

In2Math – Interaktive Mathematik- und Informatikgrundausbildung

www.in2math.de

In2Math Projektpartner:

Peter Baumgartner, Institut für Informatik, Universität Koblenz-Landau

Barbara Grabowski, Hochschule für Technik und Wirtschaft, Saarbrücken

Walter Oevel, Institut für Mathematik, Universität Paderborn

Erica Melis, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken

1 Über das *In2Math* Projekt

1.1 Förderung und Partner

Das *In2Math* Projekt wurde vom BMBF gefördert und hatte eine Laufzeit von drei Jahren (2001 bis 2003). Die beteiligten Partner sind die Universität Koblenz-Landau (Prof. Dr. Ulrich Furbach, Koordinator), Humboldt-Universität zu Berlin (Dr. Marco Roczen), die Universität Paderborn (Prof. Dr. Benno Fuchssteiner), die Hochschule für Technik und Wirtschaft des Saarlandes (Prof. Dr. Barbara Grabowski), sowie die Universität des Saarlandes (Prof. Dr. Jörg Siekmann, PD Dr. Erika Melis). Des weiteren waren die Unterauftragnehmer DFKI Saarbrücken, die Sci-Face GmbH, die SIT GmbH, der Springer-Verlag und die Firma ergosign beteiligt.

1.2 Ziel des Projektes

Ziel des Projekts ist die Unterstützung von Kernbestandteilen der Grundkurse Mathematik und Theoretische Informatik für die Grundausbildung im Rahmen von Diplom- und BA/Master-Studiengängen Mathematik und Informatik. Dies beinhaltet die Gestaltung von interaktiven und multimedialen Lehrmaterialien zu Algebra, Analysis, Logik und Statistik, so dass diese im Normalbetrieb in der Hochschulausbildung eingesetzt werden können. Die angesprochenen Fächer gelten bei Studenten im Allgemeinen als sehr anspruchsvoll, so dass eine bessere Aufbereitung der Inhalte als sie durch Lehrbücher alleine möglich ist, dringend geboten scheint. Zu diesem Zweck wurden im Projekt *Systeme* für interaktive und personalisierbare *Lehrmaterialien* entwickelt und im Hochschulbetrieb eingesetzt.

Ein weiteres Ziel ist der Einsatz der im Rahmen des Projektes entwickelten Systeme und Lehrmaterialien an den beteiligten Hochschulen. Es wurde ein *Portal* geschaffen, welches auch nach Projektende den einheitlichen und projektübergreifenden Zugang zu den Lehrmaterialien ermöglicht (siehe unten).

Das Projekt legt besonderen Wert auf die Un-

terstützung einer einfachen Integration der Möglichkeiten neuer Medien in die bestehende Lehrpraxis. Aufbauend auf Ergebnissen von Vorprojekten wurde von vorhandenen, bewährten Kursmaterialien ausgegangen. Diese Materialien wurden in semantische Einheiten segmentiert, aus denen personalisierte Lerndokumente mit interaktiven und multimedialen Komponenten zusammengestellt werden können. Dies ermöglicht die individuelle und personalisierte Zusammenstellung von Lehrmaterialien und steht Lehrenden und Studierenden gleichermaßen zur Verfügung.

Im Rahmen des Projekts entstanden Lehrmaterialien, welche die Möglichkeiten des gedruckten Buches, der CD-ROM und des Internet optimal kombinieren. Multimediale und interaktive Komponenten werden gezielt dort eingesetzt, wo dies aus didaktischen Gründen oder zur Erreichung eines für die Lehre sinnvoll einsetzbaren Mehrwerts erforderlich ist. Die Komponenten greifen auf leistungsstarke Systeme aus den jeweiligen Domänen zu. Insbesondere sind hier Computeralgebrasysteme und automatische Theorembeeweiser zu nennen. Teilweise kann die Präsentation der Lehrinhalte nicht nur "oberflächlich" (für den Bildschirm oder für den Druck) optimiert werden, sondern wird auch inhaltlich mit vorhandenem Benutzerwissen abgeglichen.

1.3 *In2Math* Portal

Das *In2Math Portal* realisiert einen einheitlichen Zugang zu im Projekt entwickelten Lehrmaterialien über ihre jeweiligen Zugangssysteme. Dazu gehört auch die Lehrmaterialien übergreifende Stichwortsuche und Register. Während die vollen Möglichkeiten mit allen Interaktiven Elementen nur im Online Zugriff unterstützt werden, ist auch eine per CD vom Springer Verlag distributierte "Offline" Version des Portals erhältlich, welche die statischen Inhalte in PDF Form liefert. Die CD und auch der Online Zugriff können für Lehrzwecke kostenfrei verwendet werden. Siehe www.in2math.de für den Online Zugriff.

Im folgenden werden nun die Teilprojekte im Ein-

zelen vorgestellt (mit Ausnahme des Teilprojektes "Lineare Algebra Individuell" der Humboldt-Universität Berlin). Im Anschluß daran fassen wir die Erfahrungen im praktischen Einsatz zusammen.

2 Die ACTIVEMATH Lernumgebung

Innerhalb des *In2Math* Projektes wurde die web-basierte, benutzeradaptive Lernumgebung ACTIVEMATH weiterentwickelt, mit Mathematik-Inhalten gefüllt und in Lehrveranstaltungen an der Universität des Saarlandes und an der Hochschule für Technik und Wirtschaft in Saarbrücken getestet. Eine Demonstration des stabil laufenden Systems ist unter www.activemath.org/demo/ verfügbar. Das System läuft am besten mit den Browsern Mozilla und Netscape 6.1 aufwärts, da diese die Empfehlungen des W3C umsetzen und mathematische Formeln mit Unicode und MathML darstellen können.

ACTIVEMATH wurde erfolgreich auf verschiedenen internationalen Konferenzen und Messen präsentiert, darunter cebit, learnTec, education quality forum, eLearn, ITS und Online Educa.

2.1 Die Technologie

Architektur und externe Systeme. Die Architektur des Systems ist modular und die einzelnen Module, wie Kursgenerator, Inhaltsdatenbank und Benutzermodell haben wohldefinierte APIs und kommunizieren über XML-RPC. ACTIVEMATH bietet darüber hinaus eine Architektur zur Einbindung von externen Systemen für interaktive Übungen, die bisher für die Computeralgebrasysteme MuPad, Maple, Yacas und den Beweisplaner Omega umgesetzt wurde. Mit Hilfe dieser Systeme kann derzeit auch ein einfaches Feedback für die Lernende erzeugt werden. Bekanntlich ist das für das Lernen von großer Bedeutung. Ein Benutzer, der die Rolle eines Lehrers innehat, kann die interaktiven Übungen seiner 'Schüler' über eine Konsole beobachten und direkt Vorschläge machen.

Personalisierte Kursgenerierung aus semantischem XML. Personalisierte Materialien werden aus einzelnen Instruktionselementen erzeugt, die in einem XML-Format für Mathematik in einer Inhaltsdatenbank abgelegt sind und die mit Abhängigkeiten und anderen Metadaten annotiert sind. Zunächst werden die Inhalte zusammengestellt. Die Kursgenerierung ist abhängig von den Lernzielen des Benutzers, von dessen Präferenzen und Vorwissen sowie von dem gewählten Lernszenario. Danach erzeugt eine Präsentationskomponente mithilfe von Transformationen, u.a. XSL und CSS, eine Präsentation der Inhalte.

Die einzelnen Instruktionselemente enthalten Möglichkeiten sowohl für individuelle als auch offene Notizen des Benutzers (die als zugehörig zu den Inhaltselementen gekennzeichnet sind, aber getrennt gespeichert werden), für die Anzeige des Schwierig-

keitsgrades, Hilfe und für Copyright-Information, die wichtig für Web-Präsentationen ist.

Das Inhaltsverzeichnis des generierten Kursmaterials zeigt mittels einer Ampelmetapher den vermuteten Kenntnisgrad des Lernenden im Inhaltsverzeichnis an. Diese Annotation wird – entsprechend dem Lernfortschritt – dynamisch verändert.

Eine Menge sogenannter pädagogischer Regeln bilden eine pädagogische Strategie und diese wird bei der adaptiven Kursgenerierung abgearbeitet, d.h. die Regeln bestimmen benutzerabhängig und abhängig von der Wahl des Lernszenarios die Auswahl von Inhalten, Beispielen und Übungsaufgaben, sowie deren Anordnung. Diese Regeln verarbeiten die Informationen aus dem Benutzermodell [11].

Benutzermodellierung. Das Benutzermodell von ACTIVEMATH besteht aus

- den Präferenzen der Benutzerin (Studiengbiet, Verwendung von Tools, Bevorzugung von Erscheinungsbild des Systems, etc)
- Geschichte der Benutzeraktionen (und deren Erfolg)
- Datenbank mit sogenannten mastery-Werten der Konzepte, die in den relevanten Kursen angeboten werden.

Für die mastery-Werte werden aufbauend auf Resultaten aus der Lernpsychologie und Didaktik die drei Ebenen *knowledge*, *comprehension* und *application* unterschieden [3]. Sie werden im Gefolge von Lernaktivitäten wie Lesen oder Aufgabenlösen aktualisiert. Diese Werte können vom Lernenden inspiziert und modifiziert werden. Dadurch wirkt die Lernende selbstbestimmender und mitverantwortlich für den Lernprozeß.

Inhalte und Wissensrepräsentation. Die Repräsentation der Inhalte, die der Kursgenerierung zugrundeliegt ist die semantische XML-Repräsentation für Mathematik OMDoc [8]. Aufgrund unserer Erfahrung ist OMDoc in Zusammenarbeit mit der Carnegie Mellon University weiterentwickelt worden [4]. Insbesondere die Metadaten und die Repräsentation von Übungsaufgaben wurde beträchtlich erweitert [7]. Die für das Lernsystem sinnvollen und notwendigen Metadaten wurden erfaßt, sowohl die domänen-spezifischen als auch die pädagogisch-didaktischen Metadaten. Unsere Metadaten wurden mit den existierenden internationalen Standards verglichen (die zum einen um pädagogisch relevante Metadaten erweitert werden mußten, zum anderen in ihrer sehr technischen Breite nicht komplett verwendet wurden).

Das Lehrbuch 'Analysis Individuell' [5] wurde in OMDoc übertragen. Das war ein über das Projekt hinausgehend wichtiger Versuch, da viele Lehrmaterialien für Mathematik in LaTeX vorliegen. Die Erfahrungen aus der Übersetzung von LaTeX zu OMDoc belegen,

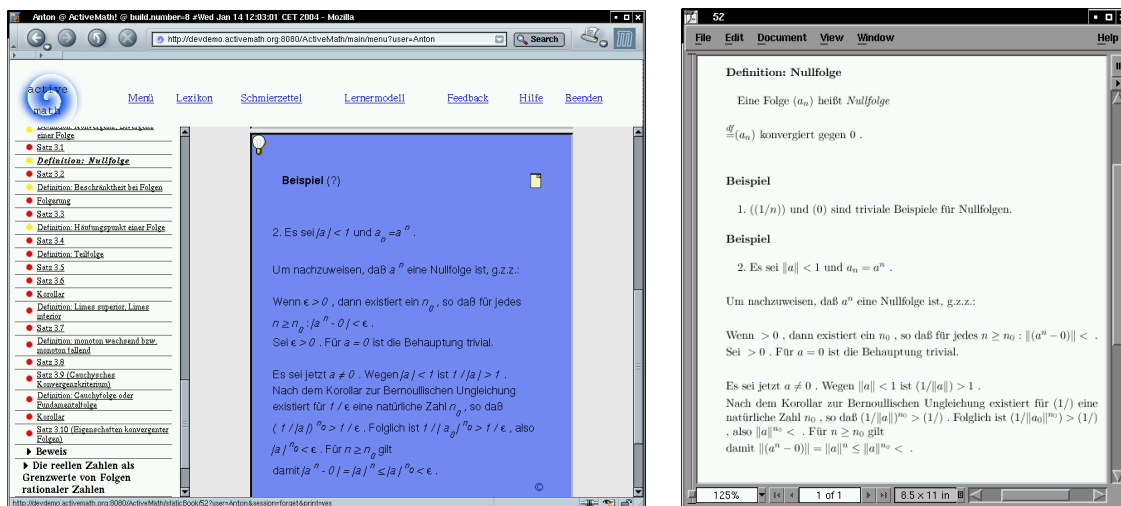


Abbildung 1: ActiveMath Präsentation der Lerninhalte in HTML (links) und PDF (rechts).

dass dies ein mit zu viel manueller Arbeit verbundener Weg der Inhaltserstellung ist, da jeder Autor seine eigenen Besonderheiten beim Schreiben von LaTeX hat und dadurch die benötigte Semantik der mathematischen Ausdrücke kaum automatisch produziert werden kann. Dasselbe trifft für eine Reihe der Struktur- und Metadaten zu, die nicht automatisch erzeugt werden konnten, wie etwa der Schwierigkeitsgrad einer Übung oder der Kontext eines Elements.

Ein Kurs über Statistik wurde innerhalb von In2Math für ACTIVE MATH erarbeitet, siehe Abschnitt 2.2. Außerhalb des Projektes wurden weitere Inhalte erstellt, die ebenfalls (teilweise) in der Demonstration im Netz verfügbar sind, u.a. Optimierung, Kombinatorik (von MathePrisma), Topology.

Adaptiver Vorschlagsmechanismus. Dieser bisher prototypische Mechanismus erzeugt dynamisch personalisierte Lernvorschläge zur Navigation, über zu wiederholende und zu vertiefende Inhalte und zu neuen Beispielen und Übungsaufgaben. Diese Vorschläge sind auf kognitionspsychologischen Grundlagen entwickelt worden. Die zugrundeliegende Architektur ist eine Blackboard-Architektur [9].

Formative Evaluation. Die formative Evaluation in Lehrveranstaltungen hat zu Verbesserungen des Systems einschliesslich der Hilfsfunktionen geführt. Ein Ergebnis war, dass Studenten neben den eigentlichen Inhalten auch das Wörterbuch und das Benutzermodell häufig nutzen, um mehr Informationen über Zusammenhänge von Konzepten bzw. eine (vom System produzierte) Abschätzung des Standes ihrer eigenen Fähigkeiten zu erhalten.

Da in den Lehrveranstaltungen das Problem auftrat, dass das Material zu umfangreich wurde, wenn der gesamte Inhalt für die Generierung zugrundegelegt wurde, wurde eine nur kapitelweise Zusammenstellung von Material implementiert.

Das Ziel eines iterativen Usability Tests war es, zur Optimierung und Qualitätssicherung durch softwareergonomische und instruktionspsychologisch-didaktische Untersuchungen beizutragen. Zu diesem Zweck wurde das Projekt durch Maßnahmen des Usability Engineerings und Designs begleitet. Zunächst lag der Fokus auf der Verbesserung der Benutzbarkeit des Systems, insbesondere im Hinblick auf Fragen der Handhabung und der Informationsdarbietung. Die in der ersten Phase gewonnenen Erkenntnisse flossen in die Überarbeitung von ACTIVE MATH ein. In der folgenden Iteration wurde das überarbeitete System erneut getestet, wobei dann verstärkt neue Komponenten des Systems und Fragen der Didaktik mitberücksichtigt wurden. Einige Details können dem Artikel [10] entnommen werden.

2.2 ActiveMath:Statistik

Innerhalb des In2Math Projektes wurde mit *ActiveMath:Statistik* ein Grundkurs zur Wahrscheinlichkeitsrechnung und mathematischen Statistik für ACTIVE MATH entwickelt, der an Fachhochschulen und Universitäten eingesetzt werden kann. Eine Demoversion ist unter markov.htw-saarland.de:8080/ActiveMath verfügbar. Die Materialien umfassen die folgenden Abschnitte:

- I. Beschreibende Statistik
- II. Grundlagen der Wahrscheinlichkeitsrechnung
- III. Einführung in die schließende Statistik

Neben den eigentlichen Lehrbuch-Inhalten entstanden eine Vielzahl interaktiver Elemente, insbesondere solche, die eine Schnittstelle von ActiveMath zu MuPAD und zu R verwenden. Diese interaktiven Elemente werden im folgenden noch beschrieben.

Die Materialien sind mit Metadaten versehen, welche den Schwierigkeitsgrad, den Abstraktheitsgrad,

das Anwendungsfeld, den Aufgabentyp bzw. den Inhaltstyp charakterisieren. Dadurch wird es möglich, analoge Inhalte unterschiedlich mit Definitionen, Beispielen und Aufgaben anzureichern und unterschiedliche Zielgruppen von Nutzern anzusprechen.

Technologie der Erstellung der Inhalte. Im Rahmen der Entwicklung des ACTIVEMATH Statistikurses wurde folgendermassen vorgegangen:

- Erstellung der Lehr-Texte und Konvertierung in das Omdoc-Format. Dafür kann der Editor “Oqmath” verwendet werden. Er bietet Vorlagen für die Metadaten und den eigentlichen Text an, die der Autor nur ausfüllen muss. Formeln werden textbasiert analog zu LaTeX geschrieben. Anschließend wird der Oqmath-Text automatisch in einen OMDOC-Text übersetzt.
- Festlegung der Metadaten zu jedem Item.
- Erstellung der Texte für das Lexikon. Neben den Texten bietet ActiveMath eine umfangreiche Lexikon-Funktion an. Der Autor kann dieses bei Bedarf zusätzlich mit Inhalt (Beispiele, Erklärungen usw.) füllen.
- Erstellung von Stylesheets. Für die Präsentation der Inhalte im Browser und die Konvertierung nach LaTeX werden Stylesheets benötigt. Die für einen Statistik-Grundkurs benötigten Stylesheets sind nun vorhanden, und stehen anderen Autoren zur Verfügung.
- Erstellung von Regeln für das Zusammenstellen von Inhalten nach Nutzerprofil und Wissensstand. Je nach Lernfortschritt werden durch ACTIVEMATH die Inhalte in unterschiedlicher Schwierigkeit, Abstraktheit und Vollständigkeit durch ActiveMath automatisch zusammengestellt. Dazu muss der Autor Regeln angeben, die in einer ”Regel”-Datei abgespeichert werden.
- Erstellung von Grafiken für die interaktiven Elemente.
- Erstellung von Autorenskripten zur Steuerung der interaktiven Elemente.

Interaktive Elemente. In *ActiveMath:Statistik* stehen eine Vielzahl abwechslungsreicher interaktiver Elemente zum Üben und Experimentieren zur Verfügung. Neben der Möglichkeit, individuell sowie Übungsaufgaben zu bearbeiten, wie der Student sie für seinen jeweiligen Wissensstand benötigt, werden die Studierenden zur aktiven und “motorischen” Auseinandersetzung angeregt, im Kontrast zu dem oft sehr passiven Verhalten in den Lehrveranstaltungen. Davon erwarten wir mehr Spaß beim Lernen und eine bessere Motivation für die mathematischen Fächer.

Interaktionsarten. Wir unterscheiden zwischen zwei Interaktionsarten:

1. “exercise”: Übungsaufgabe mit Rückmeldung und Bewertung.
2. “experiment”: Demonstration mit und ohne Interaktion / Übungsaufgabe mit und ohne Rückmeldung, in jedem Fall ohne Bewertung.

Jede der beiden Interaktionsarten umfasst verschiedene Interaktionstypen.

- MC (Multiple-Choice-Aufgaben)
- FIB (Fill-In-Blank-Aufgaben)
- Zuordnungsaufgaben
- Muplet (Text- und Grafik-Schnittstelle zu einem Computeralgebrasystem (MuPAD))
- Rlet (Text- und Grafik-Schnittstelle zu einem Statistik-Software-Paket (R))

ActiveMath:Statistik bietet darüber hinaus eine Vielzahl spezieller Applets für spezifische Fragestellungen und Demonstrationsziele.

Die Interaktionsart “experiment” dient in der Regel der (grafischen) Veranschaulichung, dem Experimentieren und der Simulation und soll den Studenten in die Lage versetzen, Hypothesen zu bilden, Schlüsse zu ziehen und diese experimentell zu überprüfen. Hierbei kommt verstärkt die Grafik-Schnittstelle von R zum Einsatz.

Die Interaktionswerkzeuge für Interaktionen vom Typ Zuordnung und die Schnittstelle zu MuPAD und R (Muplet und Rlet) wurden von der HTW des Saarlandes entwickelt. Charakteristisch für Muplet und Rlet ist das integrierte Hilfe- und Bewertungssystem, welches unten kurz beschrieben wird.

Für den Autor ist die Einbindung einer Interaktion in den Text sehr einfach. Er muss lediglich festlegen, welche Interaktionsart und welchen Interaktionstyp er verwenden will. Anschließend hat er ein Autorenskript zu schreiben, welches die konkrete Interaktion der gewählten Art steuert. Bei MC und FIB sind das Omdoc-Texte, die in die Omdoc-Dateien integriert werden. Bei allen anderen Interaktionsarten sind die Autorenskripte auf der Basis einer xml-basierten Skriptsprache zu entwerfen. Diese Skriptsprache ist sehr einfach gehalten, berücksichtigt, dass auch ACTIVEMATH-Laien interaktive Elemente entwerfen können, und ermöglicht den Einsatz der interaktiven Elemente auch außerhalb der ACTIVEMATH-Umgebung.

Schnittstelle zu R und MuPAD mit integrierter Hilfe und Bewertung. Bei Interaktionen zu statistischen Aufgaben kommt es im Gegensatz zu Aufgaben aus der Analysis oder der Algebra neben der Bewertung von Termumformungen insbesondere

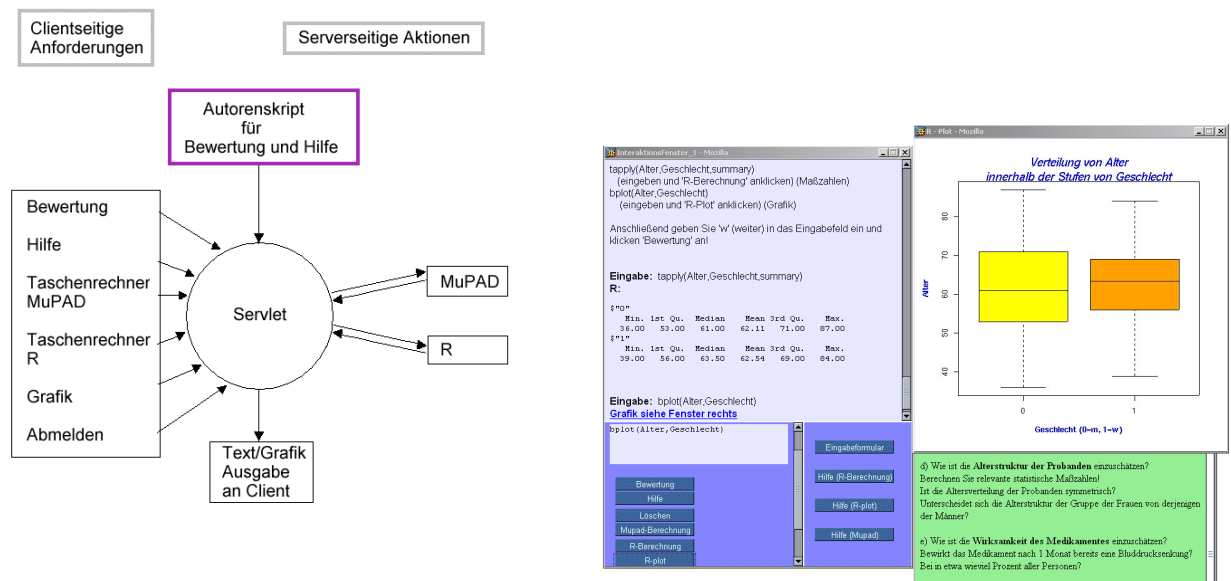


Abbildung 2: Links: Servlet-Architektur. Rechts: Aufgabe zur Datenanalyse unter Verwendung von R

darauf an, Datensätze für eine Berechnung bereitzustellen und verarbeiten zu können, aus statistischen Verfahren auswählen zu können, Grafiken dynamisch erzeugen zu können, komplexe statistische Untersuchungen bewerten zu können und Tabelleninhalte zu bewerten und zu verarbeiten.

Diese Anforderungen werden durch die von der HTW entwickelte Servlet-Schnittstelle zu R und MuPAD bewältigt. Die Architektur dieses Servlets ist in Abbildung 2 (Links) dargestellt. Abbildung 2 Rechts zeigt das User-Interface dieser Schnittstelle für eine Aufgabe zur Datenanalyse, die unter Verwendung von R zu lösen ist.

Auf der einen Seite dient diese Schnittstelle als Interface zwischen Client und R bzw. MuPAD. Diese Schnittstelle reicht clientseitige Anforderungen einfach an R und MuPAD weiter und gibt die Antworten von R und MuPAD an den Client zurück. Dadurch steht diese Schnittstelle als Taschenrechner bei der Lösung von Aufgaben zur Verfügung. Weiterhin werden Grafik-Befehle an R weitergereicht und die durch R erzeugten Grafiken an den Client gesendet.

Weiterhin gibt das Servlet auf der Basis eines steuernden Autorenskriptes Instruktionen zur Bearbeitung der Aufgabe sowie Hilfen und Bewertungen an den Client. Abbildung 3 zeigt das Userinterface des Servlets für eine Aufgabe zur Wahrscheinlichkeitsrechnung, die mit einer Hilfe und Bewertung hinterlegt ist. Das zugehörige Autorenskript ist links dargestellt. Das Autorenskript besetzt aus drei Teilen. Im startup-Teil werden u.a. die Aufgabe und die Lösung generiert, und allgemeine Instruktionen gegeben. Der Autor hat hier auch die Möglichkeit, bestimmte Befehle für R und MuPAD im Taschenrechner-Modus zu verbieten. Damit wird z.B. verhindert, dass Stu-

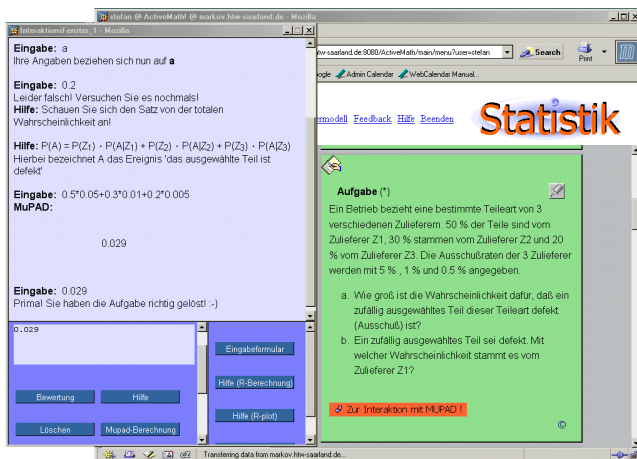
denten den MuPAD-Befehl `solve` verwenden, um die Aufgabe durch MuPAD lösen zu lassen. Es folgt der interaction-Teil, in welchem durch if-then-Regeln die Ein- und Ausgaben des Servlets an R, MuPAD und den Client gesteuert und verarbeitet und insbesondere die passenden Hilfen und Bewertungen erzeugt werden. Der letzte Teil ist der support-Teil, in dem Hilfen durch den Autor aufgelistet werden.

Das Autorenskript bietet u.a. darüber hinaus die Möglichkeit, der zufälligen Generierung beliebig vieler weiterer Übungsaufgaben des gleichen Typs, einschließlich der Lösungen, der Anzeige des bisherigen richtigen Lösungsweges, der Anzeige eines durch den Autor eingegebenen Lösungsweges, der Unterscheidung zwischen Syntaxfehlern, richtiger und falscher Umformung, des Erkennens, ob eine Aufgabe fertig bearbeitet wurde. Weiterhin ist die Definition von Niveaus bei der Bearbeitung der Aufgabe möglich, sowie das Anzeigen von Hilfen entsprechend dem erreichten Niveau.

3 Living Books

Im Rahmen des *In2Math* Projektes wurde an der Universität Koblenz-Landau *Living Book* entwickelt; Das *Living Book* Konzept meint personalisierte, benutzerorientierte Lehrmaterialien mit integrierten interaktiven Komponenten. *Living Book* unterstützt wichtige Teile des Grundstudiums aus dem Bereich der theoretischen Informatik. Das Hauptziel ist die Unterstützung aktiven, selbstbestimmten Lernens in Vorlesungen, Tutorien und im Selbststudium.

Ein wichtiger Aspekt bei der technischen Umsetzung war der Wunsch, die Vorteile gedruckter Bücher, wie gute typographische Qualität, mit der Flexibilität und Aktualität Web-basierter Techniken zu kombi-



```

<ervlet>
<startup> /* Anzeige von Informationen zur Aufgabe, Sperren von MuPAD-Befehlen */
/* Ermitteln der Lösung und Speichern in einer globalen Variablen */
<verbotten type="MuPAD">solve</verbotten>
<text>Geben Sie zunächst an, ob a oder b ein, nachdem, welche Teilaufgabe Sie bearbeiten wollen</text>
<var name="_LSG">0.029</var>
</startup>
<interaktion> /* Regeln zur Steuerung der Ein- und Ausgaben des Servlets an den */
/* Client-Browser und die externen Programme (MuPAD, R) */
<ip 0_EINGABE != a && 0_EINGABE != b </ip>
<then> <text> Sie müssen erst a oder b eingeben! </text></then>
<ip 0_EINGABE == a </ip> <then> <text> Ihre Angaben beziehen sich nun auf <b>a</b> </text> </then>
<ip 0_EINGABE == b </ip> <then> <text> Ihre Angaben beziehen sich nun auf <b>b</b> </text> </then>
<ip _TEILAUFGABE == a </ip>
<then>
<var name="_TEST"><mupad> is (round(0_EINGABE*10^3)/10.0^3 - _LSG) </mupad></var>
<ip _TEST == TRUE </ip> <then> <text> Prima, Sie haben die Aufgabe richtig gelöst! </text> </then>
<ip _TEST == FALSE </ip> <then> <text> Leider falsch! Versuchen Sie es nochmal! </text> </then>
<ip _TEST != TRUE && _TEST != FALSE </ip> <then> <text> Syntaxfehler! </text> </then>
</then>
</interaktion>
<support /* Angabe von Hilfetexten pro Teilaufgabe */
<teilaufgabe> <view>
<hile><text>Schauen Sie sich den Satz von der totalen Wahrscheinlichkeit an!</text></hile>
<hile><text> $P(A) = P(Z_1) \cdot P(A|Z_1) + P(Z_2) \cdot P(A|Z_2) + P(Z_3) \cdot P(A|Z_3)$ 
&mdot;  $P(A|Z_1) + P(Z_2) \cdot P(A|Z_2) + P(Z_3) \cdot P(A|Z_3)$ .
Hierbei bezeichnet A das Ereignis "das ausgewählte Teil ist defekt". </text></hile>
<hile><text>Es ist keine weitere Hilfe verfügbar!</text></hile>
</view></teilaufgabe>
<teilaufgabe> ... </teilaufgabe>
</support>
</ervlet>

```

Abbildung 3: Links: Aufgabe mit Bewertung und Interaktion mit MuPAD. Rechts: Autorenskript für Links

nieren. Die statische, sequentielle Struktur klassischer Textbücher wird durch die Möglichkeit der dynamischen Zusammenstellung von selektierten Inhalten übergangen. Damit können zum Beispiel nur solche Inhalte zusammengestellt werden, die zur Prüfungsvorbereitung für ein bestimmtes Themengebiet relevant sind.

Ein weiterer Aspekt von *Living Book*, auf den hier nicht weiter eingegangen wird, ist die *automatische*, aufgabenbezogene Zusammenstellung von Lehrmaterialien unter Berücksichtigung von Benutzerwissen (siehe [2]). Dieses Teilsystem kann auch vom ActiveMath System (Abschnitt 2) des Saarbrückener Projektpartners angesprochen und genutzt werden.

3.1 Die Technologie

Das *Living Book* System ist sehr allgemein angelegt. Im Koblenzer Teilprojekt wurde die Instanz "Logik für Informatiker" (Prof. Furbach) entwickelt. Die folgende Beschreibung der Hauptaspekte von *Living Book* orientiert sich deshalb daran.

Slicing Technology. Die Inhalte des *Living Book* sind als Sammlung elementarer Einheiten ("slices") organisiert. Die Zergliederung erfolgt bis auf Definition-, Theorem- und Aufgabenebene, etc. Weiterhin werden Metadaten verwendet um z.B. Typinformationen und verschiedene Abhängigkeiten zwischen den Einheiten zu beschreiben. Diese Metadaten zusammen mit vom Benutzer gewählte initiale Einheiten bilden die Grundlage zur Bestimmung personalisierter Lehrmaterialien. Verschiedene Benutzer können deshalb mit verschiedenen personalisierten Lehrmaterialien arbeiten. Auch kann ein und derselbe Benutzer verschiedene personalisierte Lehrmaterialien nach Bedarf unterhalten.

Diese Möglichkeiten basieren auf der *Slicing Information Technology (SIT)* zur Erzeugung und Verwaltung sich personalisierter Lehrmaterialien (siehe

www.slicing-infotech.de/en/index.php). Es handelt sich um eine Client-Server Architektur, wobei die Client-Seite durch einen üblichen Web-Browser gegeben ist. Im Rahmen des Projektes wurde die Serverfunktionalität erheblich erweitert. Dazu gehören "Kleinigkeiten" wie zum Beispiel eine Notizzettelverwaltung, in erster Linie aber Erweiterungen zur Möglichkeit der Integration interaktiver Elemente.

3.2 Interaktive Elemente

Das *Living Book* Konzept unterstützt exploratives Lernen. Damit ist gemeint, dass Benutzer die Möglichkeit haben, nicht nur Lehrmaterialien aus verschiedenen Blickwinkeln und unter verschiedenen Detaillierungsgraden einzusehen, sondern auch auf *interaktive Elemente* zugreifen zu können. Diese sind jeweils passend in den laufenden Text und in die Übungsaufgaben integriert. Für den Logikkurs wurden nicht nur bewährte Standardsysteme aus dem Bereich der Logik angebunden, sondern auch diverse interaktiven Elemente neu entwickelt. Die Flash-Sprache hat sich dabei als besonders günstig zur Realisierung hochinteraktiver Systeme herausgestellt. Ein wichtiger Aspekt ist, dass der Benutzer nicht die Eingabesprachen der verschiedenen Logiksysteme lernen muss; diese wird durch eine durchgängig verwendete, intuitive Eingabesprache abstrahiert. Das Bild 4 gibt einen Eindruck der Interaktionsmöglichkeiten in *Living Book*.

Momentan unterstützt *Living Book* die folgenden interaktiven Elemente:

- *Formel Datenbank:* Operationen zur Verwaltung einer momentanen Menge von Formeln (Eingeben, Löschen, Speichern etc).
- *Wahrheitstafel Generator:* Das System erzeugt zu gegebenen Formeln Wahrheitstafeln. Sie werden am Anfang des Kurses benötigt, in welchem Stu-

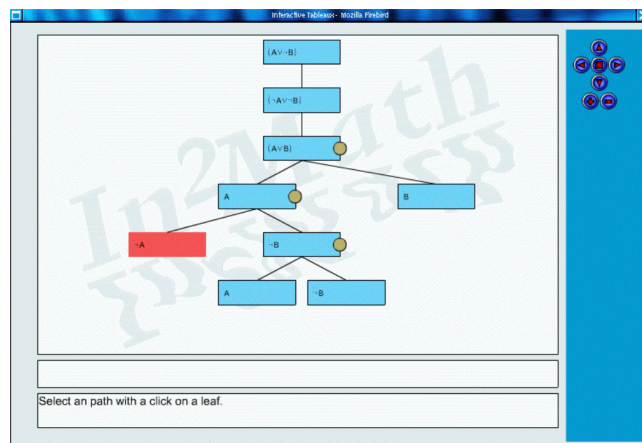
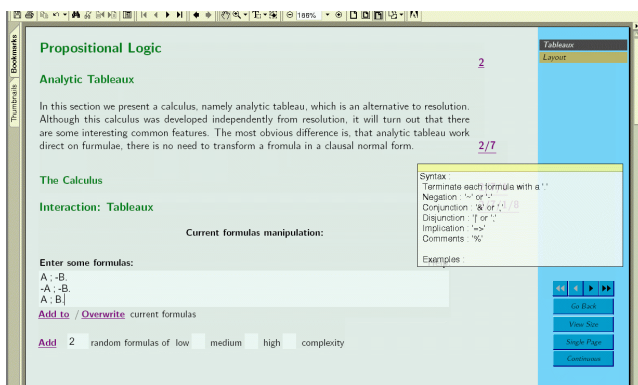


Abbildung 4: Links: Die PDF-Präsentation des Logikkurses. Rechts: Eine Flash-Interaktion in *Living Book*.

denen elementare Konzepte wie “Erfüllbarkeit” und “Gültigkeit” lernen.

- *Tableau Beweiser*: “Tableau” sind ein besonders anschauliches Beweisverfahren. Es stehen zwei Systeme zur Verfügung: das eine System arbeitet hoch interaktiv (vgl. Bild 4), und das andere liefert automatisch das Ergebnis. Sinnvolle Anwendungen beinhalten zum Beispiel das (semi-)automatische Lösen von Logik-Puzzles.
- *CNF Transformation*. Mit der CNF-Transformation werden die Formeln in eine äquivalente, syntaktisch einfachere Struktur gebracht, wie sie für die meisten automatischen Beweissysteme erforderlich ist.
- *Automatische Beweissysteme*. Diverse externe automatische Beweissysteme können direkt aus *Living Book* heraus angestoßen werden. Dies beinhaltet u.A. den von der Koblenzer Gruppe entwickelten Beweiser Protein [1], und den in das Saarbrücker MathWeb integrierten Beweiser Otter, welcher online, über eine Netzverbindung aufgerufen wird. Es stehen aber auch aus didaktischen Gründen “naive” Verfahren zur Verfügung.

Des weiteren wurden die Übungsaufgaben des Logikkurses in die Repräsentationssprache OMDoc übersetzt und in das System des Saarbrückener Projektpartners integriert. Somit stehen die Aufgaben auch dort zur Verfügung.

4 Miniprojekte

Für die Massenveranstaltungen *Mathematik für Informatiker* mit bis zu über 600 Teilnehmern werden seit einigen Semestern an der Universität Paderborn regelmäßig webbasierte Tools als Ergänzung zum klassischen Übungsbetrieb eingesetzt. Diese Tools erlauben zunächst die Anmeldung zu und die Verwaltung von Übungsgruppen. Während des laufenden Semesters ist es dann einerseits möglich, jedem Studierenden

individuell Aufgaben (“Miniprojekte”) zu den Vorlesungsinhalten zu stellen, die Lösungen per Webformular eingeben zu lassen und automatisch durch das angeschlossene Computeralgebrasystem MUPAD auf Korrektheit überprüfen zu lassen. Abgaben und Bewertung werden in einer Datenbank gespeichert, so dass die Daten letztlich bequem und flexibel für die Kontrolle des Leistungsstands, zur Notenvergabe, zur Erstellung von Statistiken etc. extrahiert und verarbeitet werden können.

Andererseits können mit diesen Tools Trainingseinheiten erstellt werden, die von den Studierenden unabhängig vom Übungsbetrieb zum freien und experimentellen Einüben mathematischer Inhalte eingesetzt werden können.

4.1 Studierendensicht

Der Studierende registriert sich auf einer Webseite und bekommt für die weitere Arbeit mit dem Online-Übungsbetrieb eine Identität und ein Passwort zugewiesen. Er kann dann im Laufe des Semesters vom Dozenten gestellte Aufgaben abrufen, bearbeiten und seine Lösung in einem Webformular abliefern. Die einzelnen Probleme gehören zu den in der Vorlesung vorgestellten Themenkreisen, die für alle Studierenden gleich sind, die Aufgaben werden jedoch für jeden Nutzer gemäß seiner Identität individuell während des Abrufs generiert. Durch die Zuteilung individueller Aufgaben wird das Kopieren von Übungsleistungen verhindert.

4.2 Dozentensicht

Ein neues Miniprojekt kann vom Dozenten bequem über ein Web-Interface erzeugt und installiert werden. Er hat dazu zwei MUPAD-Prozeduren `generiere` und `pruefe` zu definieren.

Fordert ein Studierender seine Aufgabe an, ruft das Übungstool `generiere(M)` mit der Matrikelnummer M des Studierenden auf. Die Prozedur hat dann eine individuell auf die Matrikelnummer zugeschnittene Aufgabe zu erzeugen, die als Liste von Zeichenketten (z.B.

```
// Miniprojekt 23
generiere := proc(Matrikelnummer) begin
  SEED:= Matrikelnummer:
  A:= matrix::random(3, 3, random(0..10));
  ["Berechne die Determinante der Matrix", A]
end:

pruefe := proc(Matrikelnummer, Soll) begin
  SEED:= Matrikelnummer:
  A:= matrix::random(3, 3, random(0..10));
  bool(Soll = linalg::det(A))
end:
```

| | |
|---|---|
| Miniprojekt 23 (Determinanten, 3 Punkte) | |
| Berechne die Determinante der Matrix | $\begin{pmatrix} 4 & 4 & 8 \\ 1 & 1 & 9 \\ 4 & 3 & 9 \end{pmatrix}$. |
| Aufgabe abgeben: | <input type="text"/> |

Abbildung 5: Eine webbasierte Übungsaufgabe. Links die Sicht des Dozenten, rechts die des Studierenden.

im \LaTeX -Format) und MUPAD-Ausdrücken zurückzugeben ist.

Nach Abgabe der Lösung durch den Studierenden wird die Prozedur `pruefe(M, Abgabe)` aufgerufen, welche die abgegebenen Daten auf Korrektheit überprüft und die Werte `TRUE` oder `FALSE` liefert. Bei unsinnigen Eingaben kann auch ein Fehler mit einem entsprechenden aussagekräftigen Text geworfen werden, mit dem der Nutzer auf die erwartete Form der Eingabe hingewiesen werden kann.

Diverse Einstellmöglichkeiten erlauben, die Laufzeit des Miniprojekts, die für eine korrekte Antwort zugeschriebene Punktzahl, die Anzahl der Abgabeveruche etc. festzulegen. Die Form der für die Studierenden sichtbaren Eingabefelder (Textfelder, Multiple-Choice etc.) kann eingestellt werden. Das Übungstool erlaubt neben generischem SQL-Zugriff auf die in der unterliegenden Datenbank gespeicherten Abgabedaten viele vereinfachte Zugriffsmöglichkeiten (Abfrageassistent) zur statistischen und visuellen Aufbereitung der Detaildaten einzelner Studierender oder der Daten der gesamten Studierendengruppe, zum Export für das Prüfungsamt oder die Tutoren der einzelnen Übungsgruppen. Es können zusätzliche Bonuspunkte etwa für die mündliche Mitarbeit in den Präsenztutorien eingetragen werden, Klausurergebnisse können übernommen und mit den Bonuspunkten für die Miniprojekte verrechnet werden.

4.3 Technisches

Die momentane Form des Übungstools basiert auf PHP-Skripten, die Ein- und Ausgabefelder auf Webseiten und den Zugriff auf eine unterliegende SQL-Datenbank verwalten. Die zentrale Rechenmaschine, die über PHP angesteuert wird, ist das Computeralgebrasystem MUPAD, welches einerseits die für jeden Studierenden individuell gelieferten Aufgaben erzeugt und andererseits die per Webformular abgelieferten Lösungen auf Korrektheit überprüft. MUPAD-Ausdrücke werden dabei als \TeX -Output geliefert, der dann nach automatischer Konvertierung als png-Graphiken in die Webseiten eingebunden wird.

5 Erfahrungen aus der Einführung in die Lehre

In diesem Abschnitt fassen wir die in den einzelnen Projekten gewonnenen praktischen Erfahrungen in der Lehre zusammen. Obwohl die einzelnen Projekte teilweise unterschiedliche Umsetzungen von “Interaktivität” realisieren, sind die Erfahrungen vergleichbar: eine generell positive Bewertung der Studierenden der neuen Medien, insbesondere der “Interaktionen”; Stabilität und leichte Benutzbarkeit des Systems als unabdingbare Voraussetzung für deren Akzeptanz; extrinsische Motivation (Punkte, Scheine) steigert die Bereitschaft der Studierenden zur Benutzung der Systeme; teilweise noch Verbesserungsbedarf bei den Autorentsystemen.

5.1 Die ACTIVE MATH Lernumgebung

Es hat sich wieder einmal bestätigt, dass die Motivation der Lernenden ein erstrangiger Faktor ist und wir müssen klar einschätzen, dass die Studenten in erster Linie durch Scheine, Punkte und Zensuren motiviert sind. Das wird von der Erfahrung belegt, dass dort, wo es nicht verpflichtend in einer Lehrveranstaltung eingebunden war und wo es keine Pluspunkte für das Lernen mit einem System gab, die interaktiven Übungen kaum genutzt wurden, sondern bestenfalls Texte vom Netz heruntergeladen und ausgedruckt wurden.

Dort wo eine Grundmotivation durch den Gesamtrahmen einer Lehrveranstaltung gegeben war, haben wir wiederholt die (nicht erstaunliche) Erfahrung gemacht, dass nicht nur interaktive Übungen beim Lernen halfen, sondern die besten Ergebnisse im Lernen von Mathematik durch eigene Beiträge der Studenten erzielt wurden. Das heisst, konstruktivistische Vorstellungen vom Lernen haben sich hier bewährt.

In einer *nicht* obligatorischen und *nicht* bewerteten Übung zur Vorlesung Mathematik für Informatiker wurde das online Material Analysis Individuell in einem realen Kontext verwendet. Das Material wurde mit interaktiven Übungen angereichert. Wegen des freiwilligen Charakters der Übungen war die Zusammensetzung der Teilnehmer nicht repräsentativ und nahm die Teilnehmerzahl während des Semesters ab (nach Aussagen der Studenten wegen der ansonsten zu hohen zeitlichen Belastung, die keine zusätzlichen

Lernaktivitäten erlaubte). Dahingegen wurde das Seminar 'Mathematik für Informatiker' sehr gut und mit großem Enthusiasmus besucht.

Ein anderes Resultat der Evaluation ergab, dass normale Lehrbuchmaterialien zu wenig geeignet sind für web-basiertes multimediales Lernen. Die ausschliessliche Präsentation von Lehrstoff bringt allein noch keinen Vorteil gegenüber traditionellem Lernen außer der Zeit- und Ortsunabhängigkeit. Dahingegen wirken Hypermedia, animierte Darstellungen, Beispiele, und Übungen sowie interaktive Übungen, in denen die Lernende verschiedene Lösungsschritte ausprobieren kann, lern- und motivationsfördernd.

5.2 ActiveMath:Statistik

Aus Autorensicht. Obwohl durch Oqmath relativ schnell Texte zu generieren sind, ist es für das volle Ausreizen der Funktionalität von ACTIVE MATH von Nachteil, dass keine Autoren-Beschreibung vorliegt. Weiterhin ist es für den Autor schwierig den Überblick über bereits vorliegende OMDOC-Elemente und Grafiken zu behalten, wenn der zu erstellende Inhalt sehr umfangreich wird. Wir schätzen es deshalb als sehr wichtig ein, eine Datenbank für die Verwaltung von Omdoc-Elementen einschließlich Grafiken aufzubauen, die z. B. die Suche nach vorhandenen Symbolen Grafiken und Identifiern zulässt.

Die Erstellung von Autorenkripten für alle Interaktionstypen wird technisch als problemlos beurteilt. Bei der Schnittstelle zu R und MuPAD (Servlet) wird die inhaltliche Seite der Angabe der passenden Hilfe und geeigneten Bewertung zum jeweiligen Schritt als Herausforderung angesehen, bei der in Zukunft Erfahrungen zu sammeln sind.

Aus Sicht der Studenten und Dozenten. Eine flächendeckende Evaluation von *ActiveMath:Statistik* hat bisher noch nicht stattgefunden. Für den Einsatz von *ActiveMath:Statistik* in Übungen gleichzeitig mit einer größeren Anzahl von Studierenden ist es erforderlich, die Stabilität des Systems zu verbessern. Eine Umfrage unter Studierenden, die *ActiveMath:Statistik* zum individuellen Training verwenden und unter Dozenten, ergab folgende Ergebnisse. Wünschenswert ist eine Einführung in ACTIVE MATH, um alle Leistungsmerkmale des Systems voll auszuschöpfen. Gut angenommen werden die verschiedenen Interaktionsmöglichkeiten, sie haben den Studenten Spaßgemacht und animierten zum Mitdenken. Besonders positiv wird die Schnittstelle zu R und MuPAD eingeschätzt, da sie alle zur Lösung einer Aufgabe nötigen Funktionen enthält, Interaktionen hoher Komplexität zulässt und die Möglichkeit der Generierung und Bearbeitung beliebig vieler Aufgaben bietet. Dozenten beurteilen besonders positiv, dass zu jeder zufällig generierten Übungsaufgabe auch der Lösungsweg mitgeneriert wird. Das entlastet sie von der Arbeit, immer wieder neu Klausur- und Übungsaufgaben erstellen zu müssen.

Weiterhin wird die Möglichkeit der Zusammenstellung von Büchern zu einem vorgegebenen Thema als sehr positiv eingeschätzt. Diese Funktion wird im Rahmen des Vorhabens, an der HTW ein Projektstudium einzuführen, noch an Bedeutung zunehmen. In jedem Fall wurde die Möglichkeit des Ausdrucks des zusammengestellten Büches von Seiten der Studierenden voll genutzt, was man sicher in Zukunft beim Einsatz von ACTIVE MATH an einer Hochschule durch den Dozenten steuern sollte.

5.3 Living Books

Living Book und die zu Grunde liegende Slicing Technology wurde in verschiedenen Kontexten evaluiert: Im EU-Projekt TRIALSOLUTION (siehe [6], im DFG Fokus Programm *Netzbasierte Kommunikation in Gruppen* (mit einem Fokus auf didaktische und pädagogische Aspekte), vor allem aber auch innerhalb des Koblenzer Teilprojektes. Das *Living Book* wird unter Ausnutzung aller Interaktionen regelmäßig an der Universität Koblenz-Landau innerhalb der Vorlesung "Logik für Informatiker" (Prof. Furbach) und im zugehörigen Übungsbetrieb eingesetzt.

Die Ergebnisse aus Studentensicht können folgendermassen zusammengefasst werden.

- Ungefähr die Hälfte der Studenten fand das System leicht zu benutzen. Die andere Hälfte hatte Schwierigkeiten oder gab keine Meinung an. Eine Einführung in die Benutzung des Systems scheint deshalb geboten.
- Zwei Drittel der Studenten fanden die Möglichkeit der personalisierten Dokumentzusammenstellung hilfreich. Man muss allerdings sehen, dass es sich um ein neues Konzept handelt, dessen Gebrauch noch nicht Allgemeingut ist. Die Studenten tendierten deshalb dazu, die Möglichkeiten nicht voll auszuschöpfen.
- Auch die Möglichkeit des selbstständigen, explorativen Lernens mit Interaktionen ist eher ungewohnt. Wir vermuten dass die Studenten im normalen Vorlesungs/Übungsbetrieb sehr starke Richtlinien bezüglich der Lehrinhalte und der verwendeten Vorgehensweisen erwarten. Dies hat sich auch im Übungsbetrieb bestätigt, wo Studenten mit Laptops zur Bearbeitung von Übungsaufgaben durch interaktive Elemente angehalten wurden.

Wir sind deshalb dabei, aufbauend auf die bisherigen im Großen und Ganzen positiven Erfahrungen, einen Logikkurs zu konzipieren, welcher sich organisatorisch stark von einer traditionellen Vorlesung unterscheidet und die Möglichkeiten von *Living Book* voll ausnutzt.

5.4 Miniprojekte

Seit dem ersten experimentellen Einsatz haben die diversen Entwicklungsstufen des Übungstools in meh-

renen Veranstaltungen mittlerweile weit über 100 000 individuelle Aufgaben erzeugt und bewertet. In den Massenveranstaltungen ist eine deutliche Korrelation zwischen den Erfolgsdaten der Webübungen und den Klausurergebnissen zu beobachten.

Die Reaktion der Studierenden auf die webbasierten Übungen ist durchweg positiv bis enthusiastisch. Speziell werden die kurzen Reaktionszeiten der Erfolgskontrolle sowie die örtliche und zeitliche Flexibilität bei der Bearbeitung der Übungen sehr positiv aufgenommen.

6 Abschließende Bemerkungen

Das Gesamtprojekt mit den dabei entwickelten Systemen, Lehrmaterialien und den gewonnenen Erfahrungen schätzen wir als sehr erfolgreich ein. Dies zeigt sich auch durch das Gesamtergebnis “Portal” (siehe Abschnitt 1.3), welches einen uniformen Zugriff auf die in den Teilprojekten erstellten Systeme realisiert. Auch die Online-Version des Portals wird über die Projektlaufzeit hinaus zur Verfügung gestellt und gewartet werden. Hier zeigt sich allerdings eine Problematik von der so mancher Teilnehmer am BMBF-Verbundvorhaben “Neue Medien in der Bildung” betroffen sein dürfte: die Finanzierung des nachhaltigen Einsatzes und die Pflege der Entwicklungen.

Literatur

- [1] Peter Baumgartner and Ulrich Furbach. PROTEIN: A *PRO*ver with a *Theory Extension* Interface. In A. Bundy, editor, *Automated Deduction – CADE-12*, volume 814 of *LNAI*, pages 769–773. Springer, 1994.
- [2] Peter Baumgartner and Ulrich Furbach. Automated Deduction Techniques for the Management of Personalized Documents. *Annals of Mathematics and Artificial Intelligence – Special Issue on Mathematical Knowledge Management*, 38(1), Kluwer Academic Publishers, 2003.
- [3] B.S. Bloom, editor. *Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain*. Longmans, Green, New York, Toronto, 1956.
- [4] J. Büdenbender, G. Gogvadze, P. Libbrecht, E. Melis, and C. Ullrich. Metadata in activemath. Seki Report SR-02-02, Universität des Saarlandes, FB Informatik, 2002.
- [5] B.I. Dahn and H. Wolters. *Analysis Individuell*. Springer-Verlag, 2000.
- [6] Diane Evans, Michael Armbruster, Ingo Dahn, Annette Pook, Aoife Brady, Britta Wienand, Alexander Dahn, Felix Golcher, and Karsten Tabelew. Trial-solution evaluation report. Deliverable D13, internal document, 2002.
- [7] G. Gogvadze, E. Melis, C. Ullrich, and P. Cairns. Problems and solutions for markup for mathematical examples and exercises. In A. Asperti, B. Buchberger, and J. H. Davenport, editors, *Proceedings of Second International Conference on Mathematical Knowledge Management, MKM03*, LNCS 2594, pages 80–93. Springer-Verlag, 2003.
- [8] Michael Kohlhase. OMDoc: Towards an internet standard for mathematical knowledge. In Eugenio Roanes Lozano, editor, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC’2000*, LNAI. Springer Verlag, 2001.
- [9] E. Melis and E. Andres. Global feedback in ACTIVEMATH. In *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (eLearn-2003)*, 2003.
- [10] E. Melis and M. Weber. Lessons for (pedagogic) usability design of elearning systems. In *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (eLearn-2003)*, 2003.
- [11] C. Ullrich. Pedagogical rules in ACTIVEMATH and their pedagogical foundations. Seki Report SR-03-03, Universität des Saarlandes, FB Informatik, 2003.