

Internet-basierter Übungsbetrieb in Technischer Informatik

Web-Based Exercises in Computer Engineering

Udo Hönig, Jörg Keller, Wolfram Schiffmann, FernUniversität Hagen

Zusammenfassung Wir stellen ein Web-basiertes System zur Abwicklung des Übungsbetriebs im Grundstudium Technische Informatik an der FernUniversität vor. Das System umfasst drei Komponenten: ein Entwurfswerkzeug zur leichten Eingabe von Digitalschaltungen und der Möglichkeit der Selbstkontrolle ihrer Korrektheit, ein Web-gestütztes System zur Einreichung der Schaltungen und Rücksendung der korrigierten Einsendungen und ein Modul zum halbautomatischen Test der Korrektheit der eingesandten Schaltungen. Wir berichten weiterhin über die positiven Erfahrungen während des Pilotbetriebs und über die positiven Ergebnisse der Evaluation des ersten Einsatzes

in einem Kurs mit mehreren Tausend Kursteilnehmern. ▶▶▶
Summary We present a web-based system to conduct exercises in undergraduate computer engineering courses. The system consists of three components: a design tool for simple schematic entry and correctness test of digital circuits, a web-based system for submission of circuits and return of corrected submissions, and a module for semi-automatic testing of submitted circuits' correctness. We report about the positive experiences during the pilot phase and the results of an evaluation during the first deployment in a course with more than one thousand students.

KEYWORDS B.6.3 [Design Aids], J.1 [Education], J.6 [Computer-Aided Design]

1 Einleitung

Große Teile des Informatik-Lehrstoffs sollen nicht nur Wissen, sondern auch den konstruktiven Umgang mit den behandelten Objekten vermitteln. Während in der Praktischen Informatik die Durchführung von Programmierübungen am Rechner bereits weit verbreitet ist, stellt sich die Situation für die Technische Informatik schwieriger dar. Dies ist unter anderem dadurch begründet, dass im Gegensatz zu Programmiersystemen die hier benötigten Entwurfs- und Simulationsumgebungen aufwändiger und weniger verbreitet sind. Dieses Problem soll am Beispiel des entsprechenden einführenden Kur-

ses an der FernUniversität behandelt werden.

Der Kurs „Technische Informatik I“ führt in die Thematik der Schaltfunktionen, Schaltnetze, Schaltwerke, einfachen Speicher und ihren Entwurf ein. Die dazugehörigen Übungen befassen sich naturgemäß ebenfalls mit diesem Thema, d.h. in den Übungsaufgaben sind verschiedene Digitalschaltungen zu entwerfen. Der Entwurf von Digitalschaltungen mit Papier und Stift ist allerdings schwierig. Viele Fehler werden nicht erkannt. Die erste Lösung wird als gegeben hingenommen und eingesandt. Eine tiefere Beschäftigung mit der Materie unterbleibt. In der Fernlehre entfällt dabei zusätzlich die lebendige und interaktive Behandlung des Stoffs in der Übungsgruppe, wo zum Beispiel Schaltun-

gen vom Übungsgruppenleiter an der Tafel entwickelt werden, wodurch die Studierenden einen weiteren Erkenntnisgewinn haben.

Dieser Situation kann abgeholfen werden, indem die Digitalschaltungen tatsächlich mittels eines CAD-Werkzeugs eingegeben und simuliert werden. Die Studierenden können so durch Simulation der eingegebenen Schaltung die Qualität ihrer Lösung selbst kontrollieren. Bei Entdeckung eines Fehlers ist die Motivation höher, diesen zu korrigieren. Insgesamt besteht die Hoffnung, dass sich die Qualität der Einsendungen der Studierenden erhöht. Weitere Vorteile für die Studierenden sind das Erlernen des Umgangs mit einem professionellen Werkzeug und das Erlernen des aktiven Arbeitens mit Digitalschaltungen. Vorteile auf Seiten des Kurs-

Die im vorliegenden Beitrag beschriebenen Forschungen wurden durch Mittel aus dem Innovationsfond 2001 der FernUniversität unterstützt.

anbieters und der Korrektoren sind die bessere Lesbarkeit der eingesandten Lösungen der Studierenden sowie die Möglichkeit der automatischen Korrektheitsprüfung¹ bei einer großen Zahl von Einsendungen. Dies entlastet die Korrektoren von monotoner und fehlerträchtiger manueller Prüfung der Korrektheit. Gleichwohl bleibt für die Korrektoren genügend Arbeit, denn sie müssen weiterhin den sinnvollen Entwurf sowie die sinnvolle Benennung der Signale und andere Kriterien prüfen.

Um diese Anforderung in der Fernlehre umzusetzen, bedarf es eines Entwurfswerkzeugs, das frei verfügbar und einfach zu erlernen ist. Gleichzeitig bedarf es eines Web-gestützten Einsendungs-systems, in das das Entwurfswerkzeug integriert werden kann. Ein wichtiges zu lösendes Detailproblem ist die Selbstkontrolle: man möchte den Studierenden eine Möglichkeit zur Verfügung stellen, sodass sie die Korrektheit ihrer Lösungen prüfen können, bevor sie sie einsenden.

Die oben entwickelte Problemstellung hat sich den Präsenzuniversitäten in einfacherer Form gestellt, da den Studierenden dort lediglich lokal Ressourcen zur Verfügung gestellt werden müssen. Folglich sind seit den 80er-Jahren CAD-Systeme [1] und Simulatoren [4] in Computer-Pools installiert und zur Bearbeitung von Übungsaufgaben benutzt worden. Allerdings hat dabei weder die Selbstkontrolle noch die automatische Korrektur eine Rolle gespielt. Die ortsübergreifende Verfügbarkeit von teuren Software-Paketen mittels Web-zugriff auf zentral installierte Programme wurde ebenfalls bereits erprobt [11], im Emerge Projekt auch für Laborgeräte (siehe [¹ Unter automatischer Korrektheitsprüfung verstehen wir nicht die Verifikation, sondern dass das Ein- und Ausgabeverhalten der Schaltung einer Vorgabe entspricht. Hierbei werden korrekte Lösungen stets als korrekt erkannt, eine falsche Lösung wird nur mit einer verschwindend geringen Wahrscheinlichkeit fälschlicherweise als korrekt betrachtet.](http://www.emerge-</p>
</div>
<div data-bbox=)

project.net). Sie bedingt allerdings längerdauernde und breitbandige Internetverbindungen und ist somit im Allgemeinen keine Option für Fernstudierende. Allen diesen Lösungen ist gemeinsam, dass sie sehr komplexe, kommerzielle CAD-Systeme verwenden und somit vorwiegend von Studierenden im letzten Studienabschnitt verwendet werden, die sich im betreffenden Gebiet vertiefen wollen. Im Fernstudium wurden bei Web-gestützten Übungen lediglich Text- und Multiple-Choice-Aufgaben angeboten [13]. Teilweise wurden die Übungen auch bei Entwurfsaufgaben konventionell durchgeführt [12]. Das Exorciser-Projekt [17] behandelt Web-gestützte Übungen mit automatischer Korrektur. Allerdings werden dort für jeden Aufgabentyp (z.B. der Entwurf eines endlichen Automaten zur Erkennung einer regulären Sprache) eigene Applets entwickelt, sodass der Entwicklungsaufwand sehr hoch ist. In [2] wurde ein Vorausblick gegeben, welche multimediale Elemente in die Fernlehre der Technischen Informatik eingebaut werden könnten. Der vorliegende Artikel stellt die Umsetzung einer der dort beschriebenen Zukunftsvisionen vor.

Der Rest dieses Artikels ist wie folgt aufgebaut. In Kapitel 2 wird das Programm beschrieben, das zur Schaltungseingabe und -simulation benutzt wird. In Kapitel 3 werden Testbenches als Möglichkeit der Korrektheitsprüfung eingeführt. In Kapitel 4 wird die Web-gestützte Abwicklung der Übungen beschrieben, insbesondere die Integration des Programms HADES mit dem WebAssign-System und den Testbenches. In Kapitel 5 wird über die Erfahrungen berichtet, die bei der Erprobung des Systems in einer Pilotphase und dem ersten großen Einsatz gewonnen wurden. Außerdem werden die Ergebnisse der Evaluation dieses Einsatzes vorgestellt. Kapitel 6 fasst den gegenwärtigen Stand zusammen und gibt einen Ausblick über weitere Aktivitäten.

2 Schaltungseingabe und -simulation

In der Technischen Informatik bestehen die Übungsaufgaben oft darin, Digitalschaltungen zu entwerfen, die binäre Eingabemuster zu entsprechenden Ausgabemustern umformen. Die Lösung dieser Übungsaufgaben bereitet Fernstudierenden erfahrungsgemäß Schwierigkeiten, weil ihnen im Gegensatz zu Präsenzstudenten der direkte Kontakt zu Kommilitonen und wissenschaftlichen Mitarbeitern fehlt. Sie können daher ihre Lösungsansätze nicht mit anderen diskutieren und haben keinerlei Möglichkeit, die Korrektheit ihrer Schaltungsentwürfe im Vorfeld zu überprüfen. Beim herkömmlichen papierbasierten Übungsbetrieb erhalten sie Hinweise auf Fehler erst mit der bereits bewerteten Übungsaufgabe. Dabei liegt zwischen Einsendung und Rückgabe der korrigierten Lösungen oft ein Zeitraum von vier Wochen. Um die Motivation und Lernsituation der Fernstudierenden zu verbessern, sollten daher Möglichkeiten geschaffen werden, die Lösungsansätze bereits vor der Einsendung selbst zu überprüfen. Hierzu benötigt man ein Entwurfswerkzeug zur Schaltungseingabe und Simulation. Kommerzielle CAD-Systeme sind nicht nur teuer, sondern auch komplex und sehr schwer zu erlernen. Sie eignen sich nicht für den Einsatz im Übungsbetrieb, da die Studierenden damit überfordert wären.

Zur Eingabe und Simulation von Digitalschaltungen wurde das Programm HADES (Hamburger Design System) [7] ausgewählt. HADES ermöglicht die grafische Eingabe von Digitalschaltungen, die aus Schalt- und Speichergliedern zusammengesetzt sein können. Neben diesen Grundelementen werden auch verschiedene Varianten von Signalgebern und Anzeigeeinheiten für digitale Signale bereitgestellt. HADES unterstützt hierarchische Entwürfe, d.h. es können Module definiert werden, die eine bestimmte

Funktionalität haben. Diese Module können dazu benutzt werden, um komplexe Schaltungen zu strukturieren, und erhöhen so die Übersichtlichkeit der Entwürfe.

Gegenüber kommerziellen Entwurfssystemen bietet dieser komfortable grafische Schaltplaneditor viele Vorteile:

- Da das Programm in Java implementiert wurde, kann es auf jedem Rechner eingesetzt werden, der über ein aktuelles Java Laufzeitsystem verfügt.
- Sowohl das lauffähige Programm, als auch der Quellcode werden kostenlos bereitgestellt. Daher können die Darstellung (Schaltsymbole) und der Funktionsumfang (Basis-Schaltglieder) erweitert sowie an die speziellen Anforderungen eines Kurses angepasst werden.
- Die Programmgröße ist kleiner als 3,5 Mbyte. Daher kann das Programm selbst mit Modem-Anbindung in vertretbarer Zeit über das Internet heruntergeladen werden.
- Das Programm kann nach dem Herunterladen sofort durch Anklicken der Datei gestartet werden und ist sehr leicht erlernbar. Neben den bereits genannten Vorteilen ist dieser Punkt für die Akzeptanz bei den Studierenden von großer Bedeutung.
- Die Schaltungen werden bereits während der Eingabe simuliert, d.h. HADES unterscheidet nicht zwischen Schaltungseingabe und Simulation. Bei kommerziellen CAD-Systemen wird die Simulation meist durch einen separaten Programmteil ausgeführt. Dies erhöht den Zeitaufwand zum Erlernen der Programmbedienung.

3 Testbenches

Um die Korrektheit eines Schaltungsentwurfs zu überprüfen, werden so genannte *Testbenches* eingesetzt. Hiermit können die Studierenden ihre Lösungsansätze zu Hause überprüfen. Sie erhalten da-

mit eine unmittelbare Bewertung ihres Schaltungsentwurfs und können ihre Lösungen solange modifizieren, bis sie eine korrekt arbeitende Schaltung gefunden haben.

Außerdem können mit Hilfe der Testbenches auch die von den Studierenden eingereichten Lösungen automatisch auf Korrektheit überprüft werden (siehe Fußnote 1). Hierzu wird ein Korrekturserver eingesetzt, der die Funktionalität eingereicherter Lösungen überprüfen kann. Die Dienste dieses Servers werden durch das übergeordnete WebAssign-System angefordert (siehe Kapitel 4). Durch den automatischen Test der eingereichten Lösungen erhalten die Korrektoren schon vorab die Information, ob die Schaltung die geforderte Funktionalität hat. Die Überprüfung der Einsendung durch den Korrektor wird durch den Korrekturserver jedoch nicht überflüssig. Ein Schaltungsentwurf kann zwar funktional korrekt sein, aber trotzdem nicht die gegebene Aufgabenstellung lösen. Beispielsweise könnte ein Schaltnetz in disjunktiver Normalform eingereicht werden, obwohl in der Aufgabenstellung die konjunktive Normalform gefordert wird. Es bleibt daher die Aufgabe des Korrektors, die eingereichte Lösung bezüglich der vorgegebenen Rahmenbedingungen und der Einhaltung weiterer Entwurfskriterien zu bewerten.

Durch die Testbenches erhalten die Studierenden eine schnelle Rückmeldung über die Qualität ihrer Entwürfe. Sie können schon zu Hause überprüfen, ob ihr Entwurf die gestellte Aufgabe soweit korrekt löst. Die Möglichkeit zur Selbstkontrolle spornt die Studierenden dazu an, möglichst fehlerfreie Lösungen einzureichen. Dadurch steigt die Motivation und die Lernsituation der Fernstudierenden wird verbessert. Dies zeigt sich auch in einer Erhöhung der Erfolgsquote bei den Einsendeaufgaben (vgl. Abschnitt 5.3).

Im Folgenden soll die Arbeitsweise einer Testbench ausführli-

cher beschrieben werden. Um mit HADES automatische Tests ausführen zu können, muss die zu testende Schaltung zwischen zwei zusätzliche Komponenten, die die Testbench bilden, eingebettet werden. Zur Erzeugung der Eingangssignale verwendet man als Mustergenerator PG (engl.: *Pattern Generator*) ein rückgekoppeltes Schieberegister, das als LFSR (engl.: *Linear Feedback Shift Register*) bezeichnet wird. Die verwendeten LFSRs haben Wortbreiten von 16 bis 32 Bit und werden beim Start mit einem Bitmuster, dem so genannten *seed*, geladen. Aufgrund der Rückkopplungen erzeugt ein LFSR eine Folge von Bitmustern, die als Ansteuersignale auf die zu testende Digitalschaltung gegeben werden. Die Testschaltung reagiert auf die Signale des PG mit Ausgangssignalen. Zur Auswertung dieser Ausgangssignale wird ein SA (engl.: *Signature Analyzer*) benutzt, der ähnlich wie der PG aus rückgekoppelten Schieberegistern besteht. Die Folgezustände der Flipflops werden hier zusätzlich zur Rückkopplung noch mit dem jeweils anliegenden Ausgangsmuster der Testschaltung verknüpft. Dies bewirkt, dass nach einer vorgegebenen Zahl von Eingangsmustern auf die Testschaltung im SA ein für die Funktion der Testschaltung charakteristisches Bitmuster, bezogen auf die Seed-Werte von PG und SA, entsteht. Diese so genannte *Signatur* kann daher für die Überprüfung der Korrektheit der entworfenen Schaltung benutzt werden. Die Signatur zu gegebenen Seed-Werten von PG und SA fasst die *Funktion* der entworfenen Digitalschaltung über eine bestimmte Zahl von Zufallsmustern in einem einzigen (hexadezimalen) Zahlenwert zusammen. Dieser Zahlenwert ist invariant bezüglich der Schaltungsdetails (so genannter *Black-Box Test*). Es kann aufgrund der Signatur aber nicht entschieden werden, ob ein Schaltnetz zum Beispiel in disjunktiver oder konjunktiver Normalform realisiert wurde. Ebenso können Digitalschaltungen Terme enthalten, die

zwar überflüssig sind, aber trotzdem keinen Einfluss auf die geforderte Funktionalität haben. Ist eine eingereichte Lösung korrekt, so stimmt ihr Signaturwert (bei gleichen Seed-Werten für PG und SA) stets mit dem durch eine Musterlösung erzeugten Signaturwert überein. Ist eine eingereichte Lösung falsch, so ist die Wahrscheinlichkeit, dass ihre Signatur mit der der Musterlösung übereinstimmt, verschwindend gering. Deshalb entscheidet der Korrekturserver nach folgendem Kriterium: stimmt der Signaturwert einer eingereichten Lösung mit dem durch eine Musterlösung vorab mit gleichen Seed-Werten für PG und SA erzeugten Signaturwert überein, so löst die mit der Testbench analysierte Digitalschaltung die Aufgabenstellung und die eingereichte Übungsaufgabe wird vom Korrekturserver als korrekt bewertet.

Bild 1 zeigt ein einfaches Beispiel mit 3-Bit PG und SA. Wir wollen annehmen, dass als Aufgabenstellung die Entwicklung eines 3-Bit-Inkrementers ohne Übertrag gegeben war. Starten wir den PG mit dem Seed-Wert 111, so nimmt er in den folgenden Takten die Zustände 110, 100, 001 usw. an. Eine korrekte Schaltung liefert damit nacheinander die Ausgaben 000, 111, 101, 010 usw. Wird der SA mit dem Seed-Wert 000 gestartet, dann nimmt er nacheinander die Werte 000, 111, 011 usw. an. Damit hat die Signatur bei den oben genannten Seed-Werten von PG und SA nach 3 Taktungen, d. h. nach 4 Ausgangswerten, den Wert 011. Wäre der Seed-Wert des SA 111 gewesen, dann lautete die Signatur 010. Hätte die Schaltung einen Fehler, der sich dahingehend äußert, dass beim Eingabewert 100 der Ausgabewert 100 (statt 101) erzeugt wird, so wäre die Signatur im ersten Falle 010 (statt 011).

Zu jeder Übungsaufgabe wird den Studierenden eine entsprechende Testbench bereitgestellt. Um ihre Entwürfe selbst überprüfen zu können, erhalten sie zusätzlich

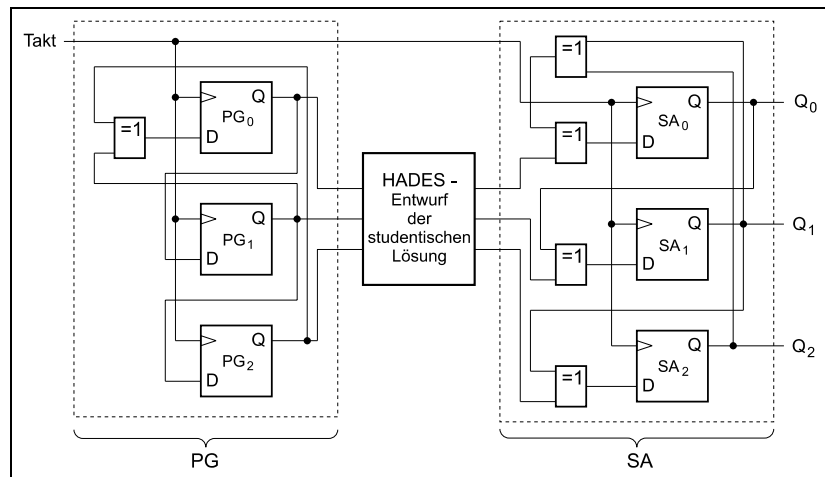


Bild 1 Beispielhafter Aufbau von PG und SA.

ein Seed-Signatur Zahlenpaar. Vor dem Testlauf stellen sie am SA den vorgegebenen Seed-Wert ein. Der Seed-Wert des PG ist fest eingestellt. Wenn der SA nach dem Testlauf die zugehörige Signatur enthält, wissen die Studierenden, dass ihre Schaltung wohl korrekt arbeitet. Die Wahrscheinlichkeit, dass eine falsche Schaltung eine korrekte Signatur liefert, ist, wie oben erläutert, verschwindend gering. In gleicher Weise – aber mit einem anderen Seed-Signatur Zahlenpaar – überprüft auch der Korrekturserver (siehe Kapitel 4) die eingereichten Übungsaufgaben.

Entspricht nach einem Testlauf der Inhalt des SA nicht der mitgelieferten Signatur, so enthält die Schaltung einen Fehler. Da die Testbench einen Black-Box Test darstellt, kann sie keinen weiteren Hinweis auf die Natur des Fehlers liefern. Eine Testumgebung mit automatisch generierten Hinweisen auf die Natur der Fehler scheint im Umfeld der Schaltungsentwicklung auf Grund der Komplexität der Zusammenhänge auf absehbare Zeit aber ohnehin ausgeschlossen. Lediglich bei scharf abgegrenzten Aufgabenstellungen können derzeit Programme, die speziell auf eine Aufgabenstellung zugeschnitten sind, ihr Anwendungswissen als Hinweis äußern. Ein Beispiel hierfür ist der bereits erwähnte Exorciser [17].

4 Systemintegration

WebAssign ist eine Web-basierte Plattform zur Bereitstellung, Bearbeitung und Korrektur von Übungsaufgaben [3]. Das Ziel von WebAssign ist die Automatisierung und Beschleunigung der Arbeitsabläufe beim Übungsbetrieb. In der bisherigen Praxis der FernUniversität wurden gedruckte Übungsaufgaben versandt, von den Studierenden bearbeitet und dann durch externe Korrektoren bewertet. Neben den Bearbeitungszeiten durch Studierende und Korrektoren führt der vierfache Postversand der Papierdokumente zu erheblichen Zeitverzögerungen, Kosten und Verwaltungsaufwand. Außerdem erhalten die Studierenden erst einige Wochen nach der Einsendung ihrer Lösungen eine Rückmeldung über die Ergebnisse. Die Lernsituation wird daher durch Internet-basierte Übungen deutlich verbessert.

WebAssign bietet entsprechend den Anforderungen an den Übungsbetrieb verschiedene Zugänge für Kursbetreuer, Studierende und Korrektoren. Die Kursbetreuer haben die Möglichkeit, Aufgaben und Musterlösungen zu einzelnen Kurseinheiten einzustellen, Termine für deren Freischaltung vorzugeben, die proportionalen Anteile der auszuführenden Korrekturen auf externe Korrektoren festzulegen sowie die korrigierten Übungsaufgaben und die erreichten Punktzahlen einzu-

sehen. Die Studierenden erhalten einen zeitgesteuerten Zugriff auf die Übungsaufgaben und später auf die zugehörigen Musterlösungen. Sie können die Übungsaufgaben online oder offline bearbeiten und ihre Lösungen als HADES-Dateien über das Internet einreichen. WebAssign teilt den Korrektoren – entsprechend der vom Kursbetreuer vorgegebenen Quote – eine gewisse Zahl von Korrekturaufgaben zu.

WebAssign ist ein universelles System, das beliebige Aufgabentypen unterstützt. WebAssign selbst bietet Multiple-Choice und Text-Aufgaben an. Weiterhin können Dateien beliebigen Formats als Lösungen eingereicht werden. Diese werden im einfachsten Fall direkt an die Korrektoren weitergereicht. WebAssign besitzt allerdings auch

eine CORBA-Schnittstelle, über die weitere Server angebunden werden können, die obige Lösungs-Dateien automatisch auf ihre Korrektheit bezüglich der Aufgabenstellung testen können. So wurde, um die Korrektoren bei ihrer Arbeit zu unterstützen, eine automatische Korrekturkomponente (Korrekturserver) zur Funktionsprüfung der mit HADES entworfenen Digital-schaltungen entwickelt, die die gegebenen Testbenches nutzt (siehe Kapitel 3).

Die oben aufgeführte Kombination von beliebigen Aufgabentypen, Offenheit und Erweiterbarkeit durch Open Source und CORBA-Schnittstelle findet sich unseres Wissens nach nur bei WebAssign. Keine der uns bekannten Lernplattformen wie zum Beispiel Goya [8],

ILLIAS [10], EDO-Workspace [5] oder Online-Campus [14] sowie weitere im Rahmen von BMBF-Projekten erstellte Systeme bietet Unterstützung für Übungen in dieser Allgemeinheit. Ein System, das WebAssign in gewisser Weise nahe kommt, ist XLX [9]. Dieses hat allerdings ein viel eingeschränkteres Ziel, nämlich die Realisierung von Übungen im Datenbank-Bereich. Bei XLX ist der Webserver mit einem Datenbankserver fest gekoppelt. Weiterhin werden keine Fremdformate unterstützt.

Um in WebAssign mit Hilfe von HADES automatische Tests ausführen zu können, werden den Studierenden die oben beschriebenen Testbenches als HADES-Dateien zum Herunterladen bereitgestellt. Gemäß

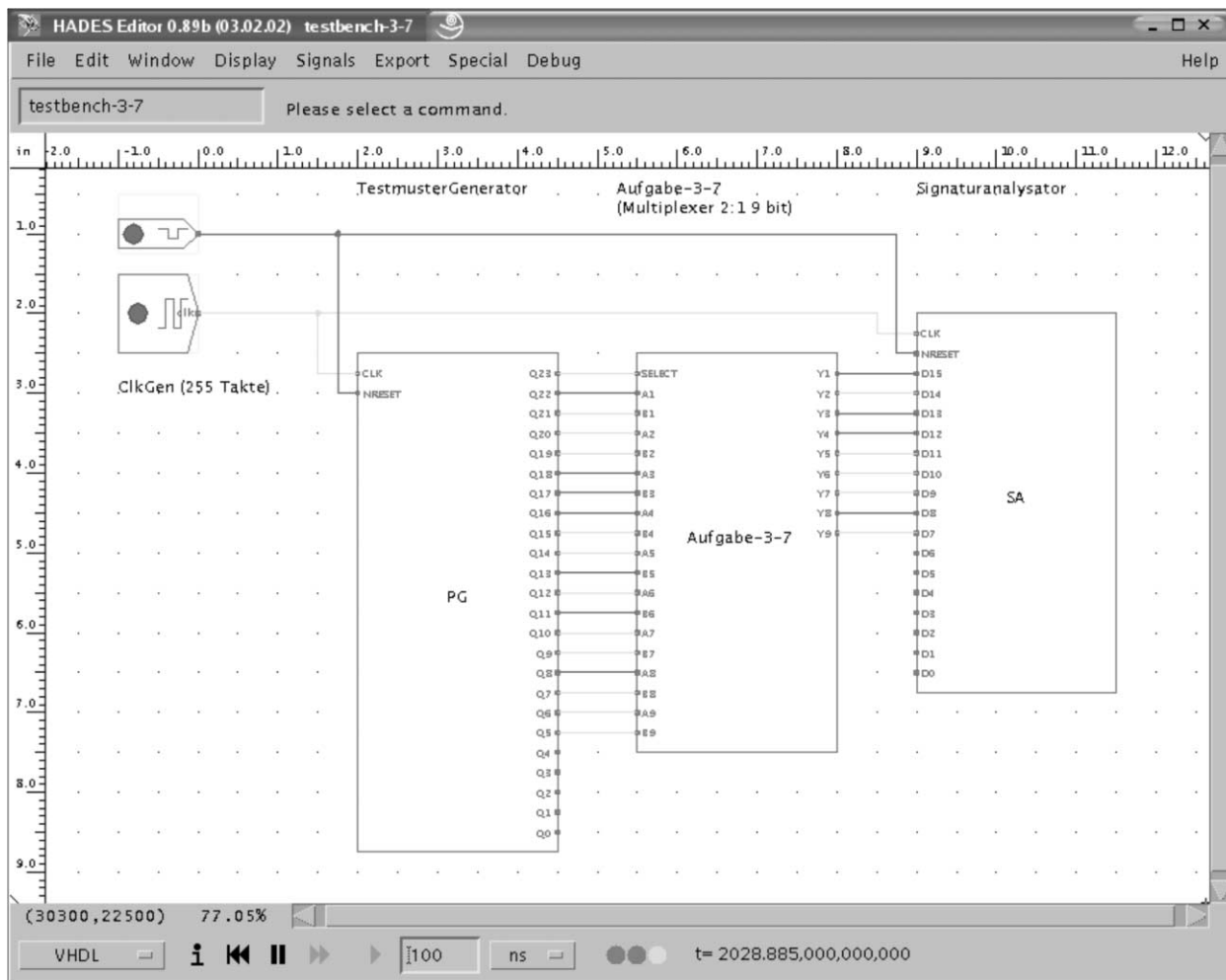


Bild 2 HADES-Screenshot einer Testbench mit Lösungsmodul des Studierenden.

der Aufgabenstellung entwerfen die Studierenden eine Digitalschaltung, die sie in die Testbench einbetten. Bild 2 zeigt einen Screenshot von HADES. Die dargestellte Schaltung zeigt eine Testbench mit dem Lösungsmodul des Studierenden. Mit Hilfe der Testbench haben die Studierenden die Möglichkeit, die Korrektheit der Schaltung anhand eines gegebenen Seed-Signatur Zahlenpaares selbst zu testen. Eine Übungsaufgabe wird in WebAssign eingereicht, indem die entsprechende HADES-Datei zum WebAssign-Server übertragen wird. Zur automatischen Bewertung der eingereichten Lösung leitet WebAssign einen Testlauf auf dem Korrekturserver ein. Dieser arbeitet zwar mit dem gleichen PG, benutzt aber ein anderes Seed-Signatur Zahlenpaar für den SA. Nach dem Testlauf vergleicht der Korrekturserver die Signatur der studentischen Lösung mit der Signatur der Musterlösung und informiert WebAssign, ob die eingereichte Lösung korrekt ist. WebAssign speichert diese Information zusammen mit der studentischen Lösung und reicht sie an die Korrektoren weiter. Ein Korrektor kann seine Anmerkungen entweder direkt in die HADES-Datei schreiben, die er sich in HADES ohnehin betrachten muss, um zum Beispiel Entwurfskriterien zu überprüfen, oder er kann in seiner WebAssign-Maske ein Textfeld benutzen. Im ersten Fall lädt er die kommentierte HADES-Datei wieder an den WebAssign-Server, der sie als Teil der korrigierten Einreichung auch wieder an den Studierenden zurückschickt.

Der Ablauf einer Online-Übung mit WebAssign aus Sicht des Studierenden wird beispielhaft in einem mehrteiligen Videofilm gezeigt, der unter der URL <http://www.ra.fernuni-hagen.de/Online-Uebungen/index.html> abgerufen werden kann.

WebAssign selbst ist ein zentrales System, das auf der Nutzerseite nur einen Web-Browser verlangt. Dezentral wurde lediglich das Pro-

gramm HADES genutzt. Dieses ist allerdings vollständig in Java geschrieben und liegt auch in einer Applet-Version vor. Die Aufgabenstellung befindet sich in einer HTML-Seite. Um eine vollständig zentralisierte Lösung zu realisieren, müsste lediglich innerhalb dieser HTML-Seite das HADES-Applet mit der Testbench als Parameter aufgerufen werden. Statt die HADES-Schaltung am Ende zunächst im Dateisystem zu speichern und dann über die HTML-Seite bei WebAssign einzureichen, würde sie direkt an WebAssign geschickt. Ein Nachteil bei dieser Lösung wäre allerdings die zu erwartende, lange Zeit zum Download des Applets. Eine Alternative wäre eine zentrale Lösung, die HADES auf dem Server startet und dem Benutzer lediglich die Benutzeroberfläche mittels VNC [15] innerhalb eines Web-Browsers zur Verfügung stellt.

Um HADES-Übungen mit einem anderen System als WebAssign durchführen zu können, müsste dieses System mindestens HADES-Dateien akzeptieren und zu den Korrektoren transportieren können. Um den Funktionstest durchführen zu können, müsste das System die HADES-Datei zusammen mit aufgabenspezifischen Parametern automatisch an einen weiteren Server, den Korrekturserver, weiterreichen können. Die Ergebnisse dieses Tests wären dann zusammen mit der HADES-Datei an den Korrektor zu schicken.

5 Ergebnisse

Das beschriebene Verfahren wurde zunächst während des Sommersemesters 2002 im Rahmen einer Pilotphase mit einer kleinen Gruppe freiwilliger Teilnehmer erprobt. Die dabei gewonnenen Erkenntnisse ermutigten zu einer baldigen Übernahme des nur geringfügig zu modifizierenden Systems in den regulären Lehrbetrieb. Im Sommersemester 2003 wurde es erstmalig von allen Studierenden des Kurses „Technische Informatik I“ zur Durchführung ihrer Übungen benutzt.

5.1 Pilotphase

Die Kombination von HADES mit WebAssign wurde erstmalig während des Sommersemesters 2002 im Rahmen einer zweiwöchigen Pilotphase mit circa 15 Teilnehmern getestet. Durch die relativ geringe Teilnehmerzahl sowie die überschaubare Dauer der Testphase wurde sichergestellt, dass die während des gesamten Versuchs gewonnenen Erkenntnisse auf wenige Betreuer konzentriert werden konnten. Dadurch wurde nicht nur eine schnelle Beseitigung der aufgetretenen Schwierigkeiten ermöglicht, sondern auch in vielen Fällen eine bessere Erkennung der aus Planungsentscheidungen resultierenden Auswirkungen.

Die Pilotanwender wurden per Zufall aus den sich bewerbenden Kursteilnehmern ausgewählt. Kurz vor Beginn der eigentlichen Bearbeitungszeit wurden ihnen neben ihren WebAssign-Zugangsdaten das Programm HADES, sowie Benutzeranleitungen für HADES und WebAssign zugesandt.

Die zur Verfügung gestellten Aufgaben wurden aus allen Themengebieten des Kurses „Technische Informatik I“ ausgewählt und bestanden teilweise aus neuen, in einigen Fällen aber auch aus bereits im bisherigen Übungsbetrieb verwendeten Aufgaben. Letztere wurden mit aufgenommen, damit die Studierenden an ihnen den Umgang mit den für sie neuen Anwendungen einüben konnten. Es wurden deutlich mehr Aufgaben angeboten, als tatsächlich innerhalb des zur Verfügung stehenden Zeitraums bearbeitet werden konnten. Für diese Entscheidung gibt es mehrere Gründe. Einerseits wurde den Pilotanwendern durch die zahlreicheren Aufgaben die Möglichkeit zum Schließen eigener Wissenslücken geboten, andererseits konnten dadurch auch mehr Aufgaben hinsichtlich ihres Einsatzes im Onlinebetrieb getestet werden. Zudem sind die Teilnehmer dazu ermuntert worden, anhand der von ihnen nicht bearbeiteten Aufgaben die Stabilität von WebAssign und dem Korrekturserver durch die

Einsendung falscher Ergebnisse und fremder Dateiformate zu überprüfen.

Aufgrund der geringen Anforderungen an Hardware und Systemumgebung konnte HADES bei allen Teilnehmern problemlos installiert werden. Auch der Zugriff auf die online zur Verfügung gestellten Übungen verlief unproblematisch.

Die Bedienung von HADES und WebAssign wurde von der Mehrzahl der Teilnehmer als intuitiv verständlich angesehen. Die im Vorfeld versandte Dokumentation wurde daher von einigen Studierenden nicht verwendet oder nur überflogen. Durch den Einsatz der Simulatorfunktionen von HADES sowie der mitgelieferten Testbenches konnten die Studierenden auch solche Aufgaben lösen, deren Schwierigkeitsgrad den der üblicherweise gestellten Aufgaben überstieg.

Gemäß den Vorgaben wurden von den Studierenden neben ernst gemeinten Lösungsversuchen auch zahlreiche Fremdformate, u. a. auch Binärdateien, eingesandt. In keinem dieser Fälle konnten Störungen bei der Verarbeitung registriert werden. Durch das für WebAssign entwickelte HADES-Korrekturmodul wurden die eingereichten Lösungen zunächst auf ihre Korrektheit überprüft. Das Ergebnis wurde dann dem Korrektor zusammen mit der Einsendung zur Verfügung gestellt. Aufgrund dieser Funktionsüberprüfung konnte sich die Korrekturkraft bei den richtigen Einsendungen auf die Begutachtung des Schaltungsdesigns beschränken, wodurch eine deutlich schnellere und einfachere Korrektur möglich wurde.

Abschließend wurde den teilnehmenden Studierenden ein Beurteilungsbogen zugesandt, in dem sie WebAssign, HADES und die von uns realisierte Integration der beiden Systeme sowie deren Einsatz im Lehrbetrieb beurteilen sollten. Insgesamt fand die getestete Lösung eine breite Zustimmung, wobei die im Vergleich zum üblichen Postweg sehr schnelle Korrektur der

Einsendungen als größter Vorteil angesehen wurde.

5.2 Einsatz in einem großen Kurs

Der Kurs „Technische Informatik I“ besteht aus sieben Teilen, so genannte *Kurseinheiten*, die im 14-tägigen Rhythmus an die Studierenden versandt werden. Zu jeder dieser Kurseinheiten gehört ein Aufgabenheft, das jeweils vor dem Ende einer vorgegebenen Bearbeitungsfrist bei der FernUniversität eingereicht werden muss. Die Übungen zu den ersten beiden Kurseinheiten bestehen ausschließlich aus Multiple-Choice Aufgaben und Aufgaben mit numerischem Ergebnis. Da WebAssign seit jeher eine automatische Korrektur dieser beiden Aufgabentypen vornehmen kann, wurde es bereits im Sommersemester 2002 zur Verwaltung und Korrektur der ersten beiden Übungshefte eingesetzt.

Die Aufgabenhefte 3 bis 7 der Print-Version enthalten hingegen zahlreiche Übungen, bei denen die Studierenden selbstentwickelte Schaltungen als Lösung abgeben müssen. Derartige Aufgaben konnten bisher von WebAssign lediglich verwaltet werden² - eine Unterstützung bei der Aufgabenkorrektur war jedoch bisher nicht möglich.

Erst mit Hilfe der hier vorgestellten Systemintegration konnte auch für diesen Aufgabentyp eine durchgängige und weitestgehend intuitiv bedienbare Plattform zwischen Studierenden, Korrektoren und Lehrenden geschaffen werden. Im Sommersemester 2003 wurde das modifizierte System erstmals im Lehrbetrieb eingesetzt. Zu diesem Zweck wurden die Aufgabenhefte 3 bis 7 des genannten Kurses nach WebAssign übertragen und die für die automatische Vorkorrektur der einzelnen Aufgaben benötigten Konfigurationsarbeiten vorgenommen. Die durchgeführten Arbeiten wurden nach Ab-

² Dabei wurde jede eingereichte Lösungsdatei als Black Box betrachtet, die von WebAssign an einen Korrektor durchgereicht wurde.

schluss im Rahmen einer Testphase überprüft.

Die Teilnehmer des Kurses wurden durch Informationsschreiben frühzeitig auf die neue Durchführungsform hingewiesen. Nur in begründeten Ausnahmefällen, beispielsweise bei Inhaftierung oder körperlicher Behinderung, konnten die Studierenden ihre Einsendungen auch weiterhin auf dem Postweg einreichen. Diese Möglichkeit wurde allerdings nur von 20 der circa 1.200 aktiv an den Übungen teilnehmenden Studierenden des Kurses beantragt.

Die übrigen Teilnehmer konnten sich die auf den Bedarf des Kurses abgestimmte HADES-Version zusammen mit den Vorlagen für die dazugehörigen Aufgaben von der Web-Seite des Kurses herunterladen. Dort wurden außerdem umfassende Bedienungsanleitungen zu HADES und WebAssign zur Verfügung gestellt. Die Mehrzahl der Teilnehmer hatte weder mit dem Herunterladen der lediglich 3,3 MB umfassenden Software noch mit der Installation des Programms HADES Probleme. Die an uns herangetragenen Schwierigkeiten bei der Installation und Inbetriebnahme konnten ausnahmslos auf Fehler in der Systemkonfiguration der von den Studierenden verwendeten Rechner, beispielsweise fehlende Pfadangaben, zurückgeführt werden.

Der Grossteil der Studierenden hatte wenig Probleme, sich in der neuen Systemumgebung zurecht zu finden. Im Gegensatz zur Pilotphase fanden sich jedoch auch einzelne Teilnehmer, die das neue System als schwer verständlich und umständlich erachteten. Gespräche mit mehreren dieser Teilnehmer haben ergeben, dass diese keinerlei Vorkenntnisse im Bereich digitaler Schaltungen besaßen und somit weder über die deutlich höhere Komplexität kommerzieller Schaltungssimulatoren noch über die Schwierigkeiten beim Schaltungsentwurf auf Papier im Bilde waren. Teilnehmer, die den Kurs als Wiederholer bearbeiteten beziehungsweise anderweitig

Erfahrung mit dem Entwurf von Digitalschaltungen sammeln konnten, äußerten sich hingegen weitgehend positiv. Die von ihnen herangetragenen Kritikpunkte lagen vorwiegend in kleineren, bei den vorangegangenen Tests nicht entdeckten Mängeln der Konfiguration, beispielsweise fehlende Empfangsbestätigungen oder unterbrochene Links, begründet. Diese Mängel wurden umgehend behoben.

Der Übungsbetrieb verlief weitgehend reibungsarm; lediglich die bereits angesprochenen Konfigurationsprobleme sorgten für ein im Vergleich zu den Vorjahren erhöhtes Kommunikationsvolumen zwischen Studierenden und Kursbetreuern. Die übrigen an die Kursbetreuung herangetragenen Fragen befassten sich – wie in den Vorjahren – primär mit inhaltlichen Aspekten.

Eines der vorrangigen Ziele bei der Einführung des beschriebenen Systems, die Gewährleistung eines schnellen Korrekturdienstes, konnte nur teilweise erreicht werden. Während die Korrektur der Aufgabenhefte zu den Kurseinheiten 5 bis 7 wie geplant sehr zügig verlief, konnte bei der vierten Einsendeaufgabe noch keine Verbesserung gegenüber dem Vorjahr erzielt werden. Die Korrektur des dritten Aufgabenheftes dauerte bei der Mehrzahl der Korrektoren sogar deutlich länger. Der Grund hierfür bestand in der anfänglich unzureichenden Vertrautheit der Korrektoren mit HADES beziehungsweise WebAssign. Offenbar genügt es nicht, den Korrektorkräften einen frühzeitigen Zugang zu den beiden Systemen zu ermöglichen, da auf diese Weise der bei der Korrektur stattfindende Arbeitsablauf nicht geeignet eingeübt werden kann. Stattdessen sollten die Korrektorkräfte anhand von alten Einsendungen beziehungsweise eigens zu diesem Zweck erzeugten Testdaten mit dem neuen Korrekturprozess vertraut gemacht werden.

Zusammenfassend kann man festhalten, dass sich die hier vorgestellte Systemintegration trotz der

für jedes neue System üblichen Kinderkrankheiten in der Praxis bewährt hat. Zwar bedeutet die Nutzung des Systems eine gewisse Einschränkung für die möglichen Aufgabentypen. Da aber weiterhin Text- und Multiple-Choice-Aufgaben möglich sind, mit denen weitere Aspekte der Kursinhalte abgefragt werden können, ist diese Einschränkung nicht gravierend. Sie erfordert lediglich eine Umgewöhnung des Kurs-Anbieters bei der Erstellung neuer Übungsaufgaben.

5.3 Evaluation des Verfahrens

Im Rahmen des ersten großen Einsatzes unseres Systems im Sommersemester 2003 wurde eine Evaluation durchgeführt. Hierbei standen die Akzeptanz des Systems, die in einzelnen Aufgaben erzielten Leistungen und die erzielte Gesamtleistung, d.h. der Erwerb eines Leistungsnachweises, im Mittelpunkt unserer Aufmerksamkeit. Soweit möglich, wurden die Daten des Sommersemesters 2003 mit denen des Sommersemesters 2002 verglichen, in dem der gleiche Kurs, allerdings ohne unser System, sondern mit konventionellen schriftlichen Übungen, durchgeführt worden war. Zu Möglichkeiten und Grenzen solcher Tests siehe zum Beispiel [16].

Die Akzeptanz versuchten wir quantitativ dadurch zu erfassen, dass wir die Anzahl der Einsendungen, jeweils als prozentualer Anteil der Kursteilnehmerzahl, beobachteten. Da die verpflichtende Einfüh-

rung Web-gestützter Übungen die einzige Veränderung war, sollte ansonsten wie in den Jahren zuvor ein etwa gleicher Anteil der Kursteilnehmer an den Übungen teilnehmen. Bild 3 stellt den prozentualen Anteil der Einsendungen beider Jahre dar. Wir hatten erwartet, dass dieser Anteil in den ersten Kurseinheiten (KE) geringer ist als im Vorjahr, da ein Teil der Kursteilnehmer von der Notwendigkeit abgeschreckt sein könnte, Software zu installieren und ihre Bedienung zu erlernen. Wir hatten gehofft, dass der Anteil weniger stark abfallen würde als im Vorjahr, d.h. dass die Studierenden besser „bei der Stange“ bleiben würden. Diese Hoffnung hat sich nur bedingt erfüllt. Allerdings hat sich die Kurve gegenüber dem Vorjahr auch nicht verschlechtert, so dass sich also nur zu Beginn eine Gruppe von Studierenden entschied, an den Übungen nicht teilzunehmen. Dies mag allerdings damit zusammenhängen, dass viele Studierende den Kurs auch als private Weiterbildung benutzen. Bei diesen ist die Toleranzschwelle gegenüber zusätzlichem Aufwand sehr viel niedriger als bei regulären Studierenden des Studiengangs.

Um die mit unserem System erzielten Leistungen zu bewerten, betrachteten wir detailliert die dritte Einsendung, da dort erstmalig Aufgaben mittels HADES zu lösen waren. Da uns die vergleichbaren Daten aus dem Sommersemester 2002 nicht vorlagen, verglichen wir die Aufgaben, die mittels HADES gelöst

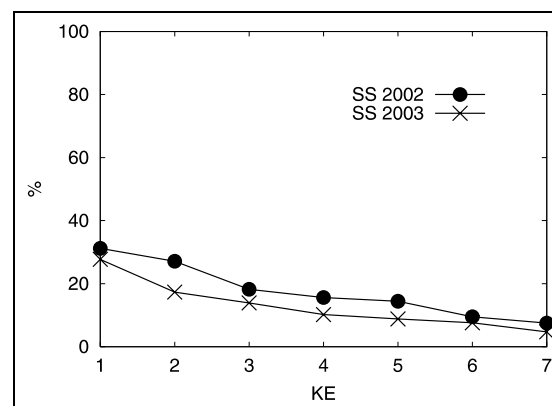


Bild 3 Anzahl der Einsendungen, als Anteil an der Kursteilnehmerzahl.

Tabelle 1 Auswertung der dritten Einsendung. ¹Maßeinheit: Anzahl Einsendungen. ²Maßeinheit: Erreichbare Punktzahl.

Aufg.	Aufgabentyp		Gesamtdaten		Lösungsversuche					
	HADES	Text/ MC	Erreichbare Punktzahl	Anzahl Einsendungen	0 Punkte ¹		Max. Punktzahl ¹		Durchschnitt ²	
					abs.	%	abs.	%	abs.	%
3-1	X		3	498	39	7,831	427	85,742	2,678	89,270
3-2	X		2	456	30	6,578	403	88,377	1,817	90,850
3-3		X	5	394	54	13,705	168	42,639	3,271	65,420
3-4	X		7	425	22	5,176	387	91,058	6,524	93,200
3-5	X		2	426	40	9,389	373	87,558	1,779	88,950
3-6		X	3	415	41	9,879	357	86,024	2,633	87,770
3-7		X	2	377	102	27,055	220	58,355	1,312	65,600
3-8	X		10	342	52	15,204	191	55,847	7,500	75,000
3-9	X		4	319	38	11,912	251	78,683	3,275	81,875
3-10		X	3	289	61	21,107	165	57,093	2,069	68,960
3-11		X	2	313	64	20,447	197	62,939	1,424	71,200
3-12		X	4	320	47	14,687	173	54,062	2,909	72,725
3-13		MC	3	366	131	35,792	104	28,415	1,389	46,300

wurden, mit denen, die mittels konventioneller Text- oder Multiple-Choice-Aufgaben gelöst wurden. Tabelle 1 zeigt die detaillierten Ergebnisse.

Es zeigt sich zunächst, dass im Mittel die HADES-Aufgaben von mehr Einsendern bearbeitet wurden als die konventionellen Aufgaben (411 Einsender im Mittel gegenüber 353). Des Weiteren zeigt sich, dass nach den drei angewandten Auswertungen die HADES-Aufgaben besser gelöst wurden. Bei HADES-Aufgaben erreichten im Mittel nur 9,35% der Einsender Null Punkte, verglichen mit 20,34% bei konventionellen Aufgaben. Die Maximalpunktzahl erreichten 81,21% beziehungsweise 55,65% der Einsender. Im Mittel wurden bei HADES-Aufgaben 86,52% der erreichbaren Punkte erzielt, verglichen mit 68,28% bei konventionellen Aufgaben. Uns sind bisher keine externen Gründe bekannt, aus denen heraus die Studierenden konventionelle Aufgaben eher ablehnen sollten als HADES-Aufgaben. Unserer Meinung nach sind zum Beispiel die konventionellen Aufgaben weder uninteressanter noch einfacher noch schwieriger als HADES-Aufgaben. Unter der Annahme, dass wirklich keine externen Gründe für das unterschiedliche Verhalten der Studierenden verantwortlich sind, zeigt sich also, dass die Verwendung von HADES die Studierenden bei der Lösung von Aufgaben gut unterstützt.

Schließlich haben wir die Ergebnisse der konventionell schriftlichen Abschlussklausuren beider Semester verglichen. Hier zeigt sich zunächst, dass die Anzahl der Klausurteilnehmer von 8,16% der Kursteilnehmer im Sommersemester 2002 auf 10,43% im Sommersemester 2003 angestiegen ist³. Im Sommersemester 2003 bestanden 58,50% der Klausurteilnehmer, verglichen mit 55,81% im Sommersemester 2002. Insgesamt haben also im Sommersemester 2003 6,10% der Kursteilnehmer einen Leistungsnachweis erworben, verglichen mit 4,56% der Kursteilnehmer im Sommersemester 2002. Dies entspricht einer Steigerung um etwa 33%. Uns sind keine externen Gründe bekannt, die bei den Studierenden zu verschiedenem Erfolg in den verschiedenen Semestern geführt haben könnten. Unter dieser Annahme zeigen die gemachten Erhebungen, dass die Einführung der Internet-gestützten Übungen mit HADES einen positiven Einfluss auf das Klausurverhalten und den Gesamterfolg der Studierenden gehabt hat.

6 Ausblick

Der Einsatz von interaktiven Web-gestützten Übungen ist im Kurs

³ Die insgesamt geringen Teilnahme- und Abschlussquoten sind darin begründet, dass die Mehrzahl der Studierenden in diesem Kurs keinen Leistungsnachweis erwerben muss, sondern eine mündliche Prüfung ablegt.

„Technische Informatik I“ der Fern-Universität mit dem Sommersemester 2003 von der Pilotphase in den Regelbetrieb übergegangen. Die bisherigen Erfahrungen im Regelbetrieb mit etwa 1.200 aktiven Teilnehmern sind durchweg gut, das System läuft stabil. Die Evaluation zeigt eine Verbesserung der Lernergebnisse gegenüber dem Vorjahr.

Das System der interaktiven Web-gestützten Übungen wird nun im Rahmen mehrerer Projekte (z. B. dem IngMedia-Projekt [6], siehe auch www.ingmedia.de) auf die Folgekurse „Technische Informatik II“ (Mikrorechnersysteme) und „Technische Informatik III“ (Rechnerarchitektur) sowie das Mikrorechnerpraktikum ausgedehnt. Hierbei wird neben dem Einsatz von Entwurfswerkzeugen und Simulationen auch ein Zugang zu fernsteuerbaren Versuchsaufbauten mit realen Digitalschaltungen über das Web realisiert, sodass zwar die Lehrvermittlung und der Zugang, nicht aber das Experiment selbst, das zum Beispiel im Messen von Signalfolgen mit einem Logikanalysator besteht, virtualisiert werden.

Wir hoffen, auf diese Weise einige Nachteile der Fernlehre gegenüber der Präsenzlehre beseitigen zu können, und darüber hinaus den Studierenden einen konstruktiven Umgang mit dem Stoff des Grundstudiums in Technischer Informatik ermöglichen zu können.

Literatur

- [1] D.L. Andrews, M.A. Thornton: Integration of CAD Tools and Structured Design Principles in an Undergraduate CE Curriculum. In: *IEEE Computer Society TC Computer Architecture Newsletter* Feb. (1999), pp. 8–9.
- [2] H. Bähring, J. Keller, W. Schiffmann: Einsatz von neuen Medien an der FernUniversität Hagen. In: *Informationstechnik und Technische Informatik*, 43 (2001) S. 215–218.
- [3] J. Brunsmann, A. Homrighausen, H.W. Six, J. Voss: Assignments in a Virtual University – The WebAssign-System. In: *Proc. 19th World Conf. on Open Learning and Distance Education*. Wien June 1999.
- [4] J. Drordjevic, A. Milenkovic, N. Grbanovic: An Integrated Environment for Teaching Computer Architecture. In: *IEEE Micro* 20(3) (2000) pp. 66–74.
- [5] EDO-Workspace. <http://ews.uni-dortmund.de>.
- [6] J. Fricke, W. Schiffmann: Telematiklabor für Digitalschaltungen. In: *Informatik 2001, Jahrestagung der GI/OCG*. Tagungsband 2. Wien, 2001. S. 1149–1153.
- [7] N. Hendrich: HADES: The Hamburg Design System. In: *SA'98, European Academic Software Award/Alt-C Conf.* Oxford, 19.–21. Sep. 1998.
- [8] S. Herschel: Goya – Vom Programmierwettbewerb zum Verwaltungssystem. In *Informatik 2003 – Innovative Informatikanwendungen*. Band 2. Frankfurt, 2003. S. 96–101.
- [9] B. Hüsemann, J. Lechtenböcker, G. Vossen, P. Westerkamp: XLX – A Platform for Graduate-Level Exercises. In: *Proc. ICCE 2002 – Int'l Conf. on Computers in Education*. Auckland, New Zealand, 2002. pp. 1262–1266.
- [10] ILIAS: Integriertes Lern-, Informations- und ArbeitskooperationsSystem. <http://www.ilias.uni-koeln.de>.
- [11] N. Kapadia, R. Figueiredo, J. Fortes: PUNCH: Web Portal For Running Tools. In: *IEEE Micro* 20(3) (2000) pp. 38–47.
- [12] D.J. Lilja: Education at a Distance: A Report From the Front. In: *Workshop on Computer Architecture Education*. Orlando, FL, Jan. 1999.
- [13] A. Milenkovic, B. Nikolic, J. Djordjevic: CASTLE: Computer Architecture Self-Testing and Learning System. In: *Workshop on Computer Architecture Education*. Anchorage, AL, May 2002.
- [14] Online-Campus. <http://www.online-campus.net>.
- [15] RealVNC Homepage. <http://www.realvnc.com>.
- [16] H.W. Six, G. Ströhlein, J. Voss: Evaluation of WebAssign. In: *20th World Conf. on Open Learning and Distance Education (ICDE Congress)*. Düsseldorf, Apr. 2001.
- [17] V. Tschertter, R. Lamprecht, J. Nievergelt: Exorciser: Automatic Generation and Interactive Grading of Exercises in the Theory of Computation. In: *4th International Conference on New Educational Environments*. Lugano, May 2002, pp. 47–50.



1



2



3

1 Dipl.-Inform. Udo Hönig studierte Informatik an der FernUniversität Hagen und erreichte 2001 seinen Abschluss als Diplom-Informatiker. Seit 2002 ist er als wissenschaftlicher Mitarbeiter am Fachbereich Informatik der FernUniversität Hagen tätig. Adresse: FernUniversität Hagen, FB Informatik, Postfach 940, 58084 Hagen, Deutschland, E-Mail: Udo.Hoenig@FernUni-Hagen.de

2 Prof. Dr. Jörg Keller ist Inhaber des Lehrgebiets Parallelität und VLSI an der FernUniversität Hagen. Seine Forschungsschwerpunkte betreffen parallele Algorithmen für kryptologische Anwendungen, fehlertolerante Rechensysteme und Neue Medien in der Lehre. Adresse: FernUniversität Hagen, FB Informatik, Postfach 940, 58084 Hagen, Deutschland, E-Mail: Joerg.Keller@FernUni-Hagen.de

3 Prof. Dr. Wolfram Schiffmann ist Inhaber des Lehrgebiets Rechnerarchitektur an der FernUniversität Hagen. Seine Forschungsschwerpunkte betreffen Scheduling-Algorithmen für parallele Rechnerarchitekturen, Gridcomputing, Telematiklabore und Neue Medien in der Lehre. Adresse: FernUniversität Hagen, FB Informatik, Postfach 940, 58084 Hagen, Deutschland, E-Mail: Wolfram.Schiffmann@FernUni-Hagen.de