„Ja, Nein, Jein“

Erweiterte elektronische Tests auf Basis von Microsoft SharePoint

Eva Altenbernd-Giani[1], Anna Lea Dyckhoff[2], Andreas Hackelöer[2], Anne-Christien Pielka[2] und Philipp Rohde[2]

[1] Lehr- und Forschungsgebiet Informatik 9, RWTH Aachen University
giani@informatik.rwth-aachen.de
[2] Centrum für integrative Lehr- und Lernkonzepte, RWTH Aachen University
{dyckhoff,hackeloeer,pielka,rohde}@cil.rwth-aachen.de

Zusammenfassung. Elektronisch gesteuerte und automatisch auswertbare Tests werden immer häufiger in der Lehre eingesetzt. Hierfür steht eine Vielzahl von Systemen zur Verfügung, die allerdings in ihrer Flexibilität und den angebotenen Fragetypen beschränkt sind. Es wird ein Ansatz vorgestellt, der es Lehrenden ermöglicht, eine Testfrage als Kombination aus mehreren Fragetypen zu erstellen und der in das bestehende Lehr- und Lernportal L^{2}P der RWTH Aachen University integriert ist, welches auf dem Microsoft Office SharePoint Server 2007 basiert.

1 Einleitung

Die Durchführung von elektronisch gesteuerten Tests wird in der Lehre mit verschiedenen Zielsetzungen eingesetzt [1]. Da die reine Präsentation von Informationen für einen gesicherten Lernerfolg nicht ausreicht, muss der Lernende sich selbst aktiv mit dem Lernstoff beschäftigen. Hierzu können elektronisch gesteuerte, *formative* Tests verwendet werden, die individuelle, zeitnahe Rückmeldungen zu den Antworten des Lernenden geben und ihm somit die Steigerung seiner Lernleistungen ermöglichen. Durch angebotene Selbsttests kann der Lernende den eigenen Lernfortschritt nachvollziehen und motiviert werden. Weiterhin werden *summative* Tests für die Bewertung von Leistungsständen herangezogen. Für die Lehrenden stellen solche Tests nicht nur ein geeignetes Instrument zur Kontrolle und Beurteilung des Wissensstandes der Lernenden dar, sondern können ebenfalls dazu dienen Vorwissen zu ermitteln, Wissensdefizite aufzudecken und die eigene Lehre zu evaluieren.

Die elektronische Durchführung birgt die Vorteile des Einsatzes multimedialer Elemente (z. B. Film, Simulation), automatischer Bewertungsroutinen, einer transparenten Darstellung einzelner Testprozesse und -ergebnisse sowie statistischer Auswertungsmöglichkeiten für Lehrende und Lernende und damit einer Reduktion des Verwaltungsaufwandes. Durch diese Reduktion können Tests, insbesondere bei der Nutzung von Fragenpools, häufiger und regelmäßiger angeboten werden.

Im Folgenden werden bestehende Ansätze zur Erstellung und Durchführung elektronischer Tests und ein neues, darüber hinausgehendes, an der RWTH Aachen University entwickeltes Konzept dargestellt.

2 Bestehende Ansätze

Ein Test besteht aus einer Menge von Aufgaben bzw. Fragen, die in unterschiedlichen Strukturen bzw. Sequenzen vom Lernenden bearbeitet werden. Die computerunterstützte Auswertung von Lösungen solcher Tests bedingt, dass die Richtigkeit der Lösungen algorithmisch festgestellt werden kann, damit entsprechende Rückmeldungen gegeben oder eine Bewertung vorgenommen werden kann. Dabei sind Lehrende bei der Wahl der automatisch durch Computer auswertbaren Fragetypen eingeschränkt. Geeignet sind Fragetypen, die *objektiv* auswertbar sind. Bei Fragen dieser Art muss die korrekte Lösung im Voraus konkret bekannt sein (vgl. [2]), wie es beispielsweise bei Aufsätzen nicht der Fall ist. Nach [3] kann zwischen Fragen unterschieden werden, bei denen der Lernende die korrekte Lösung nicht selbst formuliert, sondern aus einem Angebot auswählt (z. B. Multiple-Choice, Zuordnungsaufgaben) und andererseits solchen, bei denen die korrekte Lösung dem Lernenden nicht angezeigt wird (z. B. Lückentexte).

Nach [4] variieren Softwaresysteme für computerunterstützte Tests darin, wie sie das Design, die Bereitstellung und die Analyse von Online-Tests unterstützen und unterscheiden sich bezüglich ihrer Flexibilität und Skalierbarkeit. [2] nennen vier Komponenten, die die Mehrzahl solcher Systeme aufweist: eine Umgebung zur Erstellung von Testfragen, eine Komponente zur Verwaltung von Fragen, eine Umgebung zur Bereitstellung und Beantwortung von Fragen, und schließlich Werkzeuge zur Auswertung und Analyse.

Je nach Ausprägung dieser Komponenten lassen sich Systeme, mit deren Hilfe computerunterstützte Tests durchgeführt werden können, weiter unterscheiden. *Stand-Alone-Anwendungen*, die z. B. für Sprachkurse Verwendung finden, müssen auf Computern installiert oder von Datenträgern abgespielt werden. Die Testergebnisse werden dabei meist nicht an Lehrende weitergegeben und dienen somit den Lernenden zur Beurteilung des eigenen Lernfortschritts. In den letzten Jahren wurden jedoch vermehrt computerbasierte Tests auf webbasierte Lösungen verlagert. So bieten heutzutage viele *Autorenwerkzeuge* zur Erstellung von Web Based Trainings (WBTs), wie z. B. Adobe Captivate[3], die Möglichkeit, zusätzlich Tests in verschiedene Lehreinheiten zu integrieren. Daneben gibt es Systeme, die allein für die Erstellung und Abwicklung von Tests konzipiert sind (z. B. Hot Potatoes[4], Questionmark Perception[5]). Darüber hinaus bietet eine Vielzahl von *Lernmanagement-Systemen* (LMS), wie beispielsweise imc CLIX[6], Moodle[7] und Blackboard[8] Komponenten zur Erstellung, Beantwortung und Benotung von Tests an. Durch die Integration stehen den Lernenden und Lehrenden Bewertungen sowie statistische Auswertungen von Tests zur Verfügung. Die meisten dieser Systeme modellieren Tests als Sammlungen einzelner Fragen, wobei diese jeweils genau einem Fragetyp entsprechen.

Daneben existiert eine Vielzahl von Systemen zur Erstellung und Durchführung *fachspezifischer* Tests (z. B. [5], [6], [7]). In diesen werden vielfach spezielle Aufgaben-

[3] `www.adobe.com/products/captivate` (07.11.2008)
[4] `www.hotpotatoes.de` (07.11.2008)
[5] `www.questionmark.com` (07.11.2008)
[6] `www.im-c.de` (07.11.2008)
[7] `moodle.de` (07.11.2008)
[8] `www.blackboard.com` (07.11.2008)

und Fragetypen automatisch bewertet. Diese Einzellösungen sind allerdings meist mit hohem Entwicklungsaufwand verbunden und nur selten in Lernmanagement-Systeme integriert, was für Lehrende und Lernende bedeutet, eventuell mit verschiedenen Systemen arbeiten zu müssen. Sie zeigen aber auch, dass die klassischen Fragetypen, wie z. B. Multiple-Choice und Drag & Drop, in vielen Disziplinen nicht ausreichen, und dass es einen Bedarf an flexibleren Fragetypen, Testabläufen und Bewertungen gibt (vgl. [4]).

Um dem über klassische Fragetypen (Multiple-Choice, Lückentext, etc.) hinausgehenden Bedarf gerecht zu werden, besteht ein erster Ansatz darin, diese Fragetypen kombinieren zu können. Darüber hinaus muss eine Lösung den Anforderungen einer technischen Hochschule genügen und die Eingabe und automatische Anwendung von Formeln sowie die Integration in ein vorhandenes LMS ermöglichen.

3 Situation an der RWTH Aachen University

Das webbasierte Lehr- und Lernportal L^2P der RWTH Aachen University ermöglicht allen Lehrenden und Studierenden, von jedem Ort und zu jeder Zeit auf lehrveranstaltungsspezifische Lernmaterialien und gesteuerte Arbeitsprozesse zuzugreifen. Wichtige Eigenschaften und Funktionen des Portals sind beispielsweise die einheitliche, übersichtliche Verbreitung von Informationen und Ankündigungen, die einfache Distribution von Lernmaterialien, die Verwendung von Kommunikations- und Kollaborationsanwendungen sowie eine zentral gesteuerte Administration und ein umfassendes Supportangebot.

Um die Einbindung bestehender administrativer Abläufe und IT-Strukturen zu gewährleisten, wird L^2P administrativ durch das CAMPUS-Informationssystem gesteuert [8]. Auf Basis des Microsoft Office SharePoint Server 2007© (im Folgenden kurz SharePoint, siehe [9]) wurde ein integriertes eLearning-Portal mit einer offenen, skalierbaren Architektur entwickelt. Durch die niedrige Einstiegshürde, intuitive eLearning-Grundfunktionen und über Webservices definierte Schnittstellen zu anderen IT-Systemen gewährleistet es die Unterstützung des regulären Lehrbetriebs der gesamten Hochschule [10]. Eine wichtige eLearning-Grundfunktion, die SharePoint nicht standardmäßig zur Verfügung stellt, ist die Verwaltung und Durchführung elektronischer Tests. Da ein wachsender Bedarf an flexibel gestaltbaren elektronischen Tests vorhanden ist und andererseits Lösungen angestrebt werden, die mit den vorhandenen Technologien realisierbar sind, wird im Folgenden ein entsprechender Ansatz vorgestellt, der direkt in die Lernplattform integriert ist. Dieser Ansatz zeichnet sich dadurch aus, dass Fragen nicht genau einem Fragetypen entsprechen müssen. Statt dessen ist es möglich in einer Frage mehrere Fragetypen zu vereinen und diese zusammenhängend zu bewerten (vgl. Kapitel 4.4, Testfragen-Szenario 2).

4 Integration von elektronischen Tests in L^2P

Die Realisierung elektronischer Tests für SharePoint kann durch unterschiedliche konzeptionelle Ansätze umgesetzt werden. Beispielsweise wäre es möglich, externe Systeme zur Erstellung von Tests über *WebServices* in einer *serviceorientierten Architektur*

(SOA) in SharePoint zu integrieren oder über Schnittstellen für austauschbare elektronische Lerninhalte, wie beispielweise das *Sharable Content Object Reference Model (SCORM)*[9] unter Verwendung des *SharePoint Learning Kit*[10], anzubinden. Um jedoch eine gänzliche Integration elektronischer Tests mit den bereits vorhandenen SharePoint-Funktionen von L^2P zu ermöglichen und dadurch den Anwendern die gewohnte Benutzungsstruktur zur Verfügung zu stellen, wurde die Entwicklung einer integrierten Lösung angestrebt, die vollständig auf den in SharePoint verwendeten Technologien aufbaut und, soweit möglich, vorhandene Konstrukte und Techniken nutzt.[11] Die flexible Gestaltung von Fragetypen und deren Kombinationen ist eine weitere Anforderung an das zu entwickelnde System. Diese Flexibilität darf nicht zu Lasten der Benutzbarkeit führen und nicht voraussetzen, dass die Anwender eine Programmiersprache zur Erstellung von Fragen erlernen müssen.

Die Durchführung des Projektes ist in zwei grobe Entwicklungsphasen gegliedert. In der ersten Phase wird zurzeit eine erweiterbare und im Rahmen der universitären Lehre universell einsetzbare Rahmenarchitektur konzeptioniert. In einer zweiten Phase sollen zukünftig flexibel erweiterbare *Wizards* entwickelt werden, mit deren Hilfe die Testersteller unterschiedliche Fragetypen und elektronische Tests auf einfache Art und Weise erzeugen und verwalten können. Eine temporäre Vereinfachung wird in Bezug auf die darstellbaren Typen von Fragen getroffen, indem zunächst eine Beschränkung auf Fragen vorgenommen wird, die durch Kombinationen standardisierter HTML-Formularelemente wie Eingabefelder, Radio-Buttons und Checkboxen erzeugt werden können. Auf diese Weise lassen sich bereits viele Fragetypen wie z. B. Lückentext, Kurzantwort oder Single-/Multiple-Choice realisieren. Komplexere Frage, die etwa durch Drag & Drop oder die Auswahl von Bildbereichen beantwortet werden, sind dadurch zunächst ausgeschlossen. Die verwendete Architektur der dargestellten Lösung lässt sich jedoch in zukünftigen Entwicklungsphasen in diese Richtung erweitern (vgl. Abschnitt 5).

Zur Verwaltung von Tests, Fragen und Punkten sowie zur Speicherung der Antworten werden die bereits in SharePoint vorhandenen Listen als Datenstruktur verwendet. Die einzelnen Listenelemente werden intern in einer SQL-Datenbank gespeichert. Als geeignet hat sich das in SharePoint bereits existierende Konstrukt einer Umfrage als spezielle Form einer Liste erwiesen. Elektronische Tests können demnach als Variante einer Umfrage aufgefasst werden. Im Rahmen der Anpassung kommt jedoch eine sogenannte *Auswertungsschicht* hinzu, die es ermöglicht die Lösungen zu bewerten. Auf Basis der gegebenen Antworten wird eine entsprechende Punktzahl berechnet und eine Rückmeldung generiert.

4.1 Konzeption von Testfragen

Ein *Test* setzt sich aus beliebig vielen *Testfragen* zusammen. Technisch betrachtet ist ein Test eine Folge von beliebig vielen HTML-Dokumenten, da die einzelnen Testfragen

[9] `www.adlnet.gov` (07.11.2008)
[10] `www.codeplex.com/SLK` (07.11.2008)
[11] Tatsächlich genügen für den hier vorgestellten Ansatz bereits die *Windows SharePoint Services 3.0*, welche die Basis für den *Office SharePoint Server 2007* bilden.

durch jeweils ein HTML-Dokument repräsentiert werden, das ein oder mehrere Formularfelder enthält.[12] Um eine Testfrage zu erstellen, stehen dem Ersteller von Fragen unterschiedliche Möglichkeiten zur Verfügung. Beispielsweise kann ein in SharePoint eingebauter graphischer HTML-Editor genutzt, der HTML-Quellcode direkt eingegeben oder eine HTML-Datei hochgeladen werden. Durch die Verwendung von HTML ist es möglich, bei Bedarf Medien (z. B. Bilder, Videos) einzubinden und damit die Darstellung der Fragen flexibel zu gestalten. Dieser HTML-Code wird anschließend durch L^2P validiert. Ein Parser analysiert den vorliegenden Code und bestimmt die enthaltenen Formularfelder. Diese werden dann intern auf geeignete SharePoint-Felder einer Umfrage abgebildet. Die ASP.NET-Seite *NewItem.aspx*, die in SharePoint standardmäßig zur Erstellung eines neuen Listenelements bei der Teilnahme an einer Umfrage aufgerufen wird, wird derart umgewandelt, dass anstatt der üblichen Formulareingabe von SharePoint der zuvor erwähnte HTML-Code mit den Formularfeldern angezeigt wird. Die Nutzereingaben werden auf die entsprechenden Felder der SharePoint-Umfrage abgebildet. Alle Eingaben eines Testteilnehmers erfolgen demnach auf der Basis des vom Testersteller hinterlegten HTML-Codes. Die Speicherung der Daten hingegen erfolgt in einer standardmäßigen SharePoint-Umfrage.

Der Testersteller kann sogenannte *Test-Regeln* definieren, um die Antworten eines Testteilnehmers auszuwerten, diese anhand von Punkten zu bewerten und eine Rückmeldung zu erzeugen. Eine *Test-Regel* setzt sich aus den drei Parametern *Bedingung*, *Punkte* und *Feedback* zusammen. Die Bedingungen können durch *Microsoft Excel-Formeln* spezifiziert werden, da SharePoint über das Konstrukt von berechneten Feldern als Feldtyp für Listen verfügt. Der Wert eines berechneten Felds wird auf Basis einer Formel aus den Werten anderer Felder der gleichen Liste berechnet, wobei die Formel im wesentlichen der Notation von Excel folgt. Eine Test-Regel wird in zwei berechnete Felder übersetzt: ein Punkte-Feld vom Typ *Zahl* zur Erfassung der Punkte für diese Regel und ein Feedback-Feld vom Typ *Text* zur Erfassung der Rückmeldung.

Ein Test-Szenario kann so aussehen: Es soll eine Testfrage vom Typ Kurzantwort erstellt werden, etwa *„Wie lautet die Abkürzung der Hochschule Aachen?“*. Die richtige Antwort auf diese Frage ist „RWTH“ und wird mit 10 Punkten bewertet. Die entsprechende Test-Regel sieht dann wie folgt aus: Bedingung: $[\mathrm{feld1}] = \mathrm{'RWTH'}$; Punkte: 10; Rückmeldung: *Richtig beantwortet*. Bei der Auswertung einer Antwort wird geprüft, ob das Eingabefeld mit der ID *feld1* dem Wert *RWTH* entspricht. Ist diese Bedingung erfüllt, wird das Punkte-Feld der Regel den Wert 10 erhalten und das Feedback-Feld der Regel den Wert „Richtig beantwortet“. Wird die Bedingung nicht erfüllt, entsprechen die Werte 0 bzw. leerem Text, da keine weitere Bedingung, keine Punkte und kein Feedback angegeben wurden. Der Teilnehmende erhält damit 0 Punkte und keine Rückmeldung.

Für jede Testfrage können beliebig viele Regeln definiert werden. Für die Gesamtpunktzahl einer Frage wird die Summe aller Punkte der einzelnen Regeln berechnet, während sich das Gesamtfeedback einer Frage aus der Verknüpfung aller einzelnen Rückmeldungen entsprechend der einzelnen Regeln ergibt. Nachdem ein Testteilnehmer einen Test durchgeführt hat, werden diesem, gemäß der vom Testersteller definier-

[12] Um den Code geeignet validieren und weiterverarbeiten zu können, wird tatsächlich *XHTML 1.0 Strict* als Format vorausgesetzt, siehe `www.w3.org/TR/xhtml1` (07.11.2008).

ten Testeinstellungen, die Gesamtpunktzahl und das Gesamtfeedback aller Fragen des Tests angezeigt (vgl. hierzu Abschnitt 4.2).

Um die Erstellung und Verwaltung von Fragen zu vereinfachen, wurde die Spezifikation von XHTML 1.0 erweitert, so dass Test-Regeln und gegebenenfalls Musterlösungen direkt in einem XHTML-Dokument mit entsprechendem Namespace durch spezielle Tags angegeben werden können. So lassen sich die Fragen inklusive aller Regeln und Musterlösungen komfortabel importieren und exportieren.

4.2 Testeinstellungen

Damit ein flexibles und didaktisch nutzbares System entsteht, genügt es nicht, Testfragen und Feedback zu generieren. Der Testersteller muss weitere Einstellungen für einen Test festlegen können, damit gezielt didaktische Szenarien umsetzbar sind und der Testteilnehmer weiß, wie er mit einem Test umgehen muss. Aus diesem Grund ist es für den Testersteller möglich, zusätzlich zu den einzelnen Testfragen für jeden Test bestimmte Einstellungen zu definieren. Er kann beispielsweise festlegen, wie oft ein Testteilnehmer einen Test durchführen darf. Ebenfalls kann er bestimmen, ob jeder Testversuch gespeichert wird oder ein neuer Aufruf auf dem Ergebnis der vorherigen Testdurchführung basiert. Desweiteren kann der Testersteller festlegen, ob der Test innerhalb eines bestimmten Zeitlimits durchgeführt werden soll und ob es dem Testteilnehmer möglich sein soll, frei durch die angebotenen Testfragen zu navigieren oder nur eine lineare Bearbeitung der Fragen in einer festgelegten Reihenfolge zulässig ist. Ein flexibles Feedbackmanagement wird dadurch realisiert, dass der Testersteller für jeden Test bestimmen kann, ob Feedback nach jeder Antwort, nach Beendigung des Tests aus allen Fragen gesammelt oder gar nicht angezeigt werden soll. Zusätzlich ist es möglich, Musterlösungen zu den Test-Fragen anzugeben, die dem Testteilnehmer bei der Auswertung angezeigt werden.

4.3 Vorteile des vorgestellten Ansatzes

Der hier vorgestellte Ansatz hat aus Sicht der Software-Entwicklung den Vorteil, dass er vollständig auf den vorhandenen Strukturen von SharePoint aufbaut (Umfragen, berechnete Felder etc.). Insbesondere kann für die Auswertung der Antworten auf Excel-Formeln zurückgegriffen werden, so dass kein eigener Mechanismus zur Korrektur entwickelt werden muss. Zudem ist der Excel-Formalismus mächtig genug, um die meisten Fragetypen zu modellieren. Von Vorteil ist ebenfalls, dass durch die Nutzung von Excel numerische Berechnungen und Kombinationen von Auswertungen möglich sind. Beispielsweise kann geprüft werden, ob angegebene Zahlen in einem bestimmten Wertebereich liegen. Durch die Kombination mit HTML erhalten die Ersteller von elektronischen Tests eine nahezu unbeschränkte Freiheit in der Gestaltung und Auswertung von Fragen. So sind zahlreiche Varianten von Testfragetypen und deren Kombinationen realisierbar. Um die Wiederverwendbarkeit von Fragen für beliebig viele Tests zu ermöglichen, werden alle bereits erstellten Fragen in einem *Fragenpool* gesammelt. Die genannten Möglichkeiten heben den vorgestellten Ansatz insbesondere durch die Kombination von Feldern von den meisten handelsüblichen Testsystemen ab.

4.4 Testfragen-Szenarien

Im Folgenden weden zwei Beispiele möglicher Test-Szenarien dargestellt.

Testfragen-Szenario 1: Freitext mit differenzierten Punkten Der Testteilnehmer soll eine Texteingabe als Antwort geben, wobei bei der Auswertung nicht zwischen Groß- und Kleinschreibung unterschieden werden soll. Der Ersteller möchte für verschiedene Antworten unterschiedliche Punkte und differenzierte Rückmeldungen definieren. Dieses Szenario kann wie folgt realisiert werden: Durch die Angabe von entsprechendem HTML-Code wird dem Testteilnehmer die Frage *„Wie heißt die griechische Göttin der Liebe?“* gestellt. Die Antwort wird in einem Textfeld mit der ID *feld1* angegeben. Als Regeln werden entsprechend definiert:

- Bedingung: $\mathrm{KLEIN}([\mathrm{feld1}]) = {'}\mathrm{aphrodite}{'}$; Punkte: 10; Rückmeldung: *Richtig beantwortet.*
- Bedingung: $\mathrm{KLEIN}([\mathrm{feld1}]) = {'}\mathrm{venus}{'}$; Punkte: 5; Rückmeldung: *Venus ist das römische Pendant zu Aphrodite, dafür gibt's nur die halbe Punktzahl.*

Testfragen-Szenario 2: Kombinationen aus mehreren Eingabefeldern Da die Formularfelder in den Regelbedingungen explizit genannt sind, ist es technisch möglich, auch Kombinationen von mehreren Feldern zusammenhängend zu bewerten. Beispielsweise kann man die Auswahl einer Single-Choice-Frage mit einem Textfeld kombinieren und nur Punkte vergeben, wenn beide Angaben korrekt sind. Eine solche Kombination aus Fragetypen kann beispielsweise in einem mathematischen Kontext sinnvoll sein. Im folgenden Beispiel wird dem Testteilnehmer das Bild eines Funktionsgraphen angezeigt. Die Fragestellung lautet *„Gegeben ist folgender Funktionsgraph. Bestimmen Sie die Funktionsklasse und den zugehörigen Parameter c.“* Der Testteilnehmer kann aus einer Liste von Radio-Buttons mit der ID *klasse* eine Option auswählen (z. B. $c \cdot x^2$, $c \cdot x^3$ etc.) und in dem Textfeld mit der ID *param* eine Eingabe zur Bestimmung des Parameters c machen. Als Regeln werden definiert:

- Bedingung: $\mathrm{UND}([\mathrm{klasse}] = {'}\mathrm{opt1}{'}; \mathrm{WERT}([\mathrm{param}]) = 4)$; Punkte: 10; Rückmeldung: *Richtig beantwortet.*
- Bedingung: $\mathrm{UND}([\mathrm{klasse}] = {'}\mathrm{opt1}{'}; \mathrm{WERT}([\mathrm{param}]) > 4)$; Punkte: 5; Rückmeldung: *Richtige Klasse, aber der Parameter ist zu groß.*
- Bedingung: $\mathrm{UND}([\mathrm{klasse}] = {'}\mathrm{opt2}{'}; \mathrm{WERT}([\mathrm{param}]) = 4)$; Punkte: 0; Rückmeldung: *Der Parameter ist zwar richtig, aber die Klasse ist falsch.*

5 Ausblick

Zurzeit befindet sich das dargestellte Projekt in der Startphase. Der bereits implementierte Prototyp ermöglicht eine flexible Gestaltung von Fragetypen und deren Kombination. Dennoch müssen Testersteller Kenntnisse in HTML und Excel aufweisen. Dies kann den Benutzerkreis einschränken und zu einem höheren Aufwand bei der Erstellung von Fragen führen. In der nächsten Projektphase ist daher das Augenmerk auf die Vereinfachung der Erstellung von Fragen und deren Wieder- und Weiterverwendung zu

wenden. Hierfür müssen *Wizards* entwickelt werden, die es den Benutzenden ermöglichen, schnell und ohne zusätzliche Kenntnisse Fragen standardisierter Fragetypen zu entwerfen. Ebenfalls notwendig ist der Export und Import von Fragen mit Hilfe des QTI-Standards[13], um den Austausch mit anderen Systemen zu gewährleisten.

Neben den genannten Aspekten müssen zusätzlich weitere Standard-Fragetypen, wie beispielsweise *Zuordnungsaufgaben* mittels *Drag & Drop* und *Hot-Spot-Aufgaben*, entwickelt werden. Da es Fragetypen gibt (z. B. *Freitext-Aufgaben*), die nicht automatisch korrigierbar sind, dennoch aber häufig eingesetzt werden, ist es wünschenswert, den vorgestellten Ansatz um manuelle Bewertungsmöglichkeiten zu erweitern. Denkbar wäre hier beispielsweise die Integration in eine bereits bestehende Komponente des L^2P, die workflow-basiert eine manuelle Korrektur von Aufgaben ermöglicht [11].

Literatur

1. McAlpine, M.: Principles of Assessment, University of Luton: CAA Centre (2002)
2. Bull, J., McKenna, C.: Blueprint for computer-assisted assessment. RoutledgeFalmer, London , New York (2004)
3. Rütter, T.: Formen der Testaufgabe: Eine Einführung für didaktische Zwecke. Beck'sche Elementarbücher. Beck, München (1973)
4. Conole, G., Warburton, B.: A review of computer-assisted assessment. Research in Learning Technology **13**(1) (2005) 17–31
5. Altenberd-Giani, E., Schroeder, U., Stalljohann, P.: eAIXESSOR - a modular framework for automatic assessment of weekly assignments in higher education. In Uskov, V., ed.: Proceedings of the Seventh IASTED International Conference on Web-Based Education, ACTA Press (2008) 70–75
6. Krinke, J., Störzer, M., Zeller, A.: Web-basierte Programmierpraktika mit Praktomat. Softwaretechnik-Trends **22**(3) (2002) 51–53
7. Krempler, A., Neuper, W.: Formative assessment for user guidance in single stepping systems. In Auer, M. E., ed.: Proceedings of the conference ICL 2008. (2008)
8. Gebhardt, M., Bischof, C.: CAMPUS – das integrierte Informationssystem der RWTH Aachen. Praxis der Informationsverarbeitung und Kommunikation **27**(2) (2004) 110–115
9. English, B.: Microsoft Office SharePoint Server 2007 administrator's companion. Microsoft Press, Redmond Wash. (2007)
10. Gebhardt, M., Rohde, P., Schroeder, U.: Effizientes und nachhaltiges eLearning an der RWTH Aachen durch das integrierte Lehr- und Lernportal L^2P und das CAMPUS-Informationssytem. In Gaedke, M., ed.: Integriertes Informationsmanagement an Hochschulen: Quo vadis Universität 2.0? ; Tagungsband zum Workshop IIM 2007, Karlsruhe 01. 03. 2007, Karlsruhe, Univ.-Verl. (2007) 129–144
11. Dyckhoff, A.L., Rohde, P., Schroeder, U., Stalljohann, P.: Integriertes Übungsbetriebmodul im Rahmen eines hochschulweiten eLearning-Portals. In Seehusen, U., Lucke, U., Fischer, S., eds.: DeLFI 2008: Die 6. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V., 07. - 10. September 2008 in Lübeck, Germany. Volume 132 of LNI., Bonn, GI (2008) 185–196

[13] `www.imsproject.org/question` (07.11.2008)