

E-Learning im virtuellen Informatik-Labor VILAB

Dr. Rainer Lütticke und Prof. Dr. Hermann Helbig
Fachbereich Informatik, Lehrgebiet Praktische Informatik VII
Intelligente Informations- und Kommunikationssysteme

Abstract: Durch den Lernraum Virtuelle Universität, der über das Internet erreichbar ist, konnte die Qualität der Fernlehre deutlich verbessert werden. Ein Bestandteil dieses Lernraums ist das virtuelle Informatik-Labor (VILAB). Studierende können sich von einem beliebigen PC über das Internet mit dem Labor-Server in der FernUniversität verbinden. In der virtuellen Lernumgebung VILAB werden den Studierenden Aufgaben mit verschiedenen Inhalten aus der Informatik angeboten. Die Studierenden können hier mit komplexen Software-Werkzeugen experimentieren und diese zur Lösung von didaktisch in ihrem Schwierigkeitsgrad gestaffelten Aufgaben einsetzen. Die Besonderheit ist dabei, dass sie während der Aufgabenbearbeitung und der Lösungsfindung durch eine interaktive tutorielle Komponente unterstützt werden.

I. Einführung

Das betreute Fernstudium bietet ein Höchstmaß an Unabhängigkeit und Flexibilität und eignet sich deshalb für Studieninteressenten, die sich aufgrund ihrer persönlichen Situation oder individueller Vorlieben an keiner Präsenzhochschule einschreiben können oder wollen. Zeit- und kostenintensiven Anreisen zu Präsenzphasen an der FernUniversität in Hagen (z.B. Praktikum oder Seminar) werden daher von vielen Studierenden eher als negativ angesehen. Andererseits bereitet einigen Studierenden das Selbststudium ohne Kontakt zu anderen Studierenden oder Feedback von Lehrenden gewisse Schwierigkeiten.

Mit der Leitentscheidung im Jahre 1999 zum flächendeckenden Einsatz neuer Medien und zur Entwicklung des Lernraums Virtuelle Universität (LVU¹) hat die FernUniversität in Hagen aber auf die Anforderungen der Studierenden nach weiterer Erhöhung der Flexibilität und neuen Modellen zur Qualitätssteigerung in der Fernlehre reagiert. Ein internes Förderprogramm (Innovationsfonds), das zur Weiterentwicklung des LVU seit 2000 eingesetzt wird, führt fortlaufend zu einer Verbesserung des virtuellen Studienangebots. Die dadurch gewonnene virtuelle Mobilität hebt die Begrenzungen von Zeit und Raum auf, so dass Präsenzphasen reduziert werden können. Der Online-Austausch mit anderen Teilnehmern und Betreuern garantiert die fachliche und soziale Einbindung und löst so die Isolation der Studierenden auf. Moderne Lerntechnologien und neue Lernangebote eröffnen Möglichkeiten, sich individualisiert und selbst gesteuert Wissen und Kenntnisse anzueignen.

Einen Baustein, der zu dieser Qualitätssteigerung in der Fernlehre geführt hat, bildet das virtuelle Informatik-Labor VILAB. Die Plattform dieses Labors (InfoLab) wurde im Rahmen des internen Förderprogramms 2001 entwickelt. Lehrinhalte für VILAB wurden innerhalb des Innovationsfonds vom Lehrgebiet Praktische Informatik VIII (Prof. Beierle, [BKW03]) und vor allem vom Lehrgebiet Praktische Informatik VI (Prof. Helbig) innerhalb des medin-Projekts² (Multimediales Fernstudium in der Medizinischen Informatik) erstellt. Medin wird durch das BMBF (Bundesministerium für Bildung und Forschung) innerhalb des Programms „Neue Medien in der Bildung“ gefördert (2001 – 2004) wird. Seit dem SS 2002 wird VILAB im Lehrbetrieb an der FernUniversität erfolgreich eingesetzt.

VILAB schafft eine virtuelle Laborumgebung, die es den Studierenden (ähnlich wie in anderen Naturwissenschaften in realen Labors) erlaubt, ihre in den Lehrveranstaltungen erworbenen theoretischen Kenntnisse in praktischen Versuchen zu erproben. Dazu werden den Studierenden in VILAB komplexe Softwarekomponenten angeboten, mit denen sie (Simulations-)Experimente sowie Eigen- und Weiterentwicklungen von grundlegenden Verfahren der Informatik durchführen können. Zusätzlich werden sie von einer interaktiven tutoriellen Komponente unterstützt, die durch die Analyse

¹ <http://www.fernuni-hagen.de/LVU/index.html>

² <http://www.medin.info>

von studentischen Lösungen und Begleitung der studentischen Arbeit im Labor hilfreiche Kommentare und Bewertungen liefert. Dadurch ergeben sich für die Studierenden mehrere Vorteile: Sie sparen die zeit- und kostenaufwendige Anreise für reale Praktika, sie können sich die Lernzeiten frei einteilen, der Transfer von theoretischem Wissen in praktische Anwendung wird gefördert ([MRGC95], [Ha97], [Ba00]) und ein zusätzlicher Lernerfolg durch den motivationsfördernden Einsatz interaktiver Elemente, besonders der tutoriellen Komponente, stellt sich ein.

II. Arbeiten in VILAB

VILAB ist nach einem Client-Server-Modell aufgebaut, das sich im Wesentlichen aus einem in der Universität betriebenen Lehrserver und den Windows- oder Linux-basierten PCs als Client bei den Studierenden zusammensetzt (Abb. 1).

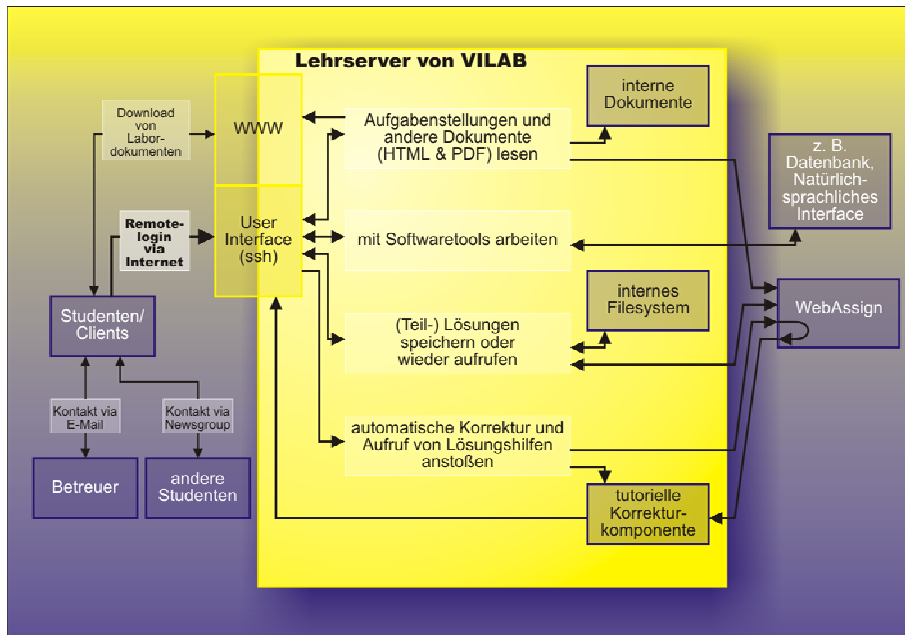


Abb. 1: Aufbau des VILAB-Systems

Der Zugang der Studierenden zu dem System über ihre eigenen PCs hat den Vorteil, dass sie jederzeit selbständig von zu Hause oder von anderswo mit dem System arbeiten können. Die Herstellung der Verbindung zwischen dem Lehrserver und den PCs geschieht über das Internet. Um den Studierenden den Zugang zum Labor technisch zu ermöglichen, brauchen die Benutzer-PCs nur mit Standard-Hardware ausgestattet sein. Neben einem Browser als Nutzerinterface gibt es als zweite Schnittstelle zwischen Laborserver und den PCs der Studierenden ein Navigationstool, dessen Prozesse auf dem Server ablaufen, aber dessen Grafik auf den PCs abgebildet wird. Ein Remote-login auf den Laborserver führt dabei zu einem automatischen Start des Navigationstools (Abb. 2). Dieses Login ist für Studierende, die bereits Linux einsetzen, trivial. Windows-Benutzer müssen zusätzlich die Open-Source-Pakete *cygwin* und *XFree86 4.x* installieren. Mit dieser Software kann eine entsprechende Unix-Funktionalität und der Zugang zum Server hergestellt werden. Zu diesem Zweck wurde eine CD-Distribution mit den o.g. Softwarepaketen speziell für VILAB konzipiert und zusammengestellt, die bei Bedarf an Studierende ausgegeben wird, so dass eine einfache Installation gewährleistet ist. Der Nachteil einer möglichen notwendigen Software-Installation für Windows-Nutzer wird aber durch die Vorteile eines Remote-logins mehr als aufgewogen. Durch den Verzicht auf die Implementierung der Benutzeroberfläche innerhalb eines Internetbrowsers ergeben sich umfangreichere technische und inhaltliche Gestaltungsmöglichkeiten für das Labor, die Performanz und Darstellungsqualität ist wesentlich besser, die Programmierung von speziellen Java-Applets ist unnötig, Software-Werkzeuge können leicht in das Labor eingebunden werden und vor allem ist die Reaktionszeit des System durch den gewählten Zugang sehr kurz.

Zur Anzeige der Laborinhalte öffnet der Nutzer auf seinem Rechner einen Browser mit einer ihm dauerhaft zugeteilten individuellen URL (Abb. 2). Über das Navigationstool sind Software-Werkzeuge, Seiten aus dem WWW sowie lokale PDF- und HTML-Dokumente ansteuerbar (Abb. 1). Die Prozesse der Software-Werkzeuge werden dabei vom System überwacht. Aktionen im Navigationstool, die zur Anzeige von Dokumente führen sollen, resultieren in einer automatischen Aktualisierung der Seite im Browser. Dies wird dadurch realisiert, dass die zentral im System abgelegten lokalen Dateien in das individuelle Directory des Nutzers in VILAB geladen und zur Anzeige auf die Datei der entsprechenden URL des Nutzers kopiert werden. Durch PHP-Skripte erkennt der Browser eine Veränderung des Inhalts unter der unveränderten URL und stößt ein automatisches Reload der Seite an. Auf diese Weise ist es möglich, dass der Server den Browser auf dem PC des Studierenden indirekt steuert. Externe Dokumente aus dem Internet werden in einen Frame eingebettet, so dass auch dort übergeordnete PHP-Skripte den Inhalt des Browsers unter der URL des Nutzers kontrollieren können. Datenanzeige und das Speichern von Dateien werden für jeden Nutzer individuell gemanagt, so dass das Labor von viele Studierende gleichzeitig verwendet werden kann. Da die Nutzer während ihrer Arbeit im Labor permanent online sein müssen, können sich die Studierenden alle im Labor aufrufbaren Dokumente auch über eine gewöhnliche URL zur offline-Nutzung herunterladen (Guided Tour³).

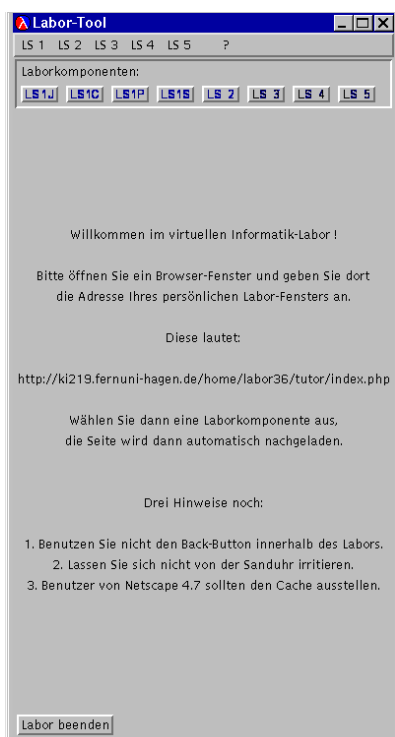


Abb. 2: Startansicht des Labortools

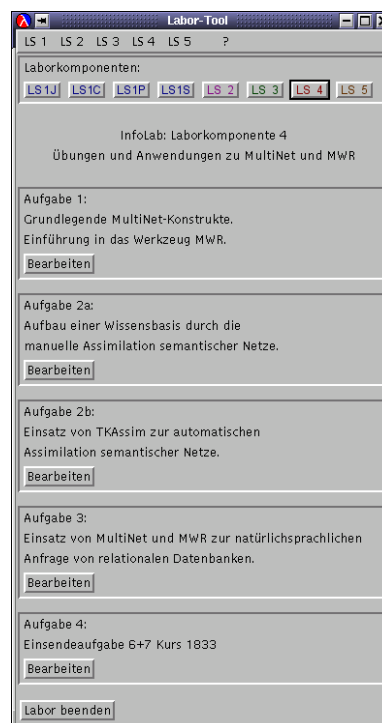


Abb. 3: Ansicht des Labortools
(in Laborstation 4)

Server-seitig besteht das System aus einem Kern auf Linux-Basis, der so konzipiert ist, dass die Studierenden weder gewollt noch ungewollt nach dem Remote-login mit dem unterliegenden Linux-System in Berührung kommen. Der Server speichert und verwaltet die Laborinhalte und bietet diese über das Navigationstool (im Lisp-Dialekt Scheme geschrieben) den Nutzern an (Abb. 2 und 3). Der Systemkern verfügt außerdem über eine tutorielle Komponente (in Java geschrieben), die automatisch bei dem Remote-login gestartet wird und interaktiv Hilfestellungen zu der gerade bearbeiteten Aufgabe abgeben kann (s. Kap. IV). Die genauen inhaltlich-didaktischen Führungsalgorithmen und Kontrollmechanismen für eine konkrete Laborstation werden allerdings in letzterer zur Verfügung gestellt [LGH02]. Der Plattform-Kern verwaltet auch die in das Lernsystem integrierten Werkzeuge und stellt die Kommunikations-Infrastruktur zu den übrigen Plattform-Komponenten sowie zu

³ <http://pi7.fernuni-hagen.de/vilab/tour/>

dem Online-Übungssystem WebAssign⁴ [BHSV99] bereit (Abb. 1). WebAssign ist dabei so in VILAB integriert, dass der Nutzer nicht merkt, bei welcher Aufgaben er sich im WebAssign-System befindet.

III. Laborstationen und Lehrinhalte

Der gesamte Lehrinhalt ist in verschiedene Laborsäulen mit mehreren Aufgaben unterteilt, die teils aufeinander aufbauen bzw. aufeinander Bezug nehmen und teils eigenständig sind. Das Spektrum reicht von einfachen Aufgaben, die das Verständnis von Grundlagen trainieren und überprüfen, bis hin zu sehr schwierigen und komplexen Aufgaben, die in einem fortgeschrittenen Stadium des Studiums zur Vertiefung der jeweiligen Lehrinhalte dienen und vor allem in Fachpraktika eingesetzt werden. Jede Laborstation besteht aus einer einführenden Seite, die über Ziele, Einsatz, Wissenshintergrund, Einbettung der Lerninhalte in einen größeren Kontext informiert und einen Überblick über die Aufgaben der Laborsäule gibt. Die Aufgaben sind wiederum in Teilaufgaben untergliedert (Abb. 3 und 4) und geben den Studierenden Aufschluss über Lernziele, Aufgabenstellung, Wissenshintergrund, benötigte, hilfreiche oder weiterführende Literatur, einzusetzende Softwarewerkzeuge, Charakteristiken der Lösung und über die Art der Kontrolle/Hilfe der tutoriellen Komponente (s. nächstes Kapitel). Die Studierenden können dabei aufgefordert werden, ihre Lösungen in Form von Standard-Tests (z.B. Multiple-Choice, Zahlabfragen, Zuordnungen, etc.), in Textfeldern innerhalb eines Software-Werkzeugs oder der Aufgabenstellungsseite (so bei Programmcode, SQL-Anfragen, Regeln, etc.) oder in Form einer Grafik (so bei Semantischen oder Neuronalen Netze) einzugeben.

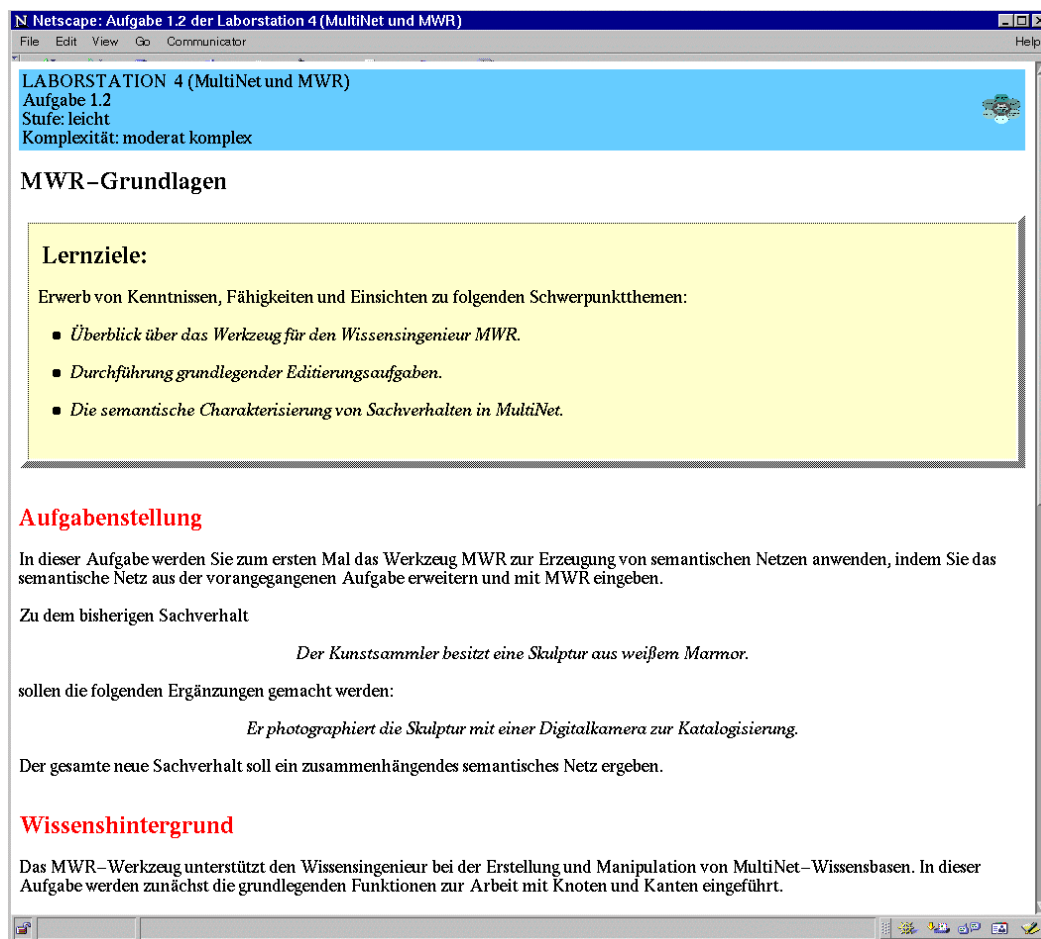


Abb. 4: Aufgabenstellung (aus Laborstation 4)

⁴ <http://niobe.fernuni-hagen.de/WebAssign/>

Aus den oben aufgeführten Beispielen lässt sich bereits erklären, welche Lehrinhalte in den einzelnen Säulen vermittelt werden:

- Grundlegende Paradigmen der Programmierung: C (prozedural), Java (objektorientiert), Scheme (funktional) und Prolog (logisch)
- Neuronale Netze
- Datenbankabfragen in SQL
- Automatische Sprachverarbeitung
 1. Semantische Netze (MultiNet [He01])
 2. Inferenzbasierte Methoden
 3. Transformation natürlichsprachlicher Ausdrücke in SQL
- Computerlinguistik

Für die semantische Repräsentation natürlichsprachlicher Informationen wurde ein grafisches Werkzeug zur Wissensrepräsentation und -verarbeitung (MWR [Gn00], Abb. 5) in VILAB integriert. Zur Darstellung Neuronaler Netze und zum Experimentieren mit denselben wurde der Stuttgarter Neuronale Netze Simulator (SNNS⁵) in VILAB integriert.

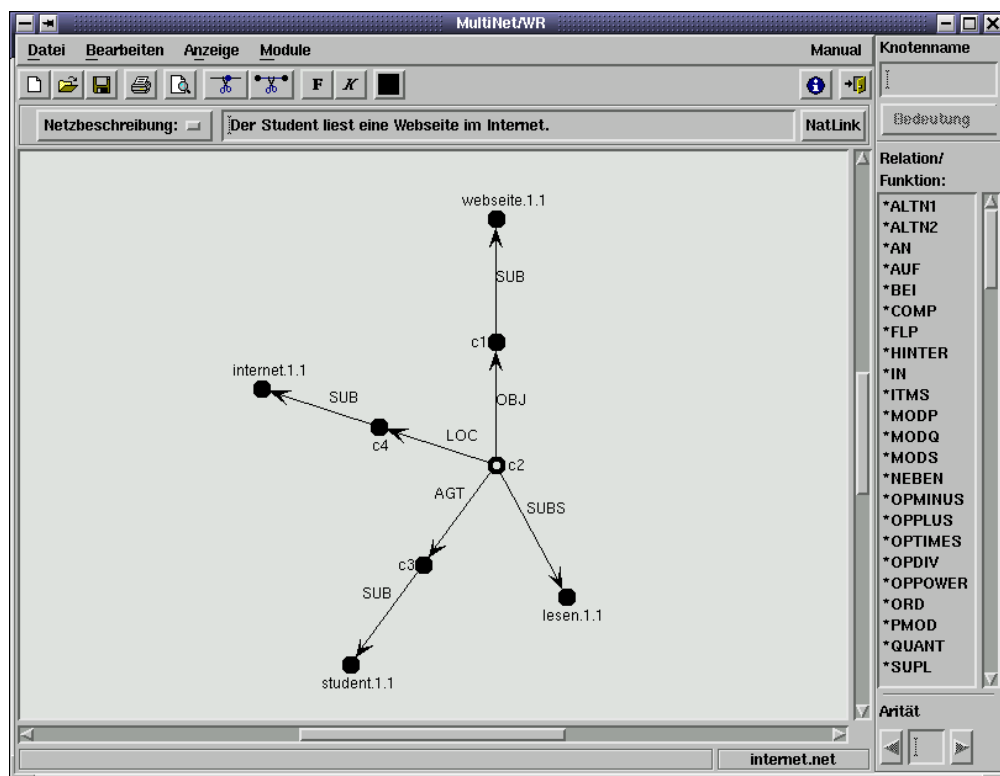


Abb. 5: Das MWR-Tool

Bei der Konzeption des Kernsystems wurde besonderes Augenmerk darauf gerichtet, die modulare Erweiterbarkeit des virtuellen Labors zu unterstützen, damit aufbauend auf der entwickelten Plattform zukünftig weitere Laborstationen mit neuen Lehrinhalten und neu zu implementierenden, aber bereits bestehenden Softwarewerkzeugen realisiert und in VILAB integriert werden können. Mit Lehrstühlen an der FernUniversität in Hagen, der TU München, der Medizinischen Universität zu Lübeck und der RWTH Aachen wird diesbezüglich bereits kooperiert. Zur einfacheren Erstellung von Lehrinhalten wird zurzeit ein Browser-basiertes Autorentool entwickelt, das die Lehrenden bei der Erstellung von Laborstationen, Aufgabenstellungen und Hypertext-Dokumenten für das tutorielle Feedback (s.u.) unterstützt und dem Administrator von VILAB die Einbindung der neuen Lehrinhalte erleichtert.

⁵ <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

IV. Interaktive tutorielle Komponente

Die Studierenden werden während ihrer Laborarbeit durch eine interaktive tutorielle Komponente unterstützt. Zu den Hauptaktivitäten dieser Komponente gehören die beiden Aktionen (Abb. 1):

- Entgegennahme einer eingesendeten studentischen (Teil-)Lösung mit Analyse und Bewertung oder Zwischenstandsmeldung
- Übermittlung eines Fehlerzustandes, zu dem erklärende Informationen angezeigt werden

In der Literatur (z.B. [Br99]) wird die erste Korrekturmethode als passiver und die zweite als aktiver Modus beschrieben.

In dem **aktiven Modus** reagiert die tutorielle Komponente sofort und automatisch auf entdeckte Fehlerzustände. Er ist vor allem für Laborkomponenten gedacht, die weitgehend autark mit dem Benutzer interagieren (auch externer Modus genannt [LGH02]). Ein Beispiel für eine derartige Komponente ist das grafische Werkzeug zur Wissensrepräsentation (s. Kap. III), das über eine eigene Benutzeroberfläche zur Anzeige und Manipulation der Wissensbasis verfügt. In diesem Fall findet die Interaktion mit dem Benutzer durch die Komponente der Lernumgebung selbst statt. Interaktive Problemunterstützungen können hier so ausgestaltet sein, dass gewisse Aktionen im Software-Werkzeug, die als falsch erkannt werden, sofort durch erläuternde Fehlertexte in Fenstern oder durch farbliche Änderungen im Werkzeug angezeigt werden (z.B. rote Farbe für falsche Elemente).

In dem **passiven Modus** wird die tutorielle Komponente erst auf studentische Aufforderung hin aktiv. Diese Aufforderung, verbunden mit einer nachfolgenden intelligenten Analyse der studentischen Lösung, kann entweder in autarken Laborkomponenten oder innerhalb der Aufgabenstellungen (interner Modus) angestoßen werden. Ist die Analyse abgeschlossen, wird ein Fehlercode und eine dynamisch erzeugte Datei mit Ergebnissen der Analyse erzeugt. Die tutorielle Komponente ist mit Hilfe einer Datei vorkonfiguriert, in der für Fehlercodes die vom Tutor zu ergreifenden Aktionen (z.B. Fehlertexte anzeigen) angegeben sind. Aus dem Fehlertext und der Datei mit den Ergebnissen der Fehleranalyse wird nun mittels PHP, Javascript und HTML dynamisch die Feedback-Datei erzeugt, die individuell auf den Fehler des Nutzer abgestimmt ist. Diese wird in das individuelle VILAB-Directory des Nutzers geladen und wie oben beschrieben im Browser angezeigt.

Die intelligente Analyse stellt eine hohe Herausforderung an die Autoren von Aufgabenstellungen dar, da sie für die Qualität der Interaktivität und somit indirekt auch für den Lernerfolg verantwortlich ist. Diese Analyse und damit verbundene Prozesse (z.B. Compilieren von Programmen) kann entweder in die Software-Werkzeuge implementiert werden, komplett auf dem VILAB-Server stattfinden, von einem sogenannten "Korrektur-Server" des WebAssign-Systems durchgeführt werden (nur für Standardaufgaben sind Korrektur-Server vorhanden), oder speziell programmierte Korrektur-Servern leiten die studentischen Lösungen in das VILAB-System, in dem dann die Korrekturanalyse erfolgt.

Inhaltlich besteht das Feedback einer Fehleranalyse aus den drei Teilen Fehlerdiagnose, Fehlerkorrektur und Leistungsbewertung (Abb. 6), die jeweils nach Aspekten für motivationales Feedback gestaltet sind [Mu00]. In der Fehlerdiagnose wird nur kurz auf den Grad der Richtigkeit der Lösung eingegangen, während die Fehlerkorrektur alle Hinweise und Hilfen enthält. Diese können in den einfachsten Fällen kommentierte Musterlösungen und das Aufzeigen von Syntaxfehlern sein. Durch komplexere Testverfahren, automatische Vergleiche mit Musterlösungen und Inferenzverfahren aus der künstlichen Intelligenz können aber auch einzelne Fehler herausgefunden werden. Abhängig von den Fehlern können dann Fehlerlisten, Beschreibung und Erklärungen von Fehlern, Hilfestellungen zur Vermeidung und Verbesserung von Fehlern, passende Links zu Lehrtexten und Literatur oder grafische Hinweise in den Software-Werkzeugen angegeben werden. Des Weiteren sind auch Hilfen in Form von Beispielen und Wegen zur Lösungsfindung möglich. Entsprechend der Analyse wird auch eine kurze Leistungsbewertung abgegeben, die besonders für die Vermittlung motivationaler Aspekte des Feedbacks genutzt wird.

Nach dem Grad und der Funktionsweise an Interaktivität und der Fehlerdiagnose/-korrektur beim Aufgabenlösen kann VILAB somit als eine „Intelligent Learning Environment“ betrachtet werden (nach der Definition von Dillenbourg et al. [DHM93]).

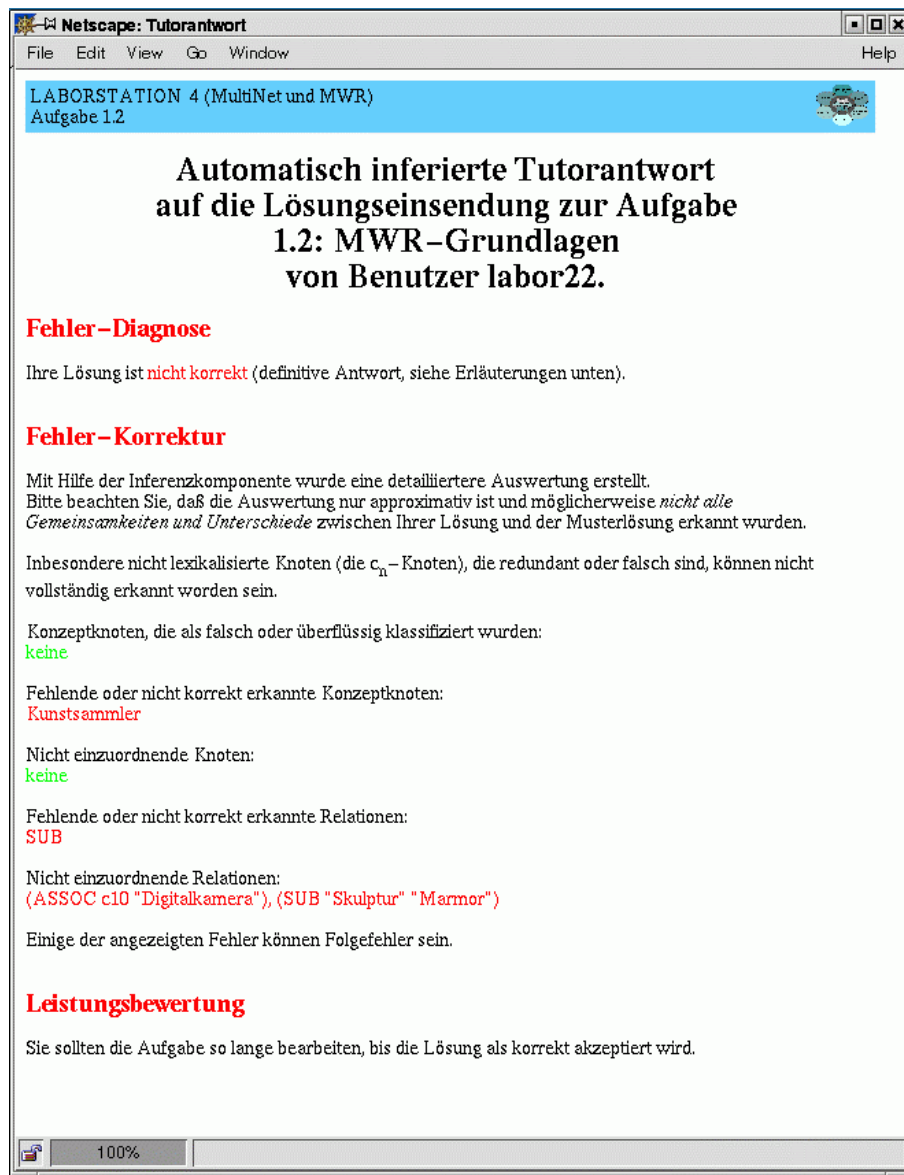


Abb. 6: Tutorantwort zur Aufgabe 1.2 in Laborstation 4

V. Curriculare Einbettung und erste Erfahrungen

Seit SS 2002 wird VILAB in verschiedenen Bereichen an der Fernuniversität in Hagen in der Informatik-Lehre eingesetzt. In einem Seminar dient es als Hilfestellung und Vertiefung des jeweiligen Seminararbeitsthemas. Der Einsatz von VILAB in Kombination aus zwei Präsenzphasen (Abb. 7) und Heimarbeit im Rahmen dieses Seminars realisiert somit die Lehrform des Blended Learnings. Da zwar die Belegung eines Seminars (Bachelor) bzw. von zwei Seminaren (Diplom) zum Pflichtprogramm eines Studierenden an der Fernuniversität gehört, aber mindestens sechs Seminare pro Semester zur Auswahl stehen, kann der Studierende aussuchen, welchen multimedialen Grad sein Seminar haben soll: klassisch, blended learning oder komplett virtuell. In einem Fachpraktikum konnten Studierende optional Teillösungen von Programmen durch die automatische Tutor-Komponente überprüfen lassen. In einem weiteren Fachpraktikum werden die Studierenden zu verschiedenen Laborsäulen geführt, um (ähnlich wie in den Naturwissenschaften) mehrere Aufgaben praktisch zu bearbeiten. Im Informatik-Studium gehört mindestens ein Fachpraktikum zum Pflichtprogramm und auch hier gibt es Wahlmöglichkeiten zwischen verschiedenen Praktika (zwei pro Semester, von Semester zu Semester wechselnd), die unterschiedlich stark multimedial ausgerichtet sind. Des Weiteren wird das Labor in einem Kurs von den Studierenden zur Unterstützung bei der Lösung von komplexen Übungsaufgaben und für Selbsttestaufgaben eingesetzt. Außerdem wird VILAB eingebunden von

Lehrveranstaltungen als Experimentierfeld und zur Prüfungsvorbereitung verwendet und die Softwarewerkzeuge kommen im Labor bei Diplomarbeiten zum Einsatz.



Abb. 7: Präsenzphase im WS 02/03

Eine Evaluation zeigt positive Ergebnisse. Eine Befragung der Studierenden ergab, dass die Windows-Nutzer (85%) die zusätzliche Software problemlos installieren konnten. Die Kommunikation aller Betriebssysteme (Windows-Versionen und Unix-Derivate) mit der Plattform funktionierte einwandfrei. Die Reaktionszeiten der Software-Tools und des PHP-gesteuerten Browsers auf dem studentischen PC waren sehr gut (selbst mit 56k-Modem). Die Navigationsmöglichkeiten wurden als gut bewertet, ebenfalls auch die Struktur der Aufgabenstellungen. Besonders die tutorielle Komponente hat durch das schnelle Feedback geholfen und motiviert. Weitere positive Aspekte waren: abwechslungsreiche Aufgabenstellungen, interessantes Arbeiten mit komplexen interaktiven Werkzeugen und die Möglichkeit, sich Lösungen „spielerisch“ durch trial-and-error zu erarbeiten. Objektiv zeigte sich, dass die Aufgabenlösungen durch das tutorielle Feedback überwiegend sehr gut waren und viel besser als die Lösungen ohne tutorielle Unterstützung. Außerdem zeugten die Seminarvorträge davon, dass die Vertiefungsmöglichkeiten zu einem Thema zu einem besseren Verständnis führten. Äußerst beachtenswert ist die Tatsache, dass die Quote der erfolgreichen Kursteilnehmer deutlich gestiegen ist (95% gegenüber sonst ~50%) und 25% der Seminarteilnehmer nun an einer Diplomarbeit aus den in VILAB angebotenen Themenbereichen oder zu VILAB selbst arbeiten bzw. mit Erfolg gearbeitet haben.

Es zeigte sich aber auch, dass selbst bei Informatik-Studierenden eine Hemmschwelle zur Nutzung von VILAB überwunden werden musste. Die Sorge vor großer Einarbeitungszeit und Aufwand bei der Installation der Software-Pakete für Windows-Nutzer ist jedoch unbegründet, wie Befragungen von VILAB-Nutzern ergeben haben. Außerdem wird zunächst nicht an die große Unterstützungsleistung der tutoriellen Komponente geglaubt, die aber die Studierenden nach der Nutzung von VILAB subjektiv empfunden haben und die objektiv festgestellt werden konnte.

VI. Ausblick

Die Weiterentwicklung des Labors zielt vor allem auf die Verfeinerung der tutoriellen Komponente sowie deren Einsatz in neuen Laborsäulen, so dass sich die Einsatzmöglichkeiten von VILAB verbreitern. Dazu bestehen neben den schon erwähnten Kooperationen (s. Kap. III) eine weitere mit der Universität Erfurt, um neuste Ergebnisse der Mediendidaktik in VILAB anwenden zu können. Beson-

ders durch die zukünftige Integration eines Studentenmodells (Vorarbeiten laufen bereits) werden sich noch bessere Reaktionsmöglichkeiten für die tutorielle Komponente ergeben. Außerdem sollen kooperative Software-Werkzeuge eingebunden werden, damit das psychologische Bedürfnis nach sozialer Eingebundenheit besser befriedigt werden kann [DR85], denn bisher sind nur Newsgroup (unter den Studierenden) und E-Mail-Verkehr (zwischen Lehrendem und Lerner) als Kommunikation außerhalb des Dialoges mit dem Lernsystem im Einsatz. Zu diesem Zweck wurde eine Kooperation mit dem Lehrgebiet Praktische Informatik VI (Prof. Haake) begonnen, um die im Innovationsfonds 2003 entwickelte Plattform CURE (Collaborative Universal Remote Education) auch innerhalb von VILAB nutzen zu können.

Um die Nachhaltigkeit von VILAB sicherzustellen, wird das Labor nach der Entwicklungsphase von einer zentralen Stelle im Fachbereich Informatik administriert werden.

Insgesamt ist VILAB sehr gut von den Studierenden angenommen worden, so dass der eingeschlagene Weg mit großer Zuversicht weiterverfolgt wird.

Literaturverzeichnis

- [Ba00] S-P. Ballstaedt et al. *Planung, Entwicklung, Durchführung von Fernstudienangeboten*. Deutsches Institut für Fernstudienforschung an der Universität Tübingen, 2000
- [BKW03] C. Beierle, M. Kulas und M. Widera. Automatic Analysis of Programming Assignments. In *Proc. der 1. e-Learning Fachtagung Informatik (DeLFI)*. A. Bode, J. Desel, S. Ratmayer, M. Wessner (Hrsg.). Lecture Notes in Informatics. P-37. Seiten 144-153, 2003
- [BHSV99] J. Brunsmann, A. Homringhausen, H.-W. Six und J. Voss. Assignment in a Virtual University – The WebAssign-System. In *Proc. 19th World Conference on Open Learning and Distance Education*, Vienna, Australien, 1999
- [Br99] P. Brusilovsky. Adaptive and Intelligent Technologies for Web-based Education. In *Künstliche Intelligenz, 13(4)*, Seiten 19-25, 1999
- [DHM93] P. Dillenbourg, M. Hilario, P. Mendelsohn, D. Schneider und B. Boorcic. *Intelligent Learning Environments. Report from the project »Les systemes exploratoires intelligents«*. TEC-FA, Universität Genf, 1993
- [DR85] E. L. Deci und R. M. Ryan. *Intrinsic motivation and self-determination in human behaviour*. Plenum, New York, 1985
- [Gn00] C. Gnörlich *MultiNet/WR: A Knowledge Engineering Toolkit for Natural Language Information. Technical-Report 278*. FernUniversität Hagen, 2000
- [Ha97] W. J. Haynie. Effects of Anticipation of Tests on Delayed Retention Learning. *Journal of Technological Education, 9*, 1997
- [He01] H. Helbig. *Die semantische Struktur natürlicher Sprache*. Springer, Heidelberg, 2001
- [LGH02] R. Lütticke, C. Gnörlich und H. Helbig. VILAB – A Virtual Electronic Laboratory or Applied Computer Science. In *Proceedings of the Conference Networked Learning in a Global Environment*, Seiten 135ff CD-ROM, Canada/The Netherlands, ICSC Academic Press, 2002
- [MRGC95] G. R. Morrison, S. M. Ross, M. Gopalakrishnan und J. Casey. The Effects of Feedback and Incentives on Achievement on Computer-Based Instruction. *Contemporary Education Psychology, 20*:32-50, 1995
- [Mu00] J. Musch. Die Gestaltung von Feedback in computergestützten Lernumgebungen: Modelle und Befunde. *Zeitschrift für Pädagogische Psychologie. 13*:148-160, 2000