

Markus Mertesacker

Aspekte des Planens im Computerunterstützten Unterricht

Dargestellt am Beispiel einer Architektur für Tutor-Systeme

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 1995 GRIN Verlag
ISBN: 9783656362050

Dieses Buch bei GRIN:

<https://www.grin.com/document/207707>

Markus Mertesacker

Aspekte des Planens im Computerunterstützten Unterricht

Dargestellt am Beispiel einer Architektur für Tutor-Systeme

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Aspekte des Planens im Computerunterstützten Unterricht

Dargestellt am Beispiel
einer Architektur für
Tutor-Systeme

Diplomarbeit von
Markus Mertesacker

Februar 1995

Vorwort

Die vorliegende Diplomarbeit entstand im Verlauf eines Projekts *Tutor-Systeme* am Institut für Software-Technik (IST) der Universität Koblenz-Landau an der Abteilung Koblenz. Ausgangspunkt für diese Arbeit war die Dissertation von Uwe Dumsclaff, der sich seit einigen Jahren mit Forschungsarbeiten im Bereich der tutoriellen Systeme aus der Sicht des Software Engineering beschäftigt. Bei ihm möchte ich mich für die vorbildliche Betreuung und die konstruktive und vertrauensvolle Zusammenarbeit, die auch über den Rahmen der Diplomarbeit hinausging, bedanken. Seine motivierende Art und seine wertvollen Anregungen und Ratschläge trugen wesentlich zu dieser Arbeit bei.

Herrn Prof. Dr. Jürgen Ebert, der mich von Beginn meines Studiums an über zahlreiche Vorlesungen begleitete, danke ich für die Unterstützung und sein Interesse an der Mitbetreuung. Er gab mir die Möglichkeit, über mehr als zwei Jahre hinweg als studentische Hilfskraft erste praktische Erfahrungen in der Teamarbeit und im Umfeld von CASE-Projekten zu machen.

Den CASE-Hiwis danke ich für die freundliche und unkomplizierte Aufnahme in ihre Projektgruppe, für die Hilfsbereitschaft und Unterstützung bei etlichen Implementations- und Layout-Fragen und nicht zuletzt für die zuvorkommende Versorgung mit frischem Kaffee.

Bei Peter Dahm möchte ich mich herzlich für das mühevollen und zeitintensive Korrekturlesen bedanken.

Kurzfassung

Aspekte der konventionellen Unterrichtsplanung können im Computerunterstützten Unterricht verwendet werden mit dem Ziel, den Unterricht durch eine geeignete Software in einem Tutor-System planen und ausführen zu lassen.

Auf der Basis einer gegebenen Architektur für Tutor-Systeme wird eine Planungskomponente geschaffen, die unter Einbeziehung von Tutor-Strategien Unterrichtspläne berechnet und zur Ausführung bringt. Die theoretische Fundierung des Planens ist einerseits in den Methoden der *Künstlichen Intelligenz* und andererseits in der Graphentechnologie verankert.

Die gegebene Architektur für Tutor-Systeme sowie das Modul zur Planung und Planausführung werden durch eine Graphenrepräsentation realisiert. Mittels einer hierzu entwickelten Beschreibungssprache für die Dokumente eines Tutor-Systems wird eine rudimentäre Laufzeitumgebung für eine computerunterstützte Unterrichtsplanung zur Verfügung gestellt.

Inhaltsverzeichnis

1	Einführung	1
2	Ansätze von Systemen für Computerunterstützten Unterricht	5
2.1	CAI-Systeme	5
2.2	Konzepte von ITS	7
2.3	Der Hypertextbasierte Ansatz	9
2.4	Eine Architektur für Tutor-Systeme	10
3	Eine Architektur für Tutor-Systeme	13
3.1	Das Inhaltsmodul	13
3.1.1	Konzepte	14
3.1.2	Prozeduren	19
3.2	Das Lernermodul	30
3.3	Das Tutormodul	35
3.4	Das Interaktionsmodul	40
4	Planen	45
4.1	Was ist ein Plan?	45
4.2	Repräsentationen eines Planungssystems	46
4.2.1	Bestandteile eines Planungssystems	46
4.2.2	Formale Repräsentation von Plänen	49

4.2.3	Repräsentation von Plänen durch Graphen	53
4.3	Lineare Planungsmethoden	58
4.3.1	Planen als Inferenz	58
4.3.2	Planen als Suche	60
4.3.3	Ein Verfahren zum linearen Planen	63
4.4	Nichtlineare Pläne	72
4.4.1	Begriff der Nichtlinearität	72
4.4.2	Teilzielinteraktionen	73
4.4.3	Ein Verfahren zum nichtlinearen Planen	74
4.5	Planungsheuristiken und -strategien	80
4.5.1	Opportunistisches Planen im CUU	81
4.5.2	Dynamisches Planen im CUU	82
4.5.3	Gemischte Planungsstrategien im CUU	82
5	Unterrichtsplanung in einer Architektur für Tutor-Systeme	85
5.1	Tutorgraphen und -pläne	85
5.1.1	Tutorgraphen	86
5.1.2	Tutorpläne	96
5.2	Tutor-Strategien	97
5.2.1	Was ist eine Tutor-Strategie?	97
5.2.2	Repräsentation von Tutor-Strategien	101
5.2.3	Tutor-Strategien und Kreise in Tutorplänen	109
5.3	Generierung von Tutorplänen aus Tutorgraphen	111
5.3.1	Möglichkeiten zur Berechnung von Tutorplänen	111
5.3.2	Plangenerierung ohne Verwendung von Tutor-Strategien	112
5.3.3	Anwendung von Tutor-Strategien bei der Plangenerierung	113
5.3.4	Plangenerierung und Kreisbehandlung	118

5.4	Ausführbarkeit von Tutorplänen	124
5.4.1	Opportunistische Ausführung	124
5.4.2	Dynamische Planausführung	127
5.4.3	Planausführung und Kreisbehandlung	132
6	Schlußbetrachtungen und Ausblick	135
	Anhang	139
A	Syntax der Dokument-Beschreibungssprache <i>DoBs</i>	139
A.1	<i>DoBs</i> für Dokumente des Inhaltsmoduls	139
A.1.1	<i>DoBs</i> für Konzept-Dokumente	139
A.1.2	<i>DoBs</i> für Datenfluß-Dokumente	139
A.1.3	<i>DoBs</i> für Pre-Post-Dokumente	140
A.1.4	<i>DoBs</i> für Kontrollfluß-Dokumente	140
A.2	<i>DoBs</i> für Dokumente des Lernermoduls	140
A.3	<i>DoBs</i> für Dokumente des Tutormoduls	140
A.4	<i>DoBs</i> zur Erzeugung von Tutorgraphen	140
A.5	<i>DoBs</i> zur Spezifikation von Tutor-Strategien	141
A.6	<i>DoBs</i> zur Erzeugung von Tutorplänen	141
A.7	<i>DoBs</i> zur Ausführung von Tutorplänen	141
A.8	<i>DoBs</i> zur Definition von Strategie-Listen	141
B	Ablaufprotokolle der Algorithmen	142
B.1	Algorithmus 4.1 in Beispiel 4.4	142
B.2	Algorithmus 4.3 in Beispiel 4.5	143
B.2.1	Vorwärts-Suche	143
B.2.2	Rückwärtstraversierung	146
B.3	Algorithmus 5.1 in Beispiel 5.1	148

B.4	Algorithmus 5.3 in Beispiel 5.4	149
B.5	Algorithmus 5.4 in Beispiel 5.5	152
B.5.1	Vorwärts-Suche	152
B.5.2	Rückwärtstraversierung	155
B.6	Algorithmus 5.5 in Beispiel 5.6	157
B.7	Algorithmus 5.7 in Beispiel 5.6	158
Literatur		165
Stichwortverzeichnis		169

1 Einführung

Ein Zusammenhang zwischen *Computerunterstütztem Unterricht* und *Aspekten des Planens* wird für den Leser möglicherweise auf den ersten Blick nicht klar ersichtlich sein. Wenn man sich überlegt, wie ein Lehrer seinen Unterricht vorbereiten könnte, dann wird man erkennen, daß eine seiner Hauptaufgaben darin besteht, den Unterrichtsstoff in geeigneter Art und Weise — also möglichst nach didaktischen Gesichtspunkten — in einer sinnvollen Reihenfolge anzuordnen. Zur Unterstützung der Vorgehensweise im Zuge einer Unterrichtsvorbereitung sind Methoden und Heuristiken der Planung für eine Lehrperson unerlässlich. Eine Unterrichtsvorbereitung stellt unter diesem Blickwinkel betrachtet bereits ein *Planungsproblem* dar.

Computerunterstützter Unterricht, im folgenden mit *CUU* abgekürzt, soll einem Lerner zur Vermittlung von Inhalten mit Hilfe eines Computers und einer entsprechenden Lernsoftware dienen. Vorherrschend in diesem Bereich sind Systeme, bei denen ein fester Unterrichtsverlauf (*Curriculum*) vorgegeben ist. Anzustreben ist jedoch ein System, welches sich an jeden einzelnen Lerner sowohl bezüglich seines Wissens als auch seiner Lernfähigkeiten derart anpaßt, daß individuelle Unterrichtspläne abhängig vom Vorwissen des Lerners und den von ihm zu erreichenden Lernziele berechnet werden können. Im vorliegenden Bericht wird versucht, Aspekte der individuellen und automatischen Unterrichtsplanung in den CUU einzubeziehen. Dies geschieht durch die Integration einer Planungskomponente in eine gegebene Architektur für Tutor-Systeme.

Innerhalb des CUU gibt es bislang eine Vielzahl von Ansätzen, welche sich in völlig verschiedene Richtungen entwickelt haben. Das gesamte Spektrum erstreckt sich von *Computer Assisted Instruction*-Systemen (CAI) und deren Autorensprachen und -systemen bis hin zu leistungsfähigen *Intelligent Tutoring Systems* (ITS), welche insbesondere Mittel und Methoden der *Künstlichen Intelligenz* (KI) verwenden. CAI-Systeme werden zu den *konventionellen* Systemen im CUU gerechnet, die ITS zu den *wissensbasierten* Systemen. Parallel dazu haben sich im Bereich der Inhaltsvermittlung per Computer die *hypertextbasierten* CUU-Systeme etabliert. In Kapitel 2 wird ein kurzer Überblick zu diesen Ausprägungen gegeben.

Ein primäres Merkmal moderner CUU-Systeme ist deren Modularisierung in einzelne Komponenten hinsichtlich der beteiligten Akteure und Schnittstellen. Eine gebräuchliche Aufteilung ist die Unterscheidung in eine *Wissensrepräsentationskomponente*, eine

Lernerkomponente sowie in eine *Tutorkomponente* und eine *Interaktionskomponente*. Die wichtigsten Merkmale dieser Module werden in Kapitel 2 herausgearbeitet. Als Beispiel für ein nach diesen Komponenten konzipiertes CUU-System wird am Ende des Kapitels eine konkrete Architektur für Tutor-Systeme nach Dumsloff [Dums94] vorgestellt.

Die in Kapitel 3 ausführlicher beschriebene Architektur für Tutor-Systeme wurde anhand einer ITS-Standardstruktur (s. [TePa87, S.321]) entwickelt und mit Hilfe einer Graphenrepräsentation prototypisch implementiert. Eine Autorenschnittstelle dazu steht auf der Ebene einer eindimensionalen Beschreibungs-Sprache zur Verfügung, die es einem Autor ermöglicht, Dokumente des Inhalts-, Lerner- und Tutormoduls zu definieren. Eine Möglichkeit zur Formulierung von Interaktions-Dokumenten ist bisher nicht vorgesehen.

Bei der Erstellung von Unterrichtseinheiten in CUU-Systemen werden Autoren meist durch spezielle Autorensoftware unterstützt, die primär der Eingabe von zu vermittelnden Inhalten und eventuell zu verwendenden Unterrichts-Strategien dient. Im Kontext solcher *Tutor-Strategien* kommt dem Autor eine gewisse Unterrichts-Vorplanung zu, die je nach gewählter Strategie den gewünschten Unterrichtsstoff in eine didaktisch geeignete Reihenfolge bringt. Diese Unterrichtssequenzen werden dann in der Lerner-situation zur Ausführung gebracht. Die Erstellung von Unterrichtssequenzen nach einer bestimmten Tutor-Strategie kann somit als ein *Planungsproblem* betrachtet werden. Mit der Lösung dieses Planungsproblems ist es vorstellbar, eine Unterrichtsplanung nach vorher vom Autor definierten Strategien durch CUU-System selbst durchführen zu lassen. Aufgabe einer Komponente zur Unterrichtsplanung wird somit die Herstellung einer sinnvollen Anordnung von auszuführenden Lehrhandlungen sein, die auf ein vom Lerner zu erreichendes Lehrziel gerichtet ist.

Ziel dieser Arbeit ist es, *Aspekte der Planung* im Rahmen des CUU zu thematisieren und insbesondere auf eine konkrete Architektur für Tutor-Systeme anzuwenden. Im vorliegenden Bericht wird versucht, Planungsmethoden aus der *Künstlichen Intelligenz* mit der gegebenen Tutor-System-Architektur so zu kombinieren, daß am Ende ein ausführbares prototypisches Tutor-System mit Planungskompetenz zur Verfügung steht. Die durch eine Planungskomponente generierten Pläne können jedoch *nichtlinearen* Charakter besitzen. Dies erfordert geeignete Heuristiken zur Linearisierung der Pläne, um zu einer ausführbaren Unterrichtssequenz zu gelangen. In diesem Kontext spielen wiederum Tutor-Strategien eine wesentliche Rolle. Der Begriff des Planens wird in Kapitel 4 unter Einbeziehung von Planungsmethoden und -heuristiken der KI behandelt. Dort werden insbesondere *lineare* und *nichtlineare Planungsverfahren* besprochen. Diese Verfahren werden auf der Basis von Graphenalgorithmien vorgestellt..

Mit dem Wissen über *Planungsmethoden* und über eine konkrete *Tutor-System-Architektur* werden die beiden Themenkomplexe in Kapitel 5 zusammengeführt. Dort werden zunächst die für eine Unterrichtsplanung notwendigen Strukturen wie *Tutor-graphen* und *-pläne* definiert. Konkrete *Tutor-Strategien* werden eingeführt und eine formale Repräsentation für diese vorgeschlagen. Weiter werden verschiedene Möglich-

keiten der *Plangenerierung* und *-ausführung* diskutiert, insbesondere unter dem Aspekt der Anwendung von Tutor-Strategien.

In Kapitel 6 dieses Berichts erfolgt eine Zusammenfassung der durch diese Arbeit erreichten Ergebnisse und ein Ausblick auf Erweiterungsmöglichkeiten des prototypischen Tutor-Systems mit Planungskompetenz.

Im nun folgenden Kapitel wird ein Überblick über historische und neuere Ansätze verschiedener CUU-Systeme gegeben. Dabei wird sowohl auf die wichtigsten Merkmale von CAI und ITS als auch auf strukturelle Unterschiede bei konventionellen, hypertextbasierten und wissensbasierten CUU-Systemen eingegangen. Anschließend wird das grundlegende Konzept der bereitgestellten Architektur für Tutor-Systeme nach Dumsclaff [Dums94] skizziert.

