

Internet-basierte Lehre im Rahmen von VILAB

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: Um praktische Aufgaben und Versuche in der Informatik online durchführen zu können, wurde ein virtuelles Informatik-Labor (VILAB) an der FernUniversität Hagen realisiert. Das Labor integriert das seit mehreren Jahren im Einsatz befindliche System WebAssign. Neben dessen ursprünglichen Verwendungsgebiet, den online-Übungen, wurde VILAB speziell als Unterstützung für Praktika und Seminare geschaffen. Bei der Aufgabenbearbeitung mit optionalem Einsatz von Softwarewerkzeugen werden die Studierenden durch eine interaktive tutorielle Komponente zur Korrektur von Lösungen unterstützt.

1 Von WebAssign zu VILAB

Seit 1998 ist WebAssign¹ ([BHSV99]) an der FernUniversität (FU) Hagen ein etabliertes System zur Durchführung universitärer Übungsveranstaltungen über das Internet. Der Schwerpunkt dieses Systems sind Web-basierte online-Übungen mit Test- und manuellen wie auch automatischen Korrekturmöglichkeiten studentischer Lösungen. Durch die automatischen Tests kann den Studierenden die (beliebig oft wiederholbare) Vorprüfung ihrer Lösungsversuche ohne zeitliche Verzögerung durch eine online-Vorkorrektur geboten werden. Lehrende werden durch die Erstellung der Aufgaben, Korrektur und Distribution von Ergebnissen via Internet unterstützt. Das produktreife System wird seit März 2001 im Rahmen der integrierten Plattform der FU Hagen vom Universitätsrechenzentrum betrieben und ist seit diesem Zeitpunkt Teil des Lernraums Virtuelle Universität (VU) der FU. WebAssign ist als Quelltext lauffähiges System im Rahmen der Initiative CampusSource verfügbar. Das System wird z.Z. in zahlreichen Kursen an deutschen Hochschulen (im Laufe der letzten zwei Jahre an 11 Universitäten und einer Fachhochschule) in natur-, ingenieur-, geisteswissenschaftlichen und medizinischen Fakultäten eingesetzt. Standardmäßig können in WebAssign Aufgaben, in denen nach Zahlen, Zuordnungen und Begriffen gefragt wird, sowie Multiple-Choice Tests automatisch korrigiert werden. Für kompliziertere Aufgabentypen mit automatischer Korrektur müssen sogenannte "Korrektur-Server" vom Autor einer Aufgabe programmiert werden, die via CORBA mit dem WebAssign System kommunizieren.

¹<http://www-pi3.fernuni-hagen.de/WebAssign/>

Neben diesen komplizierteren Aufgabentypen erfordern Aufgabenstellungen innerhalb eines Praktikums oder praktischer Übungen in der Informatik jedoch weitere Elemente, nämlich den Einsatz von Softwarewerkzeugen und Simulationsumgebungen und mehr als eine einfache Korrektur, die ein “richtig” oder “falsch” liefert. Die Vorteile eines online-Praktikums sind leicht ersichtlich, denn Studierende an der FU sparen so die zeit- und kostenaufwendige Anreise für reale Praktika und können zeitlich und räumlich flexibel in einer interaktiven Lernumgebung arbeiten, um den Transfer von theoretischem Wissen und methodischen Kenntnissen in praktische Anwendungen trainieren zu können. Außerdem könnten sich auf Dauer durch online-Praktika auch administrative Erleichterungen ergeben. Aus diesem Grunde haben wir ein virtuelles Informatik-Labor (VILAB²) entwickelt.

2 Funktionsweise von VILAB

Studierende, die an einem Kurs, in dem VILAB eingesetzt wird, teilnehmen wollen, melden sich im Vorfeld bei dem veranstaltenden Lehrgebiet an. Der Administrator des Labors richtet einen Account ein und verschickt eine Anleitung inklusive Passwort zum Einstieg in das Labor. Dieser erfolgt über gewöhnliche Rechner (Windows oder Linux), die dezentrale Benutzer-Clients darstellen, via Internet zum VILAB-Server an der Universität. Die Studierenden gelangen durch ein Remote-login auf diesen Server, auf dem die Komponenten des Labors ([LGH02]) betrieben werden. Das Remote-login ist für Studierende, die bereits Linux einsetzen, trivial. Windows-Benutzer müssen die Open-Source-Pakete *cygwin* und *XFree86 4.x* installieren, um durch die so bereitgestellte Unix-Funktionalität den Zugang herstellen zu können ([LGH02]). Nach dem Login wird automatisch das Send-/Empfangssystem der tutoriellen Komponente (Java) aktiviert und die Studierenden werden direkt in die grafische Benutzeroberfläche von VILAB geleitet. Da jeder Nutzer innerhalb des Systems individuelle Ports und Directories besitzt, kann die Datenanzeige und das Speichern von Dateien geregelt werden, so dass mehrere Studierende zur gleichen Zeit im Labor arbeiten können. Während der Laborarbeit eines Nutzers kann dieser Dokumente lesen, die für die Bearbeitung einer Aufgabe hilfreich sind, mit den Softwarewerkzeugen arbeiten, (Teil-)Lösungen speichern und einsenden und sich dabei von der interaktiven tutoriellen Korrekturkomponente unterstützen lassen. Bei auftretenden Fragen können sich die Studierenden via E-Mail an die Betreuer wenden oder untereinander in einer Newsgroup diskutieren. Bei schlechter Internetverbindung können die Studierenden auch alle Dokumente über eine gewöhnliche URL aufrufen und zur offline-Nutzung herunterladen³.

Eine Lösungseinsendung veranlasst automatisch die interaktive tutorielle Komponente zu Reaktionen. Es existieren dabei ein internes und ein externes Tutormodell, die beide eine intelligente Analyse der studentischen Lösung durchführen ([Bru99]), aber einen unterschiedlichen Grad an Eigenständigkeit der Lehrmodule voraussetzen. Das externe Modell bietet zusätzlich die Möglichkeit einer interaktiven Problemlösungsunterstützung ([Bru99]). Dieser Modus ist für Laborkomponenten gedacht, die weitgehend autark mit

²VILAB 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.

³<http://pi7.fernuni-hagen.de/infolab/texte.html>

dem Benutzer interagieren (z.B. grafisches Werkzeug s. Kap. 3). Die Laborkomponente sendet in diesem Fall nur Anweisungen in Form von Fehlercodes, während die eigentliche Interaktion mit dem Benutzer und die Analyse von Lösungen durch die Laborkomponente selbst durchgeführt wird. Der interne Modus wird typischerweise bei Aufgaben verwendet, die das WebAssign nutzen. Die studentischen Lösungen werden in diesem Fall von den in Kap. 1 erwähnten Korrektur-Servern analysiert, die vom Autor zu programmieren sind. In beiden Modi erzeugt die tutorielle Komponente dynamisch Fehlertexte (mit Fehleranalyse, Verbesserungsvorschlägen, möglichen Literaturhinweisen, und Leistungsbewertung) in HTML und Javascript, die in das individuelle VILAB-Directory des Nutzers geladen und im Browser angezeigt werden ([LGH02]). Je nach Aufgabentyp brauchen die Studierenden so nicht auf eine manuelle Korrektur zu warten, die bei traditionellen Übungs- und Praktikumsformen meist eine längere Zeit in Anspruch nimmt, oder aber sie bekommen bei komplizierten Lösungen, die noch eine detailliertere Korrektur benötigen, als die tutorielle Komponente sie liefern kann, zumindest eine Mitteilung über die Güte ihrer Lösung.

3 Gestaltung des Lernmaterials und dessen Inhalt

Die grafische Nutzeroberfläche besteht aus einem Navigationstool und einem Browser (hier *Netscape*). Über das Navigationstool (Scheme) sind verschiedene Laborstationen, die jeweils dazu gehörenden Aufgabenstellungen, Software-Werkzeuge, Seiten aus dem WWW sowie PDF- und HTML-Dokumente ansteuerbar. Auf diese Weise kann WebAssign unsichtbar in das Labor integriert werden. Das Gesamtsystem kann nicht unter der Benutzeroberflächen-Regie von WebAssign ablaufen, da der Benutzer-Client der Laborplattform einen Grad an Interaktivität erlauben soll, der sich nicht durch statische HTML-Seiten oder den Einsatz von Java-Applets realisieren lässt. Durch den Verzicht auf die Implementierung der gesamten Benutzeroberfläche innerhalb eines Internetbrowsers ergeben sich umfangreiche technische und inhaltliche Gestaltungsmöglichkeiten für das Labor. Die Softwarewerkzeuge darin können so direkt in einem eigenen Fenster aufgerufen werden, ohne dass beispielsweise die Ein- und Ausgabeschnittstellen der Laborwerkzeuge auf ein Java-Applet umgeleitet werden müssten, wie dies bei einem ausschließlichen Einsatz einer Browser-Umgebung nötig wäre. Dadurch gewährleistet der gewählte Zugang zu VILAB eine gute Performanz und Darstellungsqualität.

Die Lehrinhalte sind zur Zeit in die beiden Laborstationen *Grundlegende Paradigmen der Programmierung* und *Wissensrepräsentation und -verarbeitung* aufgeteilt. In die erste Station sind Java (und zukünftig auch C) und von einem kooperierenden Lehrgebiet Prolog und Scheme eingebunden. Die zweite Station dient zum Kennenlernen des Wissensrepräsentationsparadigmas MultiNet⁴ und zum Experimentieren und Arbeiten mit einem grafischen Werkzeug (MWR [Gnö00]), das auf diesem Paradigma basiert. Aus dem Bereich des natürlichsprachlichen Zugangs zu relationalen Datenbanken (SQL) bzw. der programmtechnischen Unterstützung (Scheme) des Wissenserwerbs stammen die inferenzbasierten Methoden.

⁴MultiNet ([Hel01]) ist ein Wissensrepräsentationsparadigma, das generell für die Wissensverarbeitung und insbesondere für die semantische Repräsentation natürlichsprachlicher Informationen geschaffen wurde.

4 Curriculare Einbettung

VILAB befindet sich im SS2002 in seinem Ersteinsatz. Da jedoch das Labor auf der mehrjährigen Erfahrung in Internet-basierter Lehre aufbaut, ist ein relativ reibungsloser Betrieb gewährleistet. Auf Dauer wird VILAB Seminare in der Informatik durch praktische Elemente ergänzen, in Praktika und zu online-Übungsveranstaltungen (der Domäne von WebAssign) unter Verwendung von Softwarewerkzeugen eingesetzt werden.

Der zuerst genannte Einsatz ist in unserem Lehrgebiet realisiert. Da jedes Semester mindestens sechs Seminare in der Informatik angeboten werden, haben die Studierenden viele Wahlmöglichkeiten zwischen den einzelnen Seminarformen, um ihr Pflichtseminar zu absolvieren. In unserem Seminar-Konzept stehen die schriftlichen Seminararbeiten mit zugehörigem Vortrag zwar im Vordergrund, aber im Vorfeld werden diese durch praktische Übungen unter Verwendung von Softwarewerkzeugen in VILAB begleitet. Obwohl eine Evaluation noch nicht erfolgen konnte, haben wir positive Anzeichen hinsichtlich der Annahme dieser Lehrform von studentischer Seite bekommen (erhöhte Belegungszahlen, positive Rückmeldungen der Studierenden, Anfragen nach Bachelor- und Diplomarbeiten). Es zeigt sich aber auch, dass die Internetanbindung vieler Studierender zu Hause nicht gut genug ist, um ansprechende Reaktionszeiten des Labor-Systems zu erhalten. Jedoch kann von den Studienzentren, die eine gute Anbindung besitzen, angenehm in VILAB gearbeitet werden. Prinzipiell also hat das Labor-System sehr kurze Reaktionszeiten.

Im WS2002/03 wird VILAB im Rahmen eines Fachpraktikums in Java eingesetzt werden. Ein solches Praktikum ist Pflicht sowohl für den Bachelor- als auch für Diplomstudiengang. Da jeweils zwei Fachpraktika pro Semester angeboten werden, besteht aber auch hier eine Wahlmöglichkeit für die Studierenden. Die Praktikumssteilnehmer werden in VILAB ihre (Teil-)Ergebnisse eines komplexen Programms testen oder mit Simulationen experimentieren können, um so durch schnelles Feedback zu ihren (Teil-)Lösungen in kürzerer Zeit zum Ziel gelangen zu können, motivierter zu bleiben und um die Vorteile der standardisierten Entwicklungsumgebung in VILAB nutzen zu können.

Literaturverzeichnis

- [BHSV99] J. Brunsmann, A. Homrighausen, H.-W. Six, and 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.
- [Hel01] H. Helbig. *Die semantische Struktur natürlicher Sprache: Wissensrepräsentation mit MultiNet*. Springer, Berlin, 2001.
- [LGH02] R. Lütticke, C. Gnörlich, and 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.