ungebildet, unorganisiert, unzuverlässig und faul halten. Keiner der Hörenden erkennt in dieser Studie, daß die Sprecher ertaubt sind. Da die Kommunikation mit Ertaubten anstrengend ist, werden Gespräche von Seiten der Hörenden oft möglichst einfach und kurz gehalten oder ganz vermieden. So kommen aber auch keine sozialen Kontakte zustande, durch welche die Vorurteile aus dem Weg geräumt werden könnten. Durch ein gezieltes Sprechtraining kann ein Spätertaubter seine lautsprachbezogene Kommunikationsfähigkeit erheblich verbessern.

Um an der lautsprachlichen Kommunikation mit der Umwelt weiterhin teilnehmen zu können, ist der Spätertaubte darauf angewiesen, das Absehen zu lernen. Da Mundbilder jedoch stets individuell sind und nicht normiert werden können, werden wir keine Absehübungen in unser System einbinden, zumal dieser Bereich auch durch spezielle Absehkurse für Spätertaubte abgedeckt wird [43].

Auch die Beschäftigung mit der Körpersprache und manuellen Zeichensystemen wie dem internationalen Fingeralphabet und den lautsprachbegleitenden Gebärden (LBG) wird bereits an den Volkshochschulen und speziellen Gebärdensprachschulen angeboten. Zur Deutschen Gebärdensprache (DGS) wird zusätzlich zu dem weit verbreiteten Lexikon [27] und den zugehörigen Videokassetten vom Zentrum für deutsche Gebärdensprache und Kommunikation Gehörloser der Universität Hamburg ein Computerbasiskurs mit bewegten Gebärdenzeichen entwickelt.

Spezielle Rehabilitationseinrichtungen wie zum Beispiel das Rehabilitationszentrum Rendsburg (siehe [36]) bieten mehrwöchige Programme für Spätertaubte an, die sämtliche Bereiche der Rehabilitation umfassen. Da für den Erhalt der Sprechfertigkeit jedoch ein kontinuierliches Training erforderlich ist, muß das Sprechen auch nach der Teilnahme an einer solchen Rehabilitationsmaßnahme kontrolliert und geübt werden. Hierfür ist ein Computersystem, das ein selbständiges Training ermöglicht, besonders gut geeignet.

2.1 Auswirkungen der Ertaubung auf die Sprechfertigkeit

Generell kann gesagt werden, daß hinsichtlich des Verfalls der Sprechfertigkeit Spätertaubter eine große Bandbreite besteht. Ausschlaggebend ist das Alter beim Eintritt der Ertaubung, wobei es weniger darauf ankommt, ob bereits feste Sprechmuster ausgebildet sind, sondern vielmehr, ob eine gewisse Flexibilität zur Anpassung an die Sprechkontrolle mit den eingeschränkten Daten vorhanden ist. Die Erhaltung der Sprechfertigkeit steht also in Beziehung zu den individuellen motorischen Fertigkeiten zum Zeitpunkt der Ertaubung, die wiederum mit dem Alter zusammenhängen. Weitere wesentliche Faktoren sind die Persönlichkeit, Gesprächigkeit, Intelligenz und das soziale Umfeld.

Grundsätzlich gilt, daß sich eine Ertaubung um so nachteiliger auswirkt, je früher sie eintritt. So verlieren ertaubte Kleinkinder die bereits vorhandene Sprache, wenn nicht sofort nach dem Eintritt des Hörverlustes therapeutische Maßnahmen ergriffen werden. Die Sprache wird dann meist dunkler, verlangsamt sich und verliert an artikulatorischer Deutlichkeit [5].

Da die Gruppe der Spätertaubten recht klein ist, sind die Auswirkungen einer Ertaubung auf die Sprechfertigkeit bisher nur unzureichend analysiert worden, und die wenigen umfassenden Untersuchungen beziehen sich ausschließlich auf die englische Sprache.

So gibt es zwei ausführliche Studien der Sprechfehler von Spätertaubten im englischsprachigen Raum. In der Penn-Studie [33] wird festgestellt, daß es bei Ertaubten eine große Bandbreite an Sprechfehlern gibt, von denen einige gehäuft vorkommen. In der Belfast-Studie [10] werden die Sprechfehler von 13 Spätertaubten umfassend analysiert. In [11] werden die in den beiden Studien ermittelten Sprechfehler beschrieben und Erklärungsmodelle für deren Ursachen vorgestellt. Wir werden im folgenden die wesentlichen Sprechfehler kurz erläutern; auf die Ursachen können wir hier nicht näher eingehen.

Die Verständlichkeit des Gesprochenen wird besonders stark durch das Alter beim Eintritt der Ertaubung beeinflußt; je später der Gehörverlust, desto besser verständlich bleibt die Sprache. So führt eine Ertaubung im Kindesalter meist zu einer recht unverständlichen Sprache, während die Auswirkungen im Erwachsenenalter geringer sind. Nicht zuletzt wirkt sich auch das Niveau der gesprochenen Materie auf die Verständlichkeit aus, d.h. bei Alltäglichem treten der Wortwahl entsprechend weniger Sprechfehler auf.

Die Studien zeigen aber, daß das Sprechen nach der Ertaubung nicht nur hinsichtlich der Verständlichkeit, sondern in viel tiefgreifenderer Weise eingeschränkt wird. Grundsätzlich kann

die Verständlichkeit des Gesprochenen sogar recht gut sein, obwohl eine beträchtliche Verschlechterung der Sprechqualität eingetreten ist.

Unter den Sprechfehlern lassen sich segmentale von globalen unterscheiden, wobei die segmentalen einzelne Phoneme betreffen, während sich die globalen auf die Eigenschaften des Sprechkontinuums beziehen.

Die Belfast-Studie zeigt, daß häufig sogenannte Präfix- oder Suffix-Auslassungen auftreten, d.h. Wortanfang oder Wortende werden verschluckt. Diese segmentalen Fehler treten meist in Clustern auf, was auf ein zeitweises Aussetzen der Sprechkontrolle hindeutet. Da die betroffenen Phoneme für die jeweilige Sprache spezifisch sind, werden wir nicht auf die in der Studie ermittelten konkreten Phonemgruppen eingehen.

Die Penn-Studie ermittelt folgende globale Sprechfehler:

- Sprechtempo: häufiger zu schnell als zu langsam
- Sprechlautstärke: häufig zu leise, selten aber auch zu laut
- Tonhöhe: recht häufig monoton
- Stimmlage: häufig zurückgenommen, selten heiser, rauh, flüsternd

Die Belfast-Studie zeigt weitere globale Sprechfehler auf:

- Timing: sehr häufig Vokalverlängerung,
  Konsonantenverlängerung besonders häufig am Wortanfang
- Akzentuierung: recht häufig übertriebene Betonung, selten falsche Betonung
- Intonation: selten völlig unpassende Satzintonation,
  selten starke Schwankungen der Tonhöhe

Weitere Untersuchungen, aber auch eigene Erfahrungen im Umgang mit Spätertaubten, bestätigen, daß die fehlende akustische Eigenkontrolle bestimmte globale Sprecheigenschaften in besonderem Maße beeinflußt.

So fällt die Steuerung der Sprechlautstärke besonders schwer, da sie nicht mehr automatisch der Umgebung angepaßt, sondern nur noch sprechmotorisch kontrolliert werden kann [38]. Die Mehrzahl der uns bekannten Spätertaubten spricht entweder extrem laut oder fast unverständlich leise.

Aber auch die Natürlichkeit des Sprechens, die durch Akzentuierung und Intonation geprägt ist, wird oft beeinträchtigt. Entweder wirkt das Sprechen leiernd-schleppend und monoton oder es kommt zu übertriebenen, unpassenden Betonungen. Auch in [9] wird die Notwendigkeit der auditiven Rückmeldung für die Kontrolle der Tonhöhe belegt. Einige der uns bekannten Spätertaubten haben eine sehr eigenwillige Wortakzentuierung und Satzintonation entwickelt, die nicht dem Sinn bzw. der Sprecherabsicht entspricht und das Gesprochene so entstellt, daß es kaum verständlich ist.

Häufig können wir eine Überartikulation beobachten, manchmal aber auch eine sehr undeutliche Artikulation, bei der vor allem die Endsilben verschluckt werden. In [39] wird dargestellt, daß die Auswirkungen auf die Artikulation um so gravierender sind, je früher die Ertaubung eingetreten ist.

2.2 Computersysteme für die Rehabilitation Hörgeschädigter

Für die Hör-/Spracherziehung werden in zunehmendem Maße Ersatzperzeptionssysteme in Form von Computersystemen entwickelt (vgl. [7, 21, 29, 31]). Die vorgestellten Systeme sind

für Fachleute in der Behandlung von Hörgeschädigten gedacht und können in der Rehabilitation Schwerhöriger sowie Gehörloser eingesetzt werden. Die Geräte sind jedoch vor allem als Hilfsmittel für die Sprachanbahnung bei gehörlosen Kindern, insbesondere im Rahmen der Frühförderung, konzipiert. Es lassen sich zwei Arten von Modellen unterscheiden. Die sogenannten Demonstrationsmodelle verdeutlichen lediglich modellhaft die artikulatorischen Vorgänge, während die Rückmeldesysteme eine visuelle oder auch akustische Rückmeldung über die reale Artikulation des Sprechers geben.

Zu den Demonstrationsmodellen zählt das vom Phonetischen Institut der Universität zu Köln entwickelte UK-Dynamo-Programm [17], das zu einer Tastatureingabe beliebiger Folgen von deutschen Sprachlauten aus mittsagittalen Schnittbildern der Artikulationsorgane einen Zeichentrickfilm des Bewegungsablaufes erzeugt. Von der Humboldt Universität Berlin wurde ein ähnliches System entwickelt [26].

Für das Absehtraining wurden bereits spezielle Präsentationssysteme vorgestellt [4], und es befinden sich weitere in der Entwicklung wie die Computer-Lernhilfe für den Absehunterricht vom Institut für Elektronik der Technischen Universität Berlin [8] und das Projekt MUSKAT zum medienunterstützten kommunikativen Absehtraining vom Seminar für Hörgeschädigten- und Sprachheilpädagogik der Universität zu Köln.

Die Rückmeldesysteme, welche die Sprechbewegungen ohne Schallaufnahme und ohne Schalldarstellung direkt erfassen, geben zwar recht exakte Auskunft über die zugrundeliegenden Artikulationsereignisse, sind jedoch mit dem Umstand einer Fremdkörpereinbringung in den Mundraum verbunden. Daher sind diese Systeme schwer handhabbar und werden im allgemeinen nicht für pädagogische, sondern eher für diagnostische Zwecke eingesetzt. Hierzu gehören die Zungensonographie mit der Ultraschalldurchleuchtungsmethode [6], die elektromagnetische Artikulographie [35] und die Elektropalatographie [41], bei der Sensoren in der Mundhöhle angebracht werden.

Die Mehrzahl der Rückmeldesysteme wandelt den Sprachschall per Mikrofon und verarbeitet die dem Sprachschall entsprechenden elektrischen Schwingungen digital. Die so gewonnenen Daten des komplexen Sprachschallverlaufs oder ausgewählter Sprachschallparameter werden in Strukturen umgesetzt, die als optische Muster visualisiert werden können. Da die Beziehung zwischen Sprachschall und Artikulation nicht eindeutig ist, können die Artikulationsereignisse aus diesen Daten nicht rechnerisch rekonstruiert werden [31]. So ist der Lerner darauf angewiesen, die dargestellten Schallelemente selbst zu interpretieren und den artikulatorischen Ereignissen zuzuordnen.

Zu diesen Rückmeldesystemen gehört das von der Universität Düsseldorf entwickelte SFT-Gerät [12, 13], das eine Sprach-Farbbild-Transformation vornimmt und dessen Prinzip an der RWTH Aachen weiterentwickelt wurde [18]. Der Fachbereich Kommunikationswissenschaften der Technischen Universität Berlin hat sich mit der Entwicklung eines Systems zur Visualisierung von Sprache in Echtzeit und einer Didaktik zum Einsatz dieses Systems im Lautsprachunterricht Gehörloser befaßt [22].

Das derzeit vielseitigste Rückmeldesystem ist der von IBM entwickelte Sprechspiegel [30, 32], der eine sofortige visuelle und zum Teil auch auditive Rückmeldung über den erzeugten Sprachschall bietet und inzwischen in der dritten Version vorliegt. Diese enthält Übungen zu den Schallelementen, Stimm- und Artikulationsübungen. Die Rückmeldung über die Artikulation bezieht sich jedoch lediglich auf Einzellaute und nicht auf ganze Wörter. Der Sprechspiegel bietet einerseits spielerische Übungen, die für den Einsatz in der Früherziehung geeignet sind, andererseits aber auch komplexe graphische Darstellungen und Analysen für diagnostische Zwecke sowie Funktionen für die Verwaltung von Klientendateien.

Die Möglichkeit eines computerunterstützten Sprachaufbaus bei Gehörlosen wurde an der Heilpädagogischen Fakultät der Universität zu Köln untersucht [45].

2.3 Eignung dieser Systeme für die Rehabilitation Spätertaubter

Zum gegenwärtigen Zeitpunkt gibt es kein Computersystem, das speziell für das Training der Sprechfertigkeit und nicht für den Sprachaufbau entwickelt wurde. Damit einhergehend gibt es auch kein System, das so konzipiert ist, daß es den Bedürfnissen Spätertaubter gerecht wird.

Da die bereits entwickelten Programme zum Hören- und Sprechenlernen nicht für das selbständige Üben ausgelegt sind, sondern stets die Anwesenheit einer Fachkraft erfordern, können sie in den Therapiezentren nur stundenweise und im häuslichen Bereich überhaupt nicht eingesetzt werden, obwohl dort heute meist ein multimediafähiger Computer vorhanden ist. In den Schulen für Hörgeschädigte wird die Anzahl der Lehrkräfte immer weiter eingeschränkt, so daß keine Doppelbesetzung mehr möglich ist und die Computersysteme dort fast nur noch im Rahmen der Frühförderung benutzt werden können.

Aufgrund der bereits vorhandenen Lautsprache sind Spätertaubte in der Lage, zeitweise auch selbständig zu üben. Dabei können sie von der Zeit vor dem Hörverlust profitieren, in der sie als Normalhörender ein mehrjähriges Sprechtraining mit akustischer Kontrollmöglichkeit durchlebt haben. In dieser Zeit haben sich die Sprechmuster meist so tief eingeschliffen, daß die Sprechfehler durch gezieltes Training beseitigt werden können.

3 Beschreibung des Systems ILSE

ILSE ist ein **I**nteraktives **L**ehr-/Lernsystem für **S**pät-**E**rtaubte,

das wir speziell für die Gruppe der Spätertaubten entwickeln, um ein selbständiges, eigenverantwortliches Training der Sprechfertigkeit zu ermöglichen.

ILSE steht auch für

- **I**ndividuelle Übungen zu
- **L**autstärke,
- **S**atz-Intonation,
- **E**inzelwort-Akzentuierung und -Artikulation.

3.1 Aufbau und Funktionen

ILSE soll eine visuelle Rückmeldung über ausgewählte Sprecheigenschaften geben. Nach Ermittlung der Umgebungsgeräusche werden die Sprechlaute über ein Mikrofon aufgenommen und vom System digitalisiert. Der Verlauf der Schallschwingungen kann mit Hilfe eines Oszillogramms dargestellt werden, in dem die Schwingungskurve im Auslenkungs-Zeitdiagramm gezeigt wird. Da diese Darstellung jedoch sämtliche Informationen über den Sprachschall enthält, ist sie derart komplex, daß der Betrachter diese kaum interpretieren kann. Für die Visualisierung bietet es sich daher an, zunächst die Sprachschallelemente in einem Analyseverfahren zu trennen. Die einzelnen Parameter werden dann eliminiert, bewertet oder verändert und anschließend wieder zu einem Bild zusammengefügt. Zur Darstellung von Sprecheigenschaften, die mit mehreren Sprachschallparametern korreliert sind, müssen diese Parameter miteinander

kombiniert werden. Dies ist zum Beispiel bei der Visualisierung der Prosodie erforderlich, da sich diese aus Intonation und Dynamik zusammensetzt, d.h. aus der Variation von Tonhöhe und Lautstärke innerhalb des Sprachflusses.

Anhand des visualisierten Diagramms können die in Abschnitt 2.1 beschriebenen Sprechfehler aufgedeckt werden. Dazu werden die Diagramme mit Methoden der Bildverarbeitung (vgl. [20]) ausgewertet und die Ergebnisse dem Lerner angezeigt.

Die Visualisierung muß den einzelnen Lerninhalten entsprechend auf unterschiedliche Weise erfolgen. Einerseits werden wir Frequenzdiagramme verwenden, andererseits aber auch eine Darstellung in Form von beweglichen Bildern vornehmen. Beim Entwurf der Algorithmen für die Visualisierung der Sprache werden wir einschlägige Untersuchungen zu den verschiedenen Darstellungsmöglichkeiten (siehe [22, 34]) sowie zur Phonetik (vgl. [28, 44]) miteinbeziehen.

Neben der bildhaften Darstellung der empfangenen Sprachsignale soll mittels Spracherkennung (vgl. [37]) auch eine Ausgabe als Text erfolgen, so daß der Lerner das von ihm Gesprochene lesen kann. Um die technischen Anforderungen in einem angemessenen Zeitrahmen realisieren zu können, werden wir für die Spracherkennung bestehende Softwarelösungen adaptieren (siehe [1, 2]). Dabei werden wir DLLs (Dynamic Link Libraries) zur Aufnahme und Auswertung der Sprache entwickeln. Das System selbst werden wir mit Hilfe einer Autorensoftware erstellen (vgl. [14]).

Die Oberfläche des Lehr-/Lernsystems wird eine intuitive und einheitliche Benutzerführung mit vielen Hilfestellungen enthalten, um ein selbständiges Üben zu ermöglichen. Bei herkömmlichen Lehr-/Lernsystemen erfolgt die Benutzerführung oft durch eine Kombination von Text- und Sprachausgabe. Da ILSE ohne die Anwesenheit einer hörenden Person von Spätertaubten bedient werden können soll, muß die Benutzerführung hier lautsprachfrei sein. Eine Figur namens Ilse dient als Ansprechpartner des Systems, übernimmt die Benutzerführung und gestaltet damit den Dialog persönlicher und ansprechender.

Um den Spätertaubten zum Sprechen zu motivieren, soll die Möglichkeit bestehen, das System durch Spracheingabe zu steuern. Dabei führt der Spätertaubte stets einen Dialog mit der das System repräsentierenden Ilse, wobei diese mittels Textausgabe antwortet.

Die Lerninhalte werden mit Hilfe von Spielen, Rätseln, Lückentexten und Dialogen verschiedener Schwierigkeitsgrade vermittelt. Ansprechende Grafiken, Spiele mit Wettbewerbscharakter und positives Feedback steigern die Motivation und den Lernspaß. Die spielerische Lernumgebung stärkt auch die Konzentrationskraft, so daß der Lerner für eine längere Zeitspanne effektiv mit dem System arbeiten kann.

Die Interaktivität des Systems gewährleistet eine individuelle, effektive Förderung [40], da der Lerner stets eine Rückmeldung über seine Defizite und Erfolge erhält. Die individuellen Lernfortschritte können in einem Übungsprotokoll festgehalten werden, und das vom Lerner Gesprochene kann als Audiodatei abgespeichert werden, damit es jederzeit zur Lernkontrolle verfügbar ist (Modell des Lerners).

Das System soll soweit wie möglich aus den Fehlern erkennen können, welche Strategie zu diesen geführt hat, damit es in der Lage ist, dem Lerner eine adäquate Lösungsstrategie anzubieten. Auch die für Spätertaubte typischen Sprechfehler sollen nach Möglichkeit vom System erkannt werden, so daß der Lerner eine Rückmeldung erhalten kann, was er verbessern und üben muß.

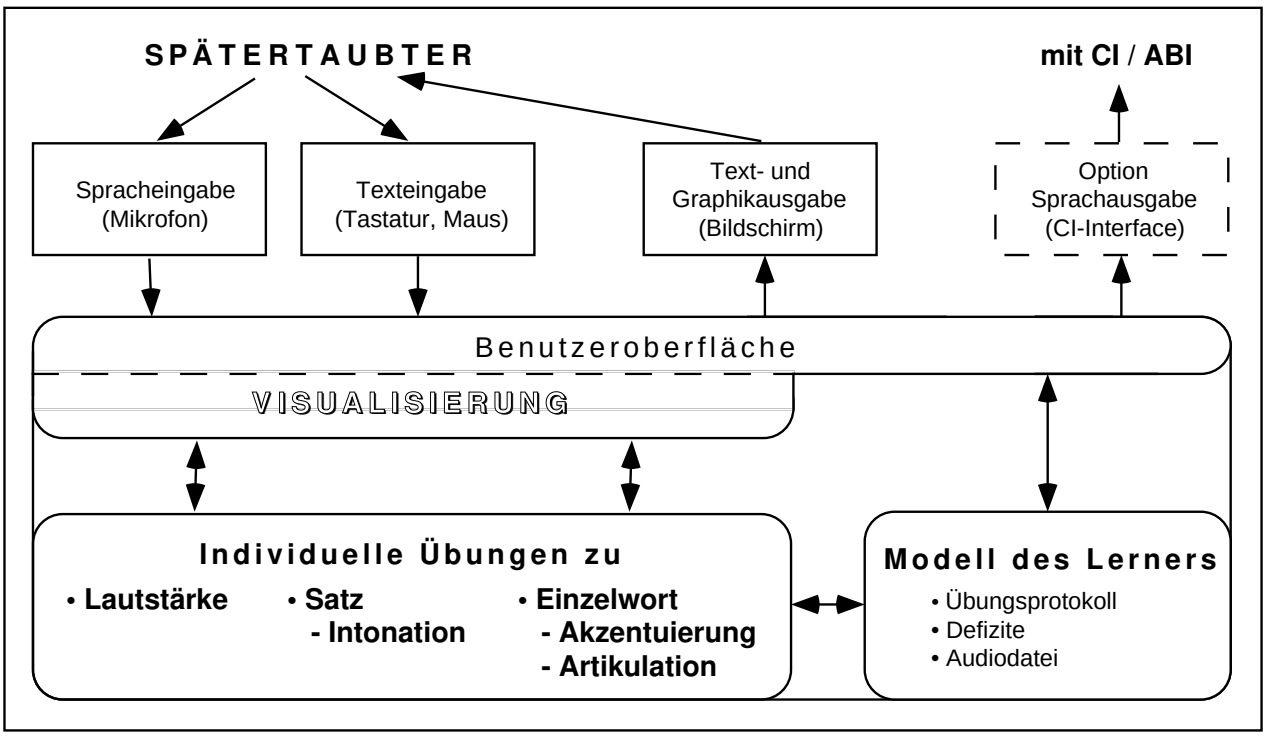
3.2 Die einzelnen Übungen

Um den Lerner nicht zu überfordern, werden in den Übungen nur ausgewählte Sprecheigenschaften betrachtet. Als Auswahlkriterien dienen die Untersuchungen der globalen Sprechfehler und unsere eigenen Erfahrungen im Umgang mit Spätertaubten. Da die Steuerung der Sprechlautstärke in der Regel besondere Schwierigkeiten bereitet, wird diese sowohl separat geübt als auch in den anderen Übungen mitberücksichtigt.

Das System wird Übungen zu den folgenden Themen enthalten:

- Lautstärke
- Satz-Intonation und Lautstärke
- Einzelwort-Akzentuierung und Lautstärke
- Einzelwort-Artikulation und Lautstärke

Für jedes Thema stellen wir spezielles Übungsmaterial zusammen. Bei der Auswahl achten wir in besonderem Maße auf den Bezug zum Alltagsgeschehen, denn diese Wörter bzw. Sätze müssen vom Lerner oft gesprochen und verstanden werden. Außerdem stärkt die Nähe zum Alltag die Motivation, sich über längere Zeit mit dem System zu beschäftigen. Wir wählen auch bekannte Kinderreime, denn diese ruhen fest im rhythmischen und melodischen Gedächtnis des Spätertaubten, der in seiner Kindheit hören konnte.



Überblick über die einzelnen Komponenten des interaktiven Lehr-/Lernsystems für Spätertaubte

Das Übungsmaterial wird folgende Bereiche umfassen:

- Wörter des Grundwortschatzes
- gebräuchliche Umgangsformen aus dem Alltagsgeschehen
- häufig vorkommende Fremdwörter
- bekannte Kinderreime

Die Wörter des Grundwortschatzes und die Fremdwörter ordnen wir alphabetisch, die Umgangsformen und Kinderreime thematisch. Das Übungsmaterial wird nicht nur im Rahmen der Spiele, Rätsel, Lückentexte und Dialoge erfaßt, sondern kann auch mit Hilfe dieser Ordnung gezielt abgefragt werden.

Um die individuelle Situation des Lerners zu berücksichtigen, soll dieser das Übungsmaterial durch eigene Wörter oder Sätze ergänzen können.

Wir haben eine etwa 800 Wörter und alltägliche Umgangsformen umfassende Liste erstellt, zu der vom Fachgebiet Technische Informatik (Rigoll) der Universität Duisburg ein Spracherkennungssystem entwickelt wurde. Dieses werden wir nun in der Arbeit mit Spätertaubten testen, um herauszufinden, inwieweit und in welcher Richtung Adaptionen gemacht werden müssen.

3.3 Form der Evaluation

Unser Forschungsvorhaben umfaßt neben der Entwicklung auch die Evaluation des interaktiven Lehr-/Lernsystems (vgl. [15]). Schon in der Entwicklungsphase werden wir die einzelnen Übungen immer wieder in der Rehabilitation Spätertaubter erproben. Dazu arbeiten wir mit der Rheinischen Schule für Schwerhörige Düsseldorf und der Kollegschule der Rheinisch-Westfälischen Berufsschule für Hörgeschädigte Essen zusammen. Außerdem haben wir Kontakt zu einer Gruppe von spätertaubten Erwachsenen, die uns Anregungen geben. Diese entwicklungsbegleitende Evaluation wird sich zwar nur auf eine recht kleine Gruppe von Spätertaubten beziehen, hat aber den Vorteil, daß die Erkenntnisse sofort umgesetzt werden können, um die Qualität des Systems zu verbessern.

Gegen Ende des Jahres 1998 soll eine Beta-Version des Systems zur Verfügung stehen, die im Sprechtraining Spätertaubter getestet wird. Dies ermöglicht eine großflächiger angelegte Evaluation als während der Entwicklung. Die Ergebnisse der Entwicklung, Implementierung und Evaluation des interaktiven Lehr-/Lernsystems für Spätertaubte werden abschließend in der Dissertation zusammengefaßt.

4 Einsatzmöglichkeiten von ILSE

Das interaktive Lehr-/Lernsystem ILSE ist für das selbständige, eigenverantwortliche Training der Sprechfertigkeit Spätertaubter konzipiert. Da für die Arbeit mit dem System keine Fachkraft anwesend sein muß, kann ILSE sowohl in der Schule als auch im häuslichen Bereich eingesetzt werden.

ILSE kann auch als Sprechunterrichtshilfe für Schwerhörige verwendet werden, wenn diese über eine gut ausgebildete Lautsprache verfügen.

Damit das System auf einem gängigen PC, wie er in den meisten Haushalten und Schulen zu finden ist, läuft, wollen wir möglichst geringe Anforderungen an die Hardware der Zielmaschine stellen. Andererseits muß die Sprachaufnahme aber qualitativ hohen Ansprüchen genügen. Zielmaschine unseres Lehr-/Lernsystems ist daher ein IBM-kompatibler Standardrechner mit Pentium-Prozessor, CD-ROM-Laufwerk, Soundkarte, Lautsprechern und einem auf das Spracherkennungssystem abgestimmten Mikrofon.

4.1 Einsatz in der Rehabilitation von CI- und ABI-Trägern

Sind bestimmte medizinisch-audiologische und logopädisch-psychologische Voraussetzungen erfüllt, so kann ein Spätertaubter ein Cochlea Implant (CI) erhalten, welches ihm durch direkte Reizung des Hörnervs wieder Höreindrücke verschafft. In Deutschland gibt es bereits mehr als 1.500 CI-Träger. Sind beide Hörnerven nicht mehr funktionstüchtig, so kann seit jüngerer Zeit auch ein Auditory Brainstem Implant (ABI), d.h. eine Hirnstammprothese, eingesetzt werden. Sowohl die CI- als auch die ABI-Träger müssen nach der Operation lernen, die neuen Höreindrücke mit den alten Hörerinnerungen in Verbindung zu bringen. In der postoperativen Rehabilitation geht es daher in erster Linie um ein Hörtraining [3, 24].

Damit unser Lehr-/Lernsystem auch für diese Spätertaubten einen optimalen Lernerfolg bieten kann, wird es die Option zur Sprachausgabe beinhalten. Dann erfolgt die Rückmeldung über die Sprecheigenschaften nicht nur visuell, sondern auch akustisch. Um den CI- bzw. ABI-Trägern das Verstehen des Gesprochenen zu erleichtern, kann der zu dem Implant gehörende Sprachprozessor direkt mit dem Computer verbunden werden (CI-Interface), so daß nur der Nutzschall empfangen wird.

Auch für diese spezielle Gruppe Spätertaubter werden wir die Eignung unseres Systems sowohl in einer entwicklungsbegleitenden als auch in einer abschließenden Evaluation untersuchen. Dazu arbeiten wir mit der Rehabilitationsklinik Werscherberg zusammen, die spezielle prä- und postoperative Programme für Kinder und Erwachsene durchführt. Dort haben wir die Möglichkeit, Erfahrungen mit den Therapeuten, CI- und ABI-Trägern auszutauschen und das System zu testen.

4.2 Weiterführende Aspekte

Bei einigen fortschreitenden Krankheiten wie Neurofibromatose besteht eine sehr große Wahrscheinlichkeit, daß es zur Ertaubung des Patienten kommt. Für solche Fälle soll die Möglichkeit gegeben sein, das Übungsmaterial vor dem Verlust des Gehörs aufzunehmen und so die eigene Stimme zu konservieren. In der Rehabilitation kann dann auf die eigene Stimme vor der Ertaubung zurückgegriffen werden. Auch wenn die Artikulation der betroffenen Person nicht den hohen Anforderungen genügt, die für das Üben mit dem System erforderlich sind, so kann das Arbeiten mit der eigenen Sprachvorgabe einen wesentlichen Beitrag zur Motivation leisten. Wir haben die Chance, dieses Konzept bei einer Person, die im März 1997 aufgrund eines operativen Eingriffes vollständig ertaubt ist, zu erproben. Dazu haben wir ihre Stimme vor der Ertaubung aufgezeichnet und digitalisiert.

Es ist eine interessante Fragestellung, inwieweit eine Person nach der Ertaubung ihre individuellen Sprecheigenschaften wie Stimmlage, Sprechlautstärke, Artikulationsweise und Intonationsmuster beibehält. Wir werden dieser Frage nicht nur bei dieser einzelnen Person, sondern auch in Zusammenarbeit mit der uns bekannten Gruppe Spätertaubter nachgehen.

Literatur

[1] Bager, J.: „Computer, ...“ — Sprache als Forschungsobjekt und Eingabemedium. c't 4 / 97

[2] Bager, J.; Kuri, J.: Nie wieder tippen? - Spracherkennung für Windows und OS/2. c't 4 / 97

[3] Banfai, P.: Das Cochlea Implant. Heidelberg: Groos (1985)

[4] Benguerel, A.-P. et al: Design of a Computer-aided Speechreadingtraining System. Annual Bulletin RILP 22 (1988) 103-114

[5] Biesalski, P.; Frank, F. (Hrsg.): Phoniatrie - Pädaudiologie: Physiologie, Pathologie, Klinik, Rehabilitation. Stuttgart, New York: Thieme (1982)

[6] Böckler, R. et al: Zungensonographische Unterstützung der Anbahnung des /k/-Lautes bei einem gehörlosen Kind. Hörgeschädigtenpädagogik 42 (1988) 337-343

[7] Bonfranchi, R.: Computer-Didaktik in der Sonderpädagogik. Luzern: Edition SZH/SPC (1992)

[8] Bothe, H.-H.; Krause, M.; Naunin, D.: Eine Computer-Lernhilfe für den Absehunterricht. Forschungsvorhaben der Institute für Elektronik und Kommunikationswissenschaft der TU Berlin. Unveröffentlicht.

[9] Collins, M.J.: Fundamental Frequency and Formant Structure in Deaf Speech: Influence of a Hearing Aid in a Case Study. J. Acoust. Soc. Am. 65, Suppl. 1 (1979) 569

[10] Cowie, R.I.D. et al: A Study of Speech Deterioration in Postlingually Deafened Adults. J. Laryngol. Otol. 96 (1982) 101-112

[11] Cowie, R.I.D.; Douglas-Cowie, E: Speech Production in Profound Postlingual Deaf. In: Lutman, M.E.; Haggard, M.P. (Edit.): Hearing Science and Hearing Disorders. London: Academic Press (1983) 183-230

[12] Esser, G.; Nolte, P.; Printzen, R.: Sprach-Farbbild-Transformation (SFT): Erweiterte Möglichkeiten beim Sprachunterricht Hörgeschädigter. In: Bund Deutscher Taubstummenlehrer (Hrsg.): Arbeitstagung für Hörerziehung. Burg Feuerstein. Essen (1983)

[13] Esser, G. et al: Verbesserung der Sprechfähigkeit durch Anwendung der Sprach-Farbbild-Transformation (SFT) bei schwerhörigen und gehörlosen Jugendlichen. In: Bundesministerium für Arbeit und Sozialforschung (Hrsg.): Forschungsbericht-Sozialforschung, Bd. 225, Bonn (1993)

[14] Freibichler, H.: Werkzeuge zur Entwicklung von Multimedia. In: Issing, L.J.; Klimsa, P. (Hrsg.): Information und Lernen mit Multimedia. Weinheim: Psychologie-Verl.-Union (1995) 221-240

[15] Fricke, R.: Evaluation von Multimedia. In: Issing, L.J.; Klimsa, P. (Hrsg.): Information und Lernen mit Multimedia. Weinheim: Psychologie-Verl.-Union (1995) 401-413

[16] Heese, G.: Sprachpflege-Kurs für Schwerhörige und Ertaubte. Berlin: Marhold (1963)

[17] Heike, G.; Philips, J.E.; Hilger, S.: Computergrafikdarstellungen von Artikulationsbewegungen zur Unterstützung des Artikulationstrainings. Sprache-Stimme-Gehör 10 (1986) 4-10

[18] Hesselmann, N.: Computerunterstütztes visuelles Artikulationtraining. Sprache-Stimme-Gehör 5 (1982) 82-88

[19] Jann, P.A.: Methoden der Sprachvermittlung beim gehörlosen Kind: Zur Grundlegung einer systematischen Spracherwerbsforschung. Heidelberg: Edition Schindele (1994)

[20] Janser, A.; Luther, W.; Otten, W.: Computergraphik und Bildverarbeitung. Braunschweig, Wiesbaden: Vieweg (1996)

[21] Jussen, H.; Kloster-Jensen, M.; Wisotzki, K.H.: Lautbildung bei Hörgeschädigten: Abriß einer deutschen Sprachlautlehre. Berlin: Edition Marhold im Wiss.-Verl. Spieß, 3. überarb. Aufl. (1994)

[22] Krause, M.: Visualisierung von Sprache für die Sprechschulung Gehörloser. it+ti - Informationstechnik und Technische Informatik 39, 2 (1997) 28-32

[23] Kultusminister des Landes NRW: Verordnung über die Feststellung des sonderpädagogischen Förderbedarfs und die Entscheidung über den schulischen Förderort vom 22.05.1995

[24] Lehnhardt, E.; Bertram, B.: Rehabilitation von Cochlea-Implant-Kindern. Berlin: Springer (1991)

[25] Lienhard, P.: Ertaubung als Lebenskrise: Bewältigung des Gehörverlustes im Erwachsenenalter. Luzern: Edition SZH/SPC (1992)

[26] Lindner, G.: Graphische Modellierung der Sprechbewegung mit Hilfe eines Kleincomputers. In: Academy of Sciences of the Estonian S.S.R. (Edit.): Proceedings XIth international congress of phonetic science 1987 in Tallin. Vol. 5. Tallin (1987) 302-305

[27] Maisch, G.; Wisch, F.H.: Gebärdenlexikon. Bd. 1-4. Hamburg: Verl. hörgeschädigte kinder (^{1}1987)

[28] Neppert, J.M.H.; Ptursson, M.: Elemente einer akustischen Phonetik. Hamburg: Buske (1986)

[29] Neppert, J.M.H.: Funktionsgeschichtlicher Abriß der Entwicklung von apparativen Sprechunterrichtshilfen vom Neubeginn nach dem zweiten Weltkrieg bis zu den heutigen Computersystemen. Hörgeschädigtenpädagogik 43 (1989) 208-221

[30] Neppert, J.M.H.: Der IBM-Sprechspiegel als optisch-visuelles Rückmeldesystem in der Sprechtherapie und im Artikulationsunterricht bei Hörgeschädigten. Retour Progrès (1990) 12-26

[31] Neppert, J.M.H.: Der Computer als Lernhilfe in der Hör-, Abseh- und Sprecherziehung. In: Jussen, H.; Claussen, W.H. (Hrsg.): Chancen für Hörgeschädigte: Hilfen aus internationaler Perspektive. München, Basel: E. Reinhardt (1991) 346-352

[32] Neppert, J.M.H.: Der IBM-Sprechspiegel II wurde auf der Cebit '92 vorgestellt: Was bringt die zukünftige Version für die Sprech-Erziehungs-Praxis? Hörgeschädigtenpädagogik 46 (1992) 164-192

[33] Penn, J.: Voice and Speech Patterns in the Hard of Hearing. Acta Otolaryngol. Suppl. 124 (1955)

[34] Ruoß, M.; Becker, R.; Eyferth, K.: Sichtbares Sprechen: Untersuchungen verschiedener Darstellungsmöglichkeiten. Zeitschrift für experimentelle und angewandte Psychologie 35 (1988) 628-656

[35] Schönle, P.-W.: Elektromagnetische Artikulographie: Ein neues Verfahren zur klinischen Untersuchung der Sprechmotorik. Berlin, Heidelberg: Springer (1988)

[36] Schuck, K.D.: Aktivität und Veränderung: Ziele und Aufgaben der Erwachsenenarbeit bei Schwerhörigen und Ertaubten. In: Jussen, H.; Claussen, W.H. (Hrsg.): Chancen für Hörgeschädigte: Hilfen aus internationaler Perspektive. München, Basel: E. Reinhardt (1991) 192-199

[37] Schukat-Talamazzini, E.G.: Automatische Spracherkennung: Grundlagen, statistische Modelle und effiziente Algorithmen. Braunschweig, Wiesbaden: Vieweg (1995)

[38] Siegel, G.M.; Pick, H.L.: Auditory Feedback in the Regulation of Voice. J. Acoust. Soc. Am. 56, 5 (1975) 1618-1624

[39] Siegel, G.M. et al: Delayed Auditory Feedback with Children. J. Speech Hearing Res. 23 (1980) 802-813

[40] Strzebkowski, R.: Realisierung von Interaktivität und multimedialen Präsentationstechniken. In: Issing, L.J.; Klimsa, P. (Hrsg.): Information und Lernen mit Multimedia. Weinheim: Psychologie-Verl.-Union (1995) 269-303

[41] Timmermann, G.: Untersuchungen zur artikulatorischen Dynamik im Deutschen mit Hilfe der Elektropalathographie. In: Barry, W.J.; Kohler, K.J. (Hrsg.): Institut für Phonetik der Universität Kiel, Arbeitsberichte Nr. 19. Kiel (1982) 5-57

[42] Tings, F.: Worterarbeitung an Gehörlosenschulen. Villingen-Schwenningen: Neckar (1976)

[43] Treue, S.: Notwendigkeit und Aufbau von Absehkursen für schwerhörige und ertaubte Erwachsene. In: Jussen, H.; Claussen, W.H. (Hrsg.): Chancen für Hörgeschädigte: Hilfen aus internationaler Perspektive. München, Basel: E. Reinhardt (1991) 265-272

[44] Ungeheuer, G.: Das Phonemsystem der deutschen Hochlautung. In: T. Siebs: Deutsche Aussprache. Berlin (1969) 27-42

[45] Wisotzki, K.H.: Computerunterstützer Sprachaufbau in der Sonderschule: Eine Untersuchung zur Didaktik. Schriften zur Hörgeschädigtenpädagogik, Bd. 18. Berlin: Marhold (1988)