

# Der Einsatz eines virtuellen Labors in der Informatik-Lehre

Rainer Lütticke, Carsten Gnörlich, Hermann Helbig

FernUniversität Hagen, Fachbereich Informatik  
Intelligente Informations- und Kommunikationssysteme  
Universitätsstr. 1  
58084 Hagen, Deutschland  
rainer.luetticke@fernuni-hagen.de

**Abstract:** Das virtuelle Informatik-Labor (VILAB) ist für die Internet-basierte Lehre an der FernUniversität Hagen geschaffen worden. In dem als Client-Server aufgebauten System werden den Studierenden Aufgaben aus Laborstationen mit verschiedenen Inhalten angeboten, die sie mit unterschiedlichen Software-Werkzeugen lösen müssen. Dabei werden die Studierenden durch eine interaktive tutorielle Komponente zur Korrektur von Lösungen unterstützt.

## 1 Einleitung

Der Fachbereich Informatik der FernUniversität Hagen verfügt über mehrjährige Erfahrungen im Einsatz der Internet-basierten Lehre im Rahmen der Virtuellen Universität (VU; [SM98]). Insbesondere im Bereich der Gestaltung und des Anbietens multimedialer Lehrinhalte durch die VU und der Unterstützung des online-Übungsbetriebes (WebAssign; [BHSV99]) wurden im Fachbereich Informatik der FernUniversität Hagen bereits umfangreiche Arbeiten geleistet und Systeme entwickelt, die bereits erfolgreich ihren Regelbetrieb aufgenommen haben.

Die Grundidee des virtuellen Informatik-Labors (VILAB<sup>1</sup>) besteht darin, eine Laborumgebung zu schaffen, die es den Studierenden ähnlich wie in anderen Naturwissenschaften (z.B. der Physik) erlaubt, ihre in den Lehrveranstaltungen erworbenen theoretischen und methodischen Kenntnisse in praktischen Laborversuchen zu erproben. Analog 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.

Dadurch ergeben sich mehrere Vorteile: FernUni-Studierende sparen die zeit- und kostenaufwendige Anreise für reale Praktika, der Transfer von theoretischem Wissen in praktische Anwendung wird gefördert ([B<sup>+</sup>00]), zusätzlicher Lernerfolg durch multimediale Bausteine ist am besten durch den Einsatz interaktiver Aufgaben und Simulationen

---

<sup>1</sup>VILAB ist Teil des medin-Projekts (Multimediales Fernstudium in der Medizinischen Informatik), welches durch das BMBF innerhalb des Programms "Neue Medien in der Bildung" gefördert wird.

gewährleistet ([Hel99]) und es können sich auf Dauer bei steigenden Studierendenzahlen administrative Erleichterungen in der Lehre ergeben.

## 2 Architektur

VILAB ist nach einem Client-Server-Modell aufgebaut, das im Wesentlichen aus einem in der Universität betriebenen Lehrserver und den Windows- oder Linux-basierten PCs als Client bei den Studierenden besteht. Server-seitig besteht das System aus einem Kern auf Linux-Basis, der die folgenden zentralen Aufgaben übernimmt:

- Speicherung der Laborinhalte (Dokumente, Aufgabenschablonen etc.), Integration der Materialien in die einzelnen Laborkomponenten und deren Aufbereitung zur Anzeige im Nutzer-Client.
- Der Plattform-Kern verfügt über eine tutorielle Komponente, die den Benutzer durch die Laborstationen führt sowie interaktiv Hilfestellungen zu den gerade bearbeiteten Aufgaben abgeben kann. Die genauen inhaltlich-didaktischen Führungsalgorithmen und Kontrollmechanismen für eine konkrete Laborstation sind allerdings in letzterer zur Verfügung gestellt (didaktische Konzeption s. [LGH02]).
- Verwaltung von externen Werkzeugen und die Bereitstellung einer Kommunikations-Infrastruktur zu den übrigen Plattform-Komponenten sowie zu WebAssign.

Die Studierenden gelangen durch ein Remote-login auf den VILAB-Server, auf dem die benötigten Werkzeuge installiert und betrieben werden. Der Server ist so konfiguriert, dass nach dem Login eine Instanz des Labors gestartet, automatisch das Send-/Empfangssystem der tutoriellen Komponente (s. Kap. 5) aktiviert wird und die Studierenden direkt in die grafische Benutzeroberfläche mit Navigationstool und einem Browser (hier *Netscape*) von VILAB geleitet werden. Dadurch kommen sie weder gewollt noch ungewollt mit dem unterliegenden Linux-System in Berührung. Über das Navigationstool (programmiert in Scheme) sind Software-Werkzeuge, Seiten aus dem WWW sowie PDF- und HTML-Dokumente ansteuerbar. Während einer Sitzung im Labor werden die gestarteten Prozesse durch das VILAB-System kontrolliert. Jeder Nutzer bekommt innerhalb des Systems individuelle Ports und Directories, so dass die Datenanzeige und das Speichern von Dateien gemanagt werden und mehrere Studierende zur gleichen Zeit im Labor arbeiten können.

Bei der Konzeption des Kernsystems wurde ein besonderes Augenmerk darauf gerichtet, die modulare Erweiterbarkeit des virtuellen Labors zu unterstützen, damit aufbauend auf der entwickelten Plattform zukünftig weitere Laborstationen mit neuen Lehrinhalten realisiert, in VILAB integriert und somit fachbereichsweit eingesetzt werden können. Mit einem Lehrstuhl an der FernUniversität Hagen wird diesbezüglich bereits kooperiert. Zu diesem Zweck sind spezielle Datei-Schablonen (in HTML und Scheme) für neue Aufgaben und Lehrstationen bereitgestellt, die von den Autoren nur noch ausgefüllt werden müssen.

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

### 3 Laborstationen und Lehrinhalte

Zur Zeit ist der Lehrinhalt in zwei Stationen aufgeteilt:

**Grundlegende Paradigmen der Programmierung:** Hier sind Java und von einem kooperierenden Lehrgebiet Prolog und Scheme prototypisch eingebunden. Durch die zukünftige Implementierung von C sind die Paradigmen der prozeduralen, objektorientierten, funktionalen und logischen Programmierung zur Verfügung gestellt.

**Wissensrepräsentation und -verarbeitung:** Neben dem Kennenlernen des Wissensrepräsentationsparadigmas MultiNet<sup>2</sup> sollen die Studierenden in dieser Station Aufgaben und Experimente mit einem grafischen Werkzeug (MWR [Gnö00]) durchführen. Die inferenzbasierten Methoden stammen dabei aus dem Bereich des natürlichsprachlichen Zugangs zu relationalen Datenbanken bzw. der programmtechnischen Unterstützung des Wissenserwerbs und basieren auf der Programmiersprache Scheme und SQL. Somit kann auf die in der anderen Lehrstation erworbenen Kenntnisse aufgebaut werden.

### 4 Benutzer-Clients

Der Zugang der Studierenden zu dem System wird über dezentrale Benutzer-Clients vorgenommen, damit die Studierenden selbständig von zu Hause mit dem System arbeiten können. Die Herstellung der Verbindung zwischen dem Lehrserver und den Clients geschieht über das Internet. Um möglichst vielen Studierenden den Zugang zum Labor technisch zu ermöglichen, muss der Benutzer-Client auf x86-Rechnern mit einer durchschnittlichen Hardware-Ausstattung und den Betriebssystemen Windows 9x, Linux oder kompatiblen Versionen lauffähig sein. Client-seitig wird vorzugsweise nur frei verfügbare Software eingesetzt, die leicht beschaffbar und installierbar ist. Das Remote-login ist für Studierende, die bereits Linux einsetzen, trivial. Windows-Benutzer müssen zusätzlich die Open-Source-Pakete *cygwin* und *XFree86 4.x* installieren, um durch die so bereitgestellte entsprechende Unix-Funktionalität den Zugang herstellen zu können. 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. Der Nachteil einer möglichen notwendigen Software-Installation 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. Dadurch gewährleistet der gewählte Zugang zu VILAB eine wesentlich bessere Performanz und Darstellungsqualität als beispielsweise ein ausschließlicher Einsatz einer Browser-Umgebung, bei dem die Ein-

<sup>2</sup>MultiNet ([Hel01]) ist ein Wissensrepräsentationsparadigma, das generell für die Wissensverarbeitung und insbesondere für die semantische Repräsentation natürlichsprachlicher Informationen geschaffen wurde.

und Ausgabeschnittstellen der Laborwerkzeuge auf ein Java-Applet umgeleitet werden müssten.

## 5 Interaktive tutorielle Komponente

Die Steuerung und Überwachung der Laborarbeit wird durch die Schnittstelle zu der interaktiven tutoriellen Komponente mit seinem Sende/Empfangssystem (Java) vorgenommen. Zu den Hauptaktivitäten, die über diese Schnittstelle abgewickelt werden, gehören die folgenden Aktionen:

- Bereitstellung der Aufgaben in verschiedenen Formaten (HTML, PDF)
- Einsendung einer (Teil-)Lösung verbunden mit der Aufforderung an die Tutor-Komponente, eine Bewertung oder eine Zwischenstandsmeldung abzugeben.
- Übermittlung eines Fehlerzustandes oder einer anderen erklärungsbedürftigen Situation durch ein Laborwerkzeug, zu der erklärende Informationen angezeigt werden sollen.
- Endgültige Einsendung und Speicherung von bearbeiteten Aufgabenlösungen.

Zur konkreten Ausgestaltung der Tutor-Schnittstelle existieren ein internes und ein externes Tutormodell, die beide eine intelligente Analyse der studentischen Lösung nach Aufforderung durchführen ([Bru99]), aber einen unterschiedlichen Grad an Eigenständigkeit der Lehrmodule voraussetzen. Das externe Modell bietet zusätzlich die Möglichkeit einer interaktiven Problemlösungsunterstützung ([Bru99]).

### 5.1 Externer Modus

Der externe Modus ist für Laborkomponenten gedacht, die weitgehend autark mit dem Benutzer interagieren. Ein Beispiel für eine derartige Komponente ist das grafische Werkzeug zur Wissensrepräsentation (s. Kap. 3), das über eine eigene Benutzeroberfläche zur Anzeige und Manipulation der Wissensbasis verfügt. In diesem Fall beschränkt sich die Aufgabe der Tutor-Komponente darauf, die Laborkomponente zu starten und von ihr Anweisungen in Form von Fehlercodes und Aufforderungen zur Einsendung bzw. Korrektur von Aufgabenlösungen entgegenzunehmen, während die eigentliche Interaktion mit dem Benutzer und die Analyse von Lösungen durch die Laborkomponente selbst durchgeführt wird. Zur Kommunikation mit den Laborkomponenten dient in diesem Fall eine Socket-Verbindung; da Socket-Schnittstellen unter fast allen gängigen Programmiersprachen vorhanden sind, ergibt sich so eine optimale Unterstützung zur Einbindung bereits vorhandener Softwarekomponenten. Der Tutor selbst wird mit Hilfe einer Ressourcendatei vorkonfiguriert, in der für Fehler- und Steuer-codes die vom Tutor zu ergreifenden Aktionen (z.B. Fehlertexte

anzeigen, Aufruf von Skripten zur Bereinigung des Fehlerzustandes) angegeben sind. Dynamisch erzeugte Fehlertexte in HTML und Javascript werden dabei in das individuelle VILAB-Directory des Nutzers geladen und im Browser angezeigt.

## 5.2 Interner Modus

Im internen Modus hat der Tutor eine stärkere Kontrolle über den Ablauf der Softwarekomponenten im Labor. Dieser Modus wird typischerweise bei der Einbindung kommandozeilenorientierter Komponenten wie z.B. von Programmiersprachen-Übersetzern angewendet. In diesem Fall behält der Tutor nach dem Starten die Kontrolle über den Prozess, indem die Standard-Eingabe- und Ausgabe-Datenströme des Prozesses permanent überwacht werden. Da die Interpretation der Standard-Ausgabe hochgradig von der Syntax der jeweiligen Softwarekomponente abhängig ist, kann die Reaktion des Tutors in diesem Fall nicht durch einen einfachen Ressourceneintrag festgelegt werden, sondern es muss durch den Autor der betreffenden Lehrstation eine geeignete Java-Klasse innerhalb des Tutors bereitgestellt werden. Diese Java-Klasse stellt innerhalb des WebAssign-Systems einen sogenannten "Korrektur-Server" dar. Dieser kann von den VILAB-Nutzern (via CORBA) zur Korrekturanfrage oder Speicherung einer Aufgabenlösung angesprochen werden. Durch die Verarbeitung der Eingaben und nachfolgend ausgelöste Prozesse (z.B. Compilieren von Programmen) werden vom Tutor im Labor entsprechende Reaktionen darauf veranlasst (z.B. Fehlertexte, s.o.).

## 6 Curriculare Einbettung und erste Erfahrungen

Ein Prototyp von VILAB befindet sich im Rahmen eines Seminars im Lehrgebiet Künstliche Intelligenz (KI) seit SS2002 an der FernUniversität Hagen im Einsatz. Da zwar die Belegung eines Seminars (Bachelor) bzw. von zwei Seminaren (Diplom) zum Pflichtprogramm eines Studierenden an der FernUniversität Hagen gehört, aber mindestens sechs Seminare pro Semester zur Auswahl stehen, ist die Teilnehmerzahl im KI-Seminar jedoch relativ gering (bzgl. der letzten zwei Jahre im Schnitt  $\sim 10$  Studierende). Da die überwiegende Anzahl der Seminare in traditioneller Weise ablaufen, bietet sich gerade in diesem Bereich die Gelegenheit, neue Methoden in der Informatik-Lehre einzusetzen, um den Studierenden Wahlmöglichkeiten zu geben. Unser neuartiges Konzept, die im Vordergrund stehenden schriftlichen Seminararbeiten mit zugehörigem Vortrag durch im Vorfeld begleitende praktische Übungen im virtuellen Labor zu ergänzen, scheinen die Studierenden attraktiv zu finden. Darauf weist jedenfalls die relativ große Anzahl von 20 Seminarteilnehmern in diesem Semester (SS2002) hin. Das virtuelle Labor selbst hat nach ersten studentischen Rückmeldungen zufolge einen vielversprechenden Eindruck hinterlassen, denn bereits drei Seminar Teilnehmer überlegen, ob sie ihre Bachelor-/Diplomarbeit im Themenbereich VILAB ansiedeln wollen. Leider hat sich jedoch auch gezeigt, dass die Internetanbindung vieler Studierender nicht gut genug ist (Modem und ISDN), um ansprechende Reaktionszeiten des Labor-Systems zu erhalten. Daher sind alle im Labor aufruf-

baren HTML- und PDF-Dokumente auch über eine gewöhnliche URL zur offline-Nutzung aufruf- und downloadbar. Das Arbeiten in VILAB von den Studienzentren aus, zeigt aber, dass das Labor-System prinzipiell sehr kurze Reaktionszeiten hat, so dass angenehm in dem System agiert werden kann. Die Installation der Software für Windows-Nutzer (85%) stellte die Studierenden vor keine allzu großen Probleme. Eine Supportseite im WWW für Hinweise über behobene Fehler in VILAB oder Tipps zum Arbeiten mit VILAB sowie eine Newsgroup für Diskussionsbedarf steht den Studierenden zur Verfügung, wird aber nur wenig in Anspruch genommen, da technische Probleme (z.B. Internetanschluss) und die Seminarthemen individuell unterschiedlich sind. Daher erfolgt die Kommunikation zwischen den Betreuern und den Seminarteilnehmern größtenteils über E-Mails.

Im WS2002/03 wird eine erweiterte Version von VILAB im Rahmen eines Fachpraktikums in Java über Neuronale Netze eingesetzt werden. Ein Fachpraktikum ist Pflicht sowohl für den Bachelor- als auch für Diplomstudiengang. Da jeweils zwei Fachpraktika pro Semester angeboten werden, ist eine größere Teilnehmerzahl zu erwarten, so dass VILAB in größerem Rahmen Anwendung finden wird. In VILAB werden die Studierenden ihre (Teil-)Ergebnisse testen oder mit Simulationen experimentieren können.

Auf längere Sicht soll VILAB auch zur Einsendung und automatischen Korrektur von Übungen zu gewöhnlichen Kursen eingesetzt und um kooperative Lehr-/Lernelemente erweitert werden.

## Literaturverzeichnis

- [B<sup>+</sup>00] S.-P. Ballstaedt et al. *Planung, Entwicklung, Durchführung von Fernstudienangeboten*. Deutsches Institut für Fernstudienforschung an der Universität Tübingen, 2000.
- [BHSV99] J. Brunsmann, A. Homrighausen, H.-W. Six, J. Voss. Assignments in a Virtual University - The WebAssign-System. In *Proc. 19th World Conference on Open Learning and Distance Education*, Vienna, Austria, 1999.
- [Bru99] P. Brusilovsky. Adaptive and Intelligent Technologies for Web-based Education. In *Künstliche Intelligenz*, 13(4), S. 19–25, 1999.
- [Gnö00] C. Gnörlich. *MultiNet/WR: A Knowledge Engineering Toolkit for Natural Language Information*. Technical-Report 278. FernUniversität Hagen, 2000.
- [Hel99] F.P. Helms. *Evaluation des CD-ROM-Kurses "Grundzüge der Betriebswirtschaftslehre II"*. Zentrum für Fernstudienentwicklung an der FernUniversität Hagen, 1999.
- [Hel01] H. Helbig. *Die semantische Struktur natürlicher Sprache: Wissensrepräsentation mit MultiNet*. Springer, Berlin, 2001.
- [LGH02] R. Lütticke, C. Gnörlich, H. Helbig. VILAB - A Virtual Electronic Laboratory for Applied Computer Science. In *Proceedings of the Conference Networked Learning in a Global Environment*, Canada/The Netherlands, 2002. ICSC Academic Press.
- [SM98] G. Schlageter, S. Mittrach. Virtuelle Universität. In *Informatik Forschung und Entwicklung, Volume 13*, S. 159–162, 1998.