Didaktisches Design für eine Online-Programmierausbildung

Prof. Gudrun Görlitz
Dipl.-Kommunikationswirt Stefan Müller

Technische Fachhochschule Berlin
Projekt Virtuelle Fachhochschule
Luxemburger Str. 10
13353 Berlin
http://vfh.tfh-berlin.de

Abstract: Im Rahmen des BMBF-Leitprojektes „Virtuelle Fachhochschule für Technik, Informatik und Wirtschaft“ wurde für den Bachelor-Studiengang Medieninformatik der Kurs "Grundlagen der Programmierung I" an der Technischen Fachhochschule Berlin entwickelt. Bereits bei der Konzeption dieses Online-Kurses zeigte sich deutlich, dass die traditionellen didaktischen Konzepte der Präsenzlehre oder des Fernstudiums nicht uneingeschränkt auf ein virtuelles Studium übertragbar sind. Komponenten des didaktischen Designs für eine Online-Programmierausbildung wurden nach dem best practice Prinzip bestimmt und Generatoren zu deren Implementierung entwickelt.

1 Einleitung

Programmieren lernt man, indem man das Wissen über die Programmierkonzepte sowie die Grammatik der jeweiligen Programmiersprache in viele eigene Programme umsetzt. Der vernetzte Personalcomputer ist das wichtigste Werkzeug für die Programmierausbildung. Zum Entwerfen, Editieren, Kompilieren und Ausführen der Programme bedienen sich alle Studierenden aktueller Software auf ihrem Computer. Die Hochschullehre in der Programmierausbildung weist jedoch eine geringe Einbeziehung der neuen Medien auf. Meist werden die Inhalte in Form von HTML- oder PDF-Skripten, um Beispielprogrammcodes ergänzt, auf einem Server zum Herunterladen angeboten. Lediglich komplexe Algorithmen wie Listen- oder Baumstrukturen sind in Form von Animationen visualisiert. Im Fachkollegenkreis wird die Meinung vertreten, dass eine multimediale Programmierausbildung nicht möglich und auch nicht sinnvoll sei.

In dieser Ausgangssituation startete im Jahr 2000 ein interdisziplinäres Team an der Technischen Fachhochschule Berlin mit der Entwicklung eines online studierbaren Kurses "Grundlagen der Programmierung I". Dieses Vorhaben ist ein Arbeitspaket des BMBF-Leitprojekts "Virtuelle Fachhochschule".

2 Didaktisches Konzept

Das didaktische Konzept des Kurses "Grundlagen der Programmierung I" orientiert sich konsequent an der Zielgruppe: Studierende, die selbstgesteuert über das Internet Programmieren erlemen wollen. Diese Grundlagenveranstaltung richtet sich an Studienanfänger, deren Vorwissen erfahrungsgemäß sehr unterschiedlich ist. Das Angebot von Wiederholungsstoff zum Schließen vorhandener Lücken oder von Vertiefungswissen für besonders interessierte Studierende unterstützt den individuellen Lernprozess. Inhaltlich steht die problemorientierte Stoffvermittlung im Vordergrund.

Das Berufsumfeld der Informatikerin und des Informatikers ist durch interdisziplinäre Teamarbeit gekennzeichnet. Im Kurs "Grundlagen der Programmierung I" werden die Studierenden zur Gruppenarbeit aufgefordert. Programmlösungen sind zu präsentieren und auszutauschen. Das Dokumentieren von Programmcode sowie das Erstellen von Dokumentationen - eine zwar unbeliebte aber notwendige Tätigkeit im Team - wird eingeübt. UML ist bereits in diesem Anfängerkurs ein hilfreiches Werkzeug zur grafischen Beschreibung von Klassenhierarchien.[1]

Übungen sind in die Stoffpräsentation integriert. Die oftmals hinderliche Trennung von Vorlesung und Laborübung in der Präsenzlehre konnte dadurch erfolgreich aufgebrochen werden. Interaktive Wissensprüfungen dienen der Selbstkontrolle und geben dem Studierenden Rückmeldungen über seinen Lernfortschritt oder seine Lerndefizite. Komplexe Programmieraufgaben sind individuell oder in Gruppen zu bearbeiten und als Einsendeaufgaben an Mentoren zur Beurteilung einzusenden. In den Online-Kurs „Grundlagen der Programmierung I“ sind zwei Präsenzphasen integriert. Während der Präsenz wird eine komplexe Aufgabe, die alle bisherigen Lernziele berücksichtigt, in der Gruppe bearbeitet und von den Mentorinnen und Mentoren im Computerlabor an der einschreibenden Hochschule betreut.

Neben der Vermittlung der Programmierkonzepte wird der Umgang mit Entwicklungswerkzeugen, wie Editoren und Java-Entwicklungsumgebungen, geübt. Informationsquellen aus dem Internet sind in den Kurs eingebunden.
Die Voraussetzung für ein erfolgreiches, selbstgesteuertes Studieren sind nachvollziehbare Lernziele sowie Vorgaben zur Lernzeit. Im Kurs "Grundlagen der Programmierung I" sind die Lernziele auf übersichtliche Teilziele heruntergebrochen. Zeitlich wurde der Stoff so skaliert, dass ein Lernen im Wochenrhythmus unterstützt wird.

Die Integration von Ton verfolgt verschiedene Zielrichtungen. Der Studierende soll die verbale Beschreibung von Programmierproblemen erlernen. Er erfährt, wie Quellcode gesprochen wird. Der akustisch orientierte Lernertyp wird motiviert.

Der Kurs "Grundlagen der Programmierung I" ist integriert in das Curriculum des Studiengangs Medieninformatik (Bachelor) der Virtuellen Fachhochschule (VFH) und wird im ersten Semester angeboten. Er umfasst Lernstoff, der 4 Semesterwochenstunden

[1] in Anlehnung an [Bal99], Seite 104 ff.

in der Präsenzlehre entspricht und wird mit 5 Credit Points nach ECTS bewertet. Ein Fortsetzungskurs "Grundlagen der Programmierung II" wird zur Zeit produziert. Die Kurse werden an der VFH als Module bezeichnet. Die Einbindung des Moduls "Grundlagen der Programmierung I" in den Studienbetrieb und das Prüfungsverfahren an den beteiligten Hochschulen regelt die bundesländerübergreifende Studien- und Prüfungsordnung.

2.1 Strukturierung des Moduls

Der Lernstoff ist in Lerneinheiten strukturiert, die nacheinander wöchentlich zu bearbeiten sind. Die Bearbeitungsreihenfolge der inhaltlich abgeschlossenen Lerneinheiten, kann den didaktischen und zeitlichen Erfordernissen angepasst werden. Z. B. ergab sich im Einsatz, dass einige Themen früher als andere behandelt wurden oder im kürzeren Sommersemester eine Einheit weniger angeboten wurde.

Nr.	Titel	Zeitbedarf	Übungen	Tests	Einsenden
LE01	Einführung	90	3	1	-
LE02	Programmiersprachen	90	1	3	-
LE03	Java	90	0	5	-
LE04	Das erste Java-Programm	120	8	5	-
LE05	Applets	120	17	-	-
LE06	Einfache Typen	120	9	4	-
LE07	Präsenz	240	1	-	-
LE08	Methoden	120+ 120	2	3	1
LE09	Sequenz und Selektion	90 + 240	2	4	2
LE10	Iterationen	90 + 180	4	3	2
LE11	Paketstrukturen	120	8	5	-
LE12	Ausnahmen	120	6	3	-
LE13	Präsenz	240	1	-	-
LE14	Vererbung	120 + 240	9	3	2
LE15	Reihungen	90 + 240	4	2	2
LE16	Zeichenketten	120 + 180	6	3	2
LE17	Klausur	120	-	-	-
		2100 + 1200	**81**	**44**	**11**

Abb. 2.1: Merkmale der Lerneinheiten des Moduls Programmieren I

2.2 Strukturierung der Lerneinheiten

Alle Lerneinheiten beginnen mit der Angabe der Lernziele und der benötigten Arbeitszeit sowie dem inhaltlichen Überblick. Sie enden mit der vertonten Zusammenfassung der prüfungsrelevanten Inhalte, einer Fortschrittskontrolle sowie Programmierübungen, die zur individuellen Korrektur an Mentoren gesandt werden müssen. Der Stoff ist zeitlich so portioniert, dass für die Bearbeitung der Lerneinheit etwa 2 Stunden und für die Einsendeaufgaben etwa 2 bis 4 Stunden benötigt werden. Diese Struktur ermöglicht Berufstätigen ein effizientes Zeitmanagement. [Zi01]

Eine individuelle Steuerung des Lernprozesses wird durch die vertikale Drei-Ebenen-Struktur unterstützt. Auf der ersten Ebene verschafft sich der Studierende einen Überblick über den Lernstoff, um zu entscheiden, ob er diese Lerneinheit bearbeiten möchte. Die zweite Ebene umfasst den für die Prüfung relevanten Lernstoff. Auf der dritten Ebene finden sich Vertiefungsstoff für den interessierten Studierenden sowie Wiederholungen für Studierende mit geringem Vorwissen. Dieser Stoff ist nicht prüfungsrelevant.

2.3 Navigation

Die Navigation muss bei der beschriebenen Struktur unterschiedliche Anforderungen erfüllen. Unterschieden wird zwischen Modul- und Lerneinheitenebene.

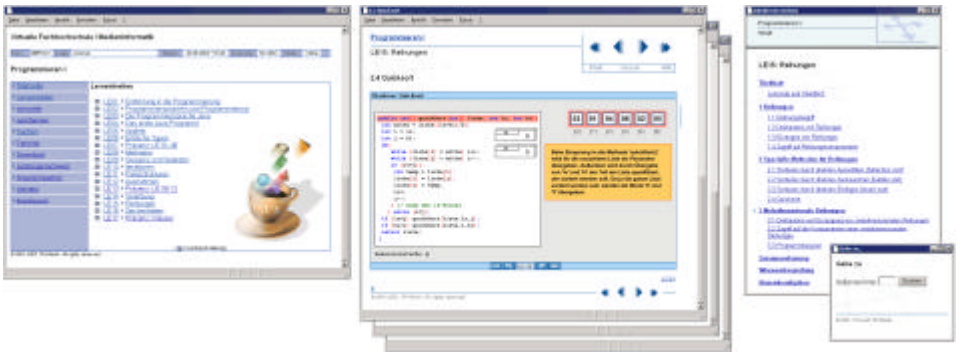

Abb. 2.2: Screenshots der Modul- und Lerneinheitenebene

Auf Modulebene stehen dem Nutzer neben den Lerneinheiten übergreifende Funktionen und Informationen zur Verfügung. Informationen, wie Lernziele, Lernformen, Literaturhinweise, Termine und Kurzbeschreibungen, geben Hinweise für den Umgang mit dem Modul und verdeutlichen den Aufbau. Bereiche, wie Download, Volltextsuche, Verzeichnisse der Übungen, Animationen und Abbildungen dienen als Werkzeuge für das gezielte Suchen nach Informationen in den Lerneinheiten. Für die Modulebene hat sich eine einfache Hypertextnavigation mit Links als ausreichend erwiesen.

In der Lerneinheit stehen verschiedene Navigationsarten zur Verfügung. Primär - und somit jederzeit verfügbar - wird der, durch die didaktische Konzeption vorgesehene, lineare Pfad angeboten. Die Inhalte sind gegliedert in entsprechend nummerierte Kapitel und Abschnitte. Die Navigation entspricht dem Blättern in Seiten, wie man es von Büchern kennt. Sekundär bietet die Anlehnung an die Buch- und Lexikonmetapher[2] die Möglichkeit zur Navigation über ein Inhaltsverzeichnis und über die Seitenzahlen. Durch die Verwendung von Server-Side Includes wurde technisch der Inhalt von der Navigation getrennt. Ein Parser erzeugt aus dem aktuellen Inhalt das entsprechende Inhaltsverzeichnis sowie die korrekten, verlinkten Seitenzahlen. Bei Bedarf kann aus dem aktuellen Datenmaterial eine offline lauffähige Version erzeugt und komprimiert zum Download angeboten werden. Eine Überführung des Datenmaterials in XML wird zur Zeit entwickelt.

[2] vgl. [Sc02] S. 54f.

3 Aufbereitung der Inhalte

Der Stoff wurde problemorientiert für Medieninformatiker aufbereitet. Die textuelle Aufbereitung, die Visualisierung durch Grafik und Animation sowie Vertonungen sind die wichtigsten Elemente bei der Erstellung dieser Lerneinheiten. Neben dem linearen Hauptpfad werden immer wieder Links zu ausgearbeitetem Vertiefungswissen oder zusätzlichen Informationen, wie z. B. die Unicode-Tabelle, angeboten. Stets wiederkehrende Elemente sind die virtuellen Studenten Melanie und Markus, die in der Lerneinheit Fragen stellen, die typisch für Präsenzlehrveranstaltungen sind. Fragen und Antworten sind als Audio im Streaming-Format hinterlegt. Durch einen Player sind die geladenen Dateien wiederholt abspielbar. Neben der Antwort selbst, wird dem Studierenden ein Gefühl für die Verbalisierung von Programmierproblemen vermittelt.

3.1 Text

Bei der textuellen Umsetzung wurden inhaltliche Strukturen durch typografische und gestalterische Mittel, wie Absätze, Textkästen mit Reitern, Hervorhebungen und Charts, sichtbar gemacht. Sie dienen der Übersichtlichkeit und erlauben ein selektives Lesen. Die Visualisierung mit Mitteln der Typografie ist heute bei allen didaktisch aufbereiteten Lernmaterialien üblich.[3] Dem Lernenden steht zusätzlich ein lerneinheitenübergreifendes Glossar zur Verfügung. Dieses wird in einem eigenen Fenster dargestellt und kann selbstständig erforscht werden. Die Glossarbegriffe sind untereinander verlinkt und bieten weitere Links zu ihren Themengebieten im WWW an.

3.2 Grafik und Animationsarten

Neben Illustrationen und Abbildungen wurden, zur Veranschaulichung komplexer Zusammenhänge und zum interaktiven Erarbeiten von Lerninhalten, verschiedene Animationsarten bestimmt und die zugehörigen Generatoren sowie Player implementiert. Die Unterscheidung der Animationsarten unterstützt die Erwartungskonformität der Nutzer an die gezeigte Visualisierung. Die aufgeführten Begriffe wurden aus dem gleichen Grund gewählt und sind nicht immer wissenschaftlich eindeutig, erzeugen aber die gewünschte Erwartung an die Bedienung.

Rolloverbilder
ermöglichen die explorative Wissensaneignung: Beim Überfahren des Bildes mit dem Cursor werden Lerninhalte sichtbar. Diese Animationsart wurde unter anderem für Erläuterungen von Quellcode mit Hilfe fliegender Fenster implementiert. Die Studierenden bestimmen selbst, zu welchen Programmzeilen sie Erklärungen benötigen. Die jeweilige Erläuterung erfolgt unmittelbar an der Quellcodezeile und nicht erst Seiten später, wie es in Büchern praktiziert wird. Für das Erschließen von Klassenhierarchien und das Erlernen der Wirkungsweise der Kontrollstrukturen sind Rolloverbilder ebenfalls gut geeignet.

Diashows
sind ein Anzahl von statischen Abbildungen (Dias) deren Übergänge (Show) animiert sind. Die didaktisch vorgesehenen Pausen dienen zur Visualisierung von

[3] vgl. [Ba99] S. 130.

Entwicklungsschritten und deren Beschreibung. Die Verweildauer bei den einzelnen Dias wird durch den Benutzer bestimmt.

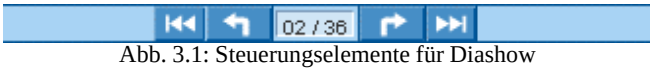

Abb. 3.1: Steuerungselemente für Diashow

In der Programmierung ist das Abarbeiten eines Programms ein Prozess, der oftmals der Veranschaulichung bedarf. Auch zur Visualisierung von Algorithmen, beispielsweise der Sortierverfahren, sind Diashows gut geeignet. Die umfangreichste Diashow zeigt den Entwurf und die Implementierung eines Programms, das die Zeit zum Rasenmähen ermittelt; präsentiert mit UML-Darstellung, Programmcode sowie einer Visualisierung der Programmausführung in 89 Dias mit Darstellung der Parameterübergabe und der Speicherorganisation.
Im VFH-Team wurde ein Player für die Diashows entwickelt. Auch Kombinationen der Diashow mit Selektionsmöglichkeit bei Benutzereingaben wurden realisiert.

Animationen
dienen ebenfalls zur Visualisierung von Prozessen. Gegenüber der Diashow wird die Animation jedoch ohne Pause abgespielt.

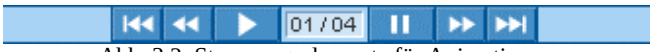

Abb. 3.2: Steuerungselemente für Animationen

Einzelne Szenen können angesteuert und somit auch szenenweise angesehen werden. Didaktisch gewollte Pausen stehen bei dieser Animationsart nicht zur Verfügung. Interaktive Eingaben seitens der Studierenden sind jedoch integrierbar.

Interaktionen
bezeichnen Animationen, die eine Eingabe durch den Benutzer erwarten. Dies kann eine Zuordnung mittels Drag und Drop sein aber auch die Auswahl oder die Eingabe bestimmter Parameter auf welche die Interaktion entsprechend reagiert und das Ergebnis anzeigt. Die Konzeption und Produktion von Interaktionen ist meist komplex und dadurch aufwendig. Für ausgewählte Themen wurden Interaktionen für das Modul entwickelt, beispielsweise:

- Interaktives Kaffeekochen, um den Algorithmus-Begriff zu erfassen,
- der Aufbau des Programmiersprachenbaums nach Benutzereingaben,
- die Modifikation der Bedingungen in geschachtelten Schleifen und Visualisierung des jeweiligen Ergebnisses.

3.3 Übungen

Zur Aktivierung der Studierenden[4] wurden in die Stoffpräsentation Übungsaufgaben mit Selbstkontrollmöglichkeiten durch die Anzeige des Ergebnis-Screenshots und der Programmlösung integriert. Der Programmcode aller Beispielprogramme kann aus dem Text herauskopiert oder heruntergeladen und sofort getestet werden. Die Darstellung des

[4] vgl. [Ke01] S. 182

Programmcodes entspricht den Farbkonventionen, wie sie die Studierenden aus dem Editor kennen.

Für die individuelle **Fortschrittskontrolle** mit sofortiger Auswertung am Ende jeder Lerneinheit wurden Aufgabenformen, wie Lückentext, Drag-and-Drop-Aufgaben, Multiple-Choice und Freitextaufgaben spezifiziert und Generatoren für deren Implementierung entwickelt. Diese beliebten Übungsaufgaben[5] dienen dem Studierenden als spielerischer Selbsttest. Die Antworten können sofort überprüft werden. Neben den Zusammenfassungen weisen sie zudem auf eine mögliche Fragestellung in der abschließenden Klausur hin.

Lückentext
Nach dem Füllen des Lückentextes per Drag-and-Drop erfolgt zunächst eine Fehleranalyse. Danach ist ein erneuter Versuch oder die Anzeige der korrekten Lösung möglich.

Drag-and-Drop-Aufgaben
Buchstaben sind mittels Drag-and-Drop so zu positionieren, dass Sachverhalte korrekt in Beziehung zueinander gesetzt werden. Tooltips mit Hinweisen auf die korrekte Lösung sind integrierbar. Nach dem Abarbeiten aller Fragen erfolgt zunächst eine Richtig-Falsch-Auswertung. Nach einem erneuten Versuch kann die korrekte Antwort angezeigt werden.

Multiple-Choice-Aufgaben
Hier stehen verschiedene Varianten zur Auswahl. Die dargestellten Aussagen werden durch die Kennzeichnung als richtig oder falsch markiert oder anderen Aussagen zugeordnet. Einfach- und Mehrfachauswahl sind möglich.

Freitexteingaben
sind in der Programmierung für solche Aufgabenstellungen geeignet, bei denen im Programmcode Variableninhalte nachzuvollziehen und deren Werte einzugeben sind.

3.4 Einsendeaufgaben

Für das Modul Programmieren I wurden 12 Einsendeaufgaben entwickelt, die von den Studierenden bearbeitet und per EMail an die Mentorinnen und Mentoren geschickt werden. Im Gegensatz zu den Übungen wird eine Musterlösung erst nach dem Einsendetermin bereitgestellt. Einsendeaufgaben ermöglichen eine individuelle Betreuung der Studierenden und liefern wertvolle Erkenntnisse darüber, in wieweit Lernziele erreicht wurden. Die Aufgaben sind Bestandteil einzelner Lerneinheiten und sind inhaltlich auf diese abgestimmt. Ein Satz alternativer Einsendeaufgaben ermöglicht eine Anpassung an das Niveau der Studierenden.

[5] Ergebnis aus der Evaluation. Vgl. [TA01].

4 Praktische Erfahrungen aus der Lehre

Das Modul „Grundlagen der Programmierung I“ wurde im Wintersemester 01/02 an sechs und im Sommersemester 02 an drei Hochschulen im Studium an der Virtuellen Fachhochschule eingesetzt. Die positive Resonanz seitens der Studierenden sowie der Mentorinnen und Mentoren bestätigte, dass das Programmieren mit diesem didaktischen Konzept effizient über das Internet erlernbar ist. Kritisiert wurden einzelne Einsendeaufgaben, die aus der Sicht der Studierenden eine zu lange Bearbeitungszeit erforderten. Ein Schwerpunkt der Weiterentwicklung des Moduls liegt deshalb im Bereich der Online- und Präsenzbetreuung. Die Programmieraufgaben für die Studierenden sollen in kleinere markant beschriebene Teilaufgaben strukturiert sowie in Muss- und Kann-Aufgaben unterschieden werden. Die Veröffentlichung mehrerer Musterlösungen für die Einsendeaufgaben dient zum explorativen Selbststudium.

Ein weiterer Schwerpunkt ist die Verbesserung der Kommunikation während der Online-Betreuung. Die bekannten Kommunikationsformen wie Newsgroups, Chats und Video Conferencing unterstützen die Kommunikation bei der Programmentwicklung nur unzureichend. Werkzeuge werden benötigt, die die Funktionalität von Audiochat, Whiteboard und Application Sharing mit gesteuerter Freigabe der Studierenden-Programmierentwicklungsumgebungen vereinen.

Einen Einblick in den Kurs "Grundlagen der Programmierung I" bietet der Demokurs, der unter http://www.vfh.de/prog1demo ohne Passwort aufgerufen werden kann.

Literaturverzeichnis

[Ba99] Ballstedt, S.P.: Texte visualisieren. In: Jakobs, E.; Knorr, D. (Hrsg.) Textproduktion und Medium; Bd. 5. Lang, Frankfurt am Main 1999, S. 129-140.

[Bal99] Balzert, H.: Lehrbuch Grundlagen der Informatik. Spektrum, Akad. Verlag, Heidelberg, Berlin 1999.

[Ke01] Kerres, M.: Multimediale und telemediale Lernumgebungen, 2., vollst. überarb. Aufl.. Oldenbourg, München, Wien 2001.

[Sc02] Schulmeister, R.: Grundlagen hypermedialer Lernsysteme, 3. korrigierte Auflage. Oldenbourg, München, Wien 2002.

[TA01] Thillosen, A; Arnold, P.: Entwicklung virtueller Studienmodule im Rahmen des Bundesleitprojekts "Virtuelle Fachhochschule für Technik, Informatik und Wirtschaft" – Evaluationsergebnisse. In: Wagner, E.; Kindt, M. (Hrsg) Virtueller Campus. Waxmann, Münster 2001, S. 402-410.

[Zi01] Zimmer, G.M.: e-Learning – Die 10 wichtigsten Anforderungen an die didaktische Gestaltung eines aufgabenorientierten e-Learning. In: LIMPACT 4, August 2001, S. 3.