

Kooperationstools der Notebook Universität Karlsruhe (TH)

Matthias Bonn¹, Sventje Dieter¹, Hartmut Schmeck¹

¹ Institut AIFB, Universität Karlsruhe (TH)
Englerstr. 11, 76131 Karlsruhe

Abstract. Dieser Beitrag beschreibt an der Universität Karlsruhe entwickelte Werkzeuge, die es ermöglichen, softwaregestützt in Lehrveranstaltungen zu interagieren und zu kooperieren. Unterstützt werden Szenarien zum elektronischen Feedback, Handmeldung und Multiple-Choice-Umfragen mit sofortiger Auswertung. Realisiert wurde das Interaktionssystem mit Hilfe der .NET Technologie unter dem Einsatz von Webservices.

1 Einführung

Lehre an der Universität hat heutzutage viele Gesichter: Da sind zum einen überfüllte Hörsäle mit teilweise 500 und mehr Studierenden, die versuchen, dem mitunter hohen Unterrichtstempo in den Grundvorlesungen zu folgen. Die Kommunikation zwischen Dozenten² und Hörern kommt dabei oft zu kurz: Es ist nicht oder nur eingeschränkt möglich, auf den Vorlesungsablauf Einfluss zu nehmen oder bei unverständlichem Sachverhalten Rückfragen zu stellen. Konzentrationsverlust oder sogar das typische „Abschalten“, bei dem man gar nicht mehr versucht, dem Stoff inhaltlich zu folgen, sind die Folge.

Zum anderen gibt es moderne Lehrveranstaltungen (Vorlesungen, Übungen, Seminare), die gleichzeitig an verschiedenen Orten stattfinden. Hierbei wird der Vortrag des Dozenten live an andere Unterrichtsräume übertragen. Sowohl Präsentationen als auch Audio- und Videodaten des Vortragenden können übermittelt werden. Dabei werden verschiedene kommerzielle und nichtkommerzielle Systeme eingesetzt (als Beispiel seien hier *E-Kreide* [1] und *mlb* [3] sowie die beiden Microsoft-Produkte *ConferenceXP* [2] und *Netmeeting* [6] genannt). Auch hier kann die Kommunikation zwischen Dozenten und Hörern zu kurz kommen, wenn auch aus anderen Gründen: Die (teilweise) räumliche Trennung erschwert den direkten Austausch.

Im Zuge der weiter fortschreitenden Ausstattung der Studierenden mit Notebooks oder so genannten Personal Digital Assistants (PDAs) und der immer besser werdenden Ausstattung mit Funknetz (WLAN)-Zugängen entstanden eini-

² Der besseren Lesbarkeit halber verwenden wir bei Bezeichnungen wie Dozent/in, Student/in u.ä. im Folgenden die männliche Form, was selbstverständlich keine geschlechtsspezifische Festlegung darstellt.

ge Systeme, die die Kooperation von Vortragenden und Hörern zum Vortragszeitpunkt unterstützen bzw. über räumliche Distanz erst möglich machen [2, 3, 8]. Auf dem Hintergrund der Erfahrungen mit bereits existierenden Lösungen wurde an der Universität Karlsruhe ein neues System mit einem technisch anderen Ansatz entwickelt.

In diesem Beitrag werden zunächst Anforderungen an derartige Kooperationswerkzeuge formuliert, darauf folgt eine Darstellung realisierter Anwendungsszenarien. Die technische Realisierung wird in Abschnitt 4 erläutert. Erste Praxiserfahrungen und eine Zusammenfassung beschließen den Beitrag.

2 Anforderungen

Das Ziel solcher Tools ist es, während einer Lehrveranstaltung auf elektronische Art und Weise zu kommunizieren und zu kooperieren, wenn dies entweder aufgrund der Menge der Anwesenden oder aufgrund einer räumlichen Distanz zwischen Hörern und Vortragendem auf herkömmliche, „klassische“ Art und Weise nicht möglich oder nicht sinnvoll ist.

Der Dozent sollte zum Beispiel Umfragen in Form von Multiple-Choice-Tests starten können, um sich ein Bild über den Wissensstand der Teilnehmer zu bilden. So könnte er auf diese Weise sehen, ob die Zuhörer ein Themengebiet verstanden haben oder durch gezielte Fragen typische Missverständnisse aufdecken. Ebenso vorstellbar wäre es, eine Art elektronische Wortmeldung anzubieten, mit der man die Aufmerksamkeit des Dozenten auf sich ziehen kann, wenn dies durch Zuruf oder die üblichen Handzeichen nicht machbar ist (siehe Abb. 1). Zurückhaltenden und räumlich abwesenden Teilnehmern bietet sich so die Möglichkeit, anonym Fragen zu stellen. Denkbar wäre auch eine Art Feedback, mit der man mitteilen kann, ob z.B. das Vorlesungstempo zu hoch ist. Idealerweise funktioniert das ganze auch über Hörsaal- oder Universitätsgrenzen hinweg, um beispielsweise bei Televeranstaltungen eingesetzt zu werden.

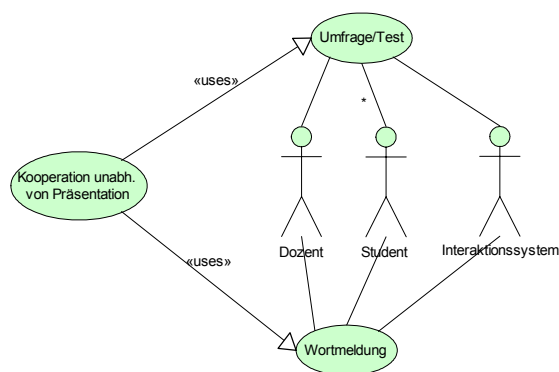


Abb. 1 Anwendungsfall Kooperation und abgeleitete Anwendungsfälle

Das Interaktionssystem soll unabhängig von der verwendeten Präsentationstechnik (Microsoft *Powerpoint*, Adobe *Acrobat* etc.) sein, also quasi als Zusatz zu den eigentlichen Lehrmaterialien „nebenbei“ verwendbar sein. Sowohl die Verteilung und die Installation als auch die Bedienung des Systems dürfen den Nutzer vor keinerlei Schwierigkeiten stellen. Trotz Prototypenstatus muss das Interaktionssystem ein gewisses Mindestmaß an Fehlertoleranz gegenüber Fehlbedienungen und Netzwerkproblemen aufweisen.

Um die Funktionalität problemlos auch über Netzwerkgrenzen hinweg zu garantieren, sollten einfache internetbasierte Kommunikationstechniken genutzt werden. Damit würde das oft aufwändige Einrichten eines VPN-Tunnels, mit dem man funktional getrennte lokale Netzwerke verbindet, entfallen.

Ein solches System wurde im Rahmen des BMBF-Projektes „Notebook-Universität“ [5] in Karlsruhe entwickelt.

3 Realisierte Anwendungsszenarien

Das System bietet die Möglichkeit zur elektronischen Kooperation während Lehrveranstaltungen wie Vorlesungen, Übungen, Teleseminaren etc. Unterstützt werden zur Zeit die folgenden drei Anwendungsszenarien:

1. Multiple-Choice-Umfragen. Der Vortragende stellt eine Multiple-Choice Frage. Diese kann direkt während der Veranstaltung erstellt oder aus einem vorher festlegbaren Katalog ausgewählt werden. Die Hörer rufen die Frage mit ihrem mobilen Gerät ab und beantworten sie durch Auswahl einer oder mehrerer der Antwortmöglichkeiten. Die Antworten werden gesammelt, und auf dem Dozentenrechner erscheint ein Histogramm über die Verteilung der bis dato eingegangenen Antworten. Jeder Teilnehmer darf nur einmal pro Frage antworten, um Ergebnisverfälschungen zu vermeiden.

2. Handmeldung beim Vortragenden. Will ein Hörer eine Frage stellen, so kann diese per Mausklick an den Dozenten übermittelt werden. Der Dozent erhält ein optisches Signal über die Meldung und entscheidet, ob er darauf eingehen möchte oder nicht. Die Frage wird in diesem Fall für einen späteren Abruf zurückgestellt. Um eine Überflutung zu verhindern können nicht mehr als fünf Fragen gleichzeitig in der Warteschlange stehen, weitere Anfragen werden abgewiesen. Diese Funktion ist vor allem sinnvoll bei Televeranstaltungen, bei denen der Vortrag des Dozenten in entfernte Hörsäle übertragen wird (der Dozent also durch Audio/Videoubertragung zu hören und evtl. zu sehen ist), die Hörer jedoch keine Möglichkeit haben, sich bemerkbar zu machen.

3. Voting. Dieses Szenario erlaubt es den Teilnehmern, eine Bewertung auszuwählen und abzugeben, zum Beispiel um auszudrücken, dass das Tempo der Veranstaltung zu hoch, angemessen oder zu niedrig ist. Diese Stimmen werden ebenfalls auf dem Dozentenrechner in einem Histogramm ausgewertet. So erhält der Vortragende ein Feedback darüber, das Unterrichtstempo ein wenig zu drosseln oder zu erhöhen. Damit sich die nach einer Anpassung des Tempos bessere

Bewertung auch positiv im Histogramm niederschlagen kann, werden periodisch die jeweils ältesten Stimmen gelöscht. Jeder Teilnehmer darf nur einmal pro Minute eine Stimme abgeben, um übermäßige Stimmabgabe zu vermeiden.

Die entwickelten Tools zur Unterstützung dieser Szenarien unterteilen sich in drei so genannte „Standalone“-Programme: Eines für die Studentenrechner und zwei für das Dozentennotebook. Diese Aufteilung wurde gewählt, um eine Überfrachtung der einzelnen Programme zu vermeiden und um zum Beispiel auf eines der obigen Szenarien verzichten zu können, falls eine bestimmte Funktionalität nicht gebraucht wird. Jedes Programm besitzt zusätzlich noch einen Konfigurationsdialog, der es erlaubt, die wichtigsten Parameter zur Laufzeit zu ändern. Die Startkonfigurationen sind jeweils in xml-Dokumenten abgelegt. Ebenfalls in xml-Form festgelegt ist der Fragenkatalog, der beim Start des Umfrage-Programms eingelesen wird. So können vor einer Veranstaltung bereits zum Stoff passende Fragen definiert werden. Sollten während einer Veranstaltung neue Multiple-Choice-Fragen erstellt werden, können diese auf Knopfdruck in den Katalog gespeichert werden, so dass sie beim nächsten Mal wieder zu Verfügung stehen. Die Screenshots in Abb. 2 zeigen die Tools in ihrem aktuellen Entwicklungsstand.

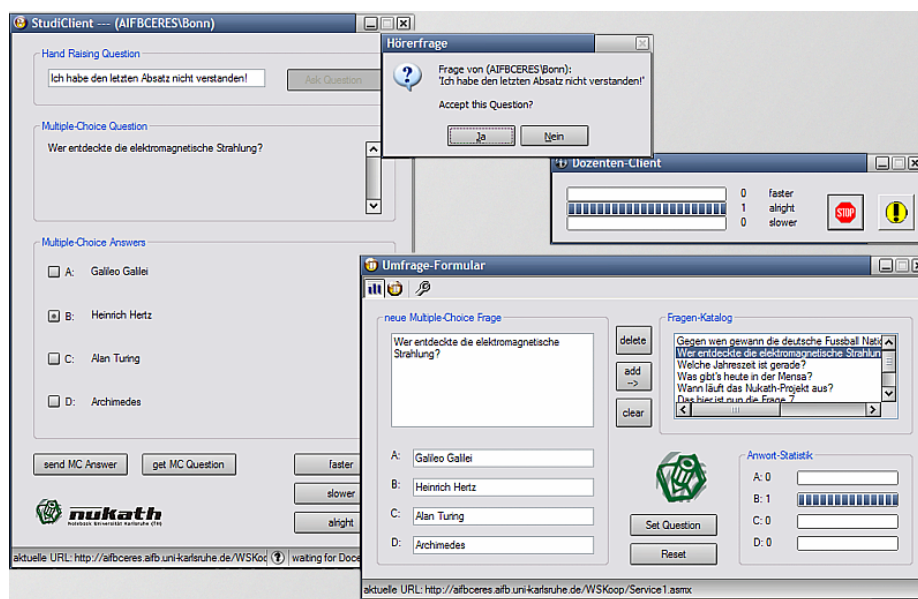


Abb. 2 Dozenten und Studenten Clients, Standalone-Versionen

Das Dozenten-Voting-Tool (in der Abbildung oben rechts zu sehen) kann in einen Präsentationsmodus geschaltet werden. Es schwebt dann ohne Titelleiste Platz sparend (und leicht transparent) im Vordergrund, so dass es während einer Präsentation im Vollbildmodus immer im Blick ist. Das Dozenten-Umfrage-Formular besitzt ebenfalls einen kleinen frei schwebenden Button (auf den Bil-

dern nicht gezeigt), der das Programm in den Desktop-Vordergrund holen kann. Die Tools sind also bei einer Vollbild-Präsentation immer schnell greifbar, ohne dass man die Präsentationssoftware verlassen muss.

Das Dozenten-Umfrage-Tool ist auch in einer Tablet-PC-Version (auf den Screenshots ebenfalls nicht gezeigt) verfügbar. Sie nutzt bei den Eingabefeldern die so genannte „*Microsoft Ink.Edit*“-Funktion. Sie übersetzt handschriftliche Eingaben in ASCII-Text. Kommt also ein Tablet-PC zum Einsatz, kann zum Erstellen von Fragen auf die Tastatur verzichtet werden. Zusätzlich sind bei dieser Version sämtliche Eingabefelder etwas vergrößert, um die Stifteingabe zu erleichtern.

Das System kann mehrere Veranstaltungen gleichzeitig verwalten, ohne dass diese sich gegenseitig beeinflussen. Jede Veranstaltung kann mit einem Passwort versehen werden, ohne dass es nicht möglich ist, Umfragen zu aktivieren oder auf Handmeldungen zu reagieren. Damit soll Sabotage von vornherein verhindert werden.

4 Technik

4.1 Architektur

Das Kooperationssystem wurde mit einer Client/Server-Architektur realisiert, unterscheidet sich also von den Ansätzen vorhandener Tools [2, 3], die IP-Multicast-Techniken nutzen, um während Lehrveranstaltungen Feedback und Umfrageszenarien zu ermöglichen. Um auf die Entwicklung eines eigenen Protokolls verzichten zu können, wurde das Datenaustauschverfahren SOAP [4] als Middleware-Technologie gewählt. Das Protokoll überträgt Methodennamen, Parameter und Rückgabewerte für entfernte Prozeduraufrufe (remote procedure call, RPC) im xml-Format. Der Server wird bei dieser Methode nicht über eine reine IP-Adresse, sondern eine URL der Form `http://servername/WSDL-Datei` angesprochen (Abb. 3). Die per URL angesprochene WSDL-Datei (Web Service Description Language [7]) beschreibt die Schnittstelle zum Server: Die vorhandenen Methoden-Namen, die erwarteten Parameter beim Aufrufen und die jeweiligen Rückgabetyper werden ebenfalls in xml-Form definiert. Das Verwenden dieser auch als Webservices bekannten Technik hat mehrere Vorteile:

- Wie bereits oben erwähnt, kann auf die Entwicklung eines eigenen Protokolls verzichtet werden, da SOAP vom WorldWideWeb-Consortium (W3C) standardisiert wurde. Es müssen lediglich Methodennamen, Aufrufparameter und Rückgabetyper spezifiziert werden.
- Es kann auf komplizierte und damit fehleranfällige TCP/IP-Socket-Programmierung verzichtet werden, da die Kommunikation mittels entfernter Prozeduraufrufe abläuft: Ein Client ruft eine Funktion auf dem

Server in genau der gleichen Weise auf, wie er eine eigene interne Funktion aufrufen würde.

- SOAP wird ins Web-Protokoll HTTP gekapselt und ist daher – im Gegensatz zu IP-Multicast – über Netzgrenzen hinweg einsetzbar, da bei HTTP jegliche Kommunikation vom Client initiiert wird, das Protokoll internetweit geroutet wird an Firewalls praktisch nie blockiert wird.

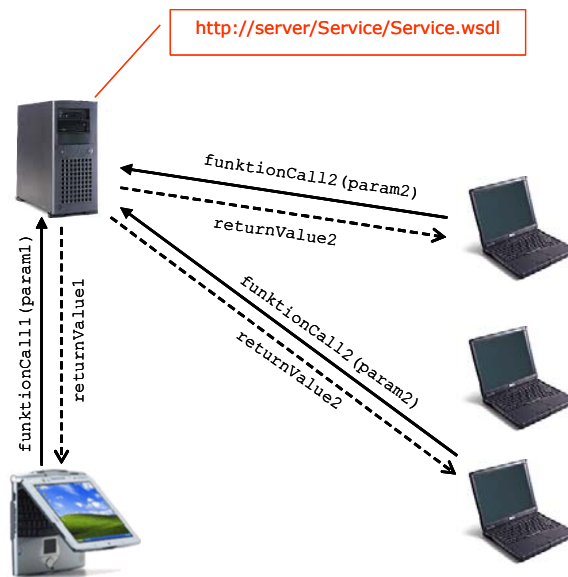


Abb. 3 Client/Server Architektur mit SOAP

Die Kommunikation läuft komplett RPC-basiert ab, das heißt, die Clients fragen beim Server permanent nach, ob es neue Daten gibt. Das ist notwendig, da es aufgrund des verwendeten HTTP-Protokolls nicht möglich ist, Daten ohne vorherige Anfrage vom Server zu den Clients zu übertragen. Z.B. ruft der Studenten-Client in regelmäßigen Abständen einen Wert ab, der anzeigt, ob eine neue Frage auf dem Server aktiv ist. Diese Zeitintervalle liegen in der Größenordnung von drei Sekunden, können aber bei schlechter Netzanbindung vom Nutzer auch beliebig verlängert werden. Da keine Echtzeitanforderungen vorliegen, stellt dieses Verfahren keinen großen Nachteil dar.

Das folgende UML-Diagramm (Abb. 4) zeigt einen groben Überblick über die Kommunikationsvorgänge beim Multiple-Choice-Szenario.

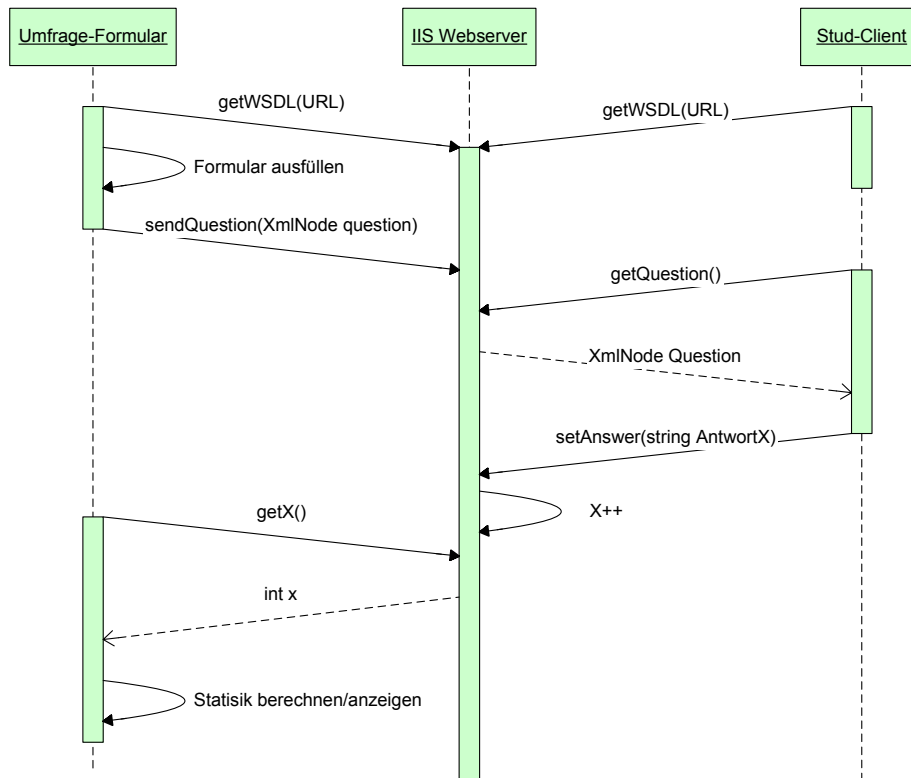


Abb. 4 UML-Sequenzdiagramm für das Multiple-Choice Szenario

Beim Übertragen der Informationen wird (als Funktionsparameter) jedes Mal der Name der gewünschten Vorlesung und ggf. ein Passwort mitgeschickt (im Diagramm aus Platzgründen weggelassen), damit der Server immer weiß, auf welche der eingerichteten Vorlesungen sich ein Dienstauftrag bezieht. Komplexere Datenstrukturen (die Umfragen) werden mit einem eigenen xml-Schema beschrieben, die wiederum mit SOAP – also quasi als xml-in-xml – übertragen werden. Die per XMLNode übertragene Frage besitzt folgende Struktur:

```

<element>
  <question>Die Frage</question>
  <answers>
    <answer id="1">Antwort A</answer>
    <answer id="2">Antwort B</answer>
    <answer id="3">Antwort C</answer>
    ...
  </answers>
</element>
  
```

4.2 Implementierung

Das System ist auf der Microsoft .NET-Plattform [9] in der Sprache C# realisiert worden, serverseitig kommt also der Internet Information Server IIS zum Einsatz, der über SOAP mit den Clients kommuniziert und von einem weiteren Teilprojekt der Notebook Universität gehostet wird. Auf Clientseite muss lediglich das .NET-Framework installiert sein. Eine Umsetzung als Web-Anwendung des Studenten-Clients für Browser ist im Rahmen des Projekts ebenfalls realisiert und für die Darstellung auf den kleinen Displays von PDAs (PocketPC und Palm-OS) optimiert worden. Diese Webschnittstelle wurde mit der ASP.NET Technik ebenfalls in C# realisiert.

Die Serveranwendung kann beliebig viele Vorlesungskontexte erzeugen und verwalten, die jeweiligen Verwaltungsdaten (Name und Passwort einer Vorlesung) werden persistent als xml-Datei auf dem Server gespeichert.

Eine Benutzerverwaltung wurde in das System noch nicht integriert, eine Anbindung an den zentralen Verzeichnisdienst der Universität ist jedoch geplant.

5 Praxiserfahrungen und Zusammenfassung

Das NUKATH-Kooperationssystem wurde im Sommersemester 2003 zur Unterstützung eines Teleseminars am *Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)* der Universität Karlsruhe in Kooperation mit der Universität Mannheim eingesetzt. Zur Übertragung der Lehrinhalte (*Powerpoint*-Folien und Webseiten) wurde die Application Sharing Funktion von *Netmeeting* [6] verwendet, für Video und Audio kam ein *Tandberg 6000* Konferenzsystem [10] zum Einsatz. Durch die Beteiligung der Universität Mannheim konnte die Zuverlässigkeit des Systems beim Einsatz über nicht-lokale Netze getestet werden.

Eine Evaluation, die am Ende der Lehrveranstaltung unter den Teilnehmern durchgeführt wurde, ergab ein insgesamt recht positives Bild: Die Kooperations-tools wirkten sich motivierend auf die Studenten aus, die meisten fanden die Veranstaltung durch den Einsatz des Systems interessanter. Die Bedienung bereitete in wesentlichen keine Probleme und das Testen der Software machte den Studenten Spaß. Letztendlich wurden die einzelnen Funktionen als nützlich für die Veranstaltung bewertet, wobei die meisten den Einsatz vor allem in größeren Veranstaltungen als sinnvoller erachten. Abb. 5 zeigt die Ergebnisse der Umfrage. Die Teilnehmer mussten jeweils eine Aussage mit Werten von -2 bis +2 bewerten, wobei -2 „Aussage trifft in keinerlei Weise zu“ und +2 „Aussage trifft voll und ganz zu“ bedeutete. Im Diagramm sind die jeweiligen Durchschnittswerte aufgetragen.

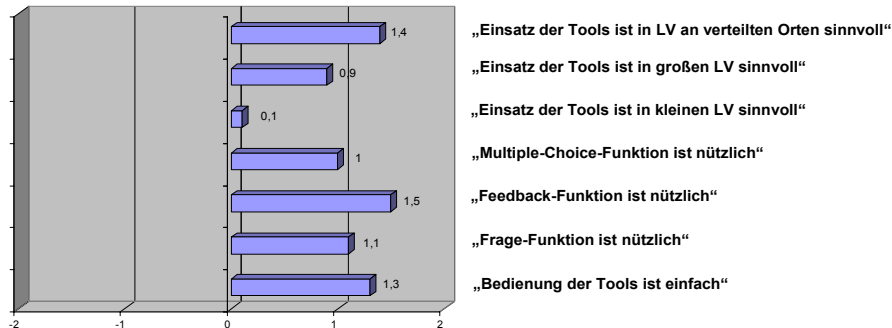


Abb. 5 Evaluationsergebnisse

Insgesamt hat die Webservice-Technik zuverlässig funktioniert, im Zuge der Veranstaltungen sind die Tools sowie die Funktionen des Servers laufend modifiziert und verbessert worden. Sie sollen im Wintersemester 2003/2004 in einer größeren Vorlesung mit mehreren hundert Teilnehmern getestet werden. Neben der Evaluation, wie sich das Serversystem und das Funknetz im Hörsaal unter erhöhter Belastung verhalten, sollen dabei auch der pädagogische Nutzen und der Mehrwert eines solchen Systems in Lehrveranstaltungen genauer beleuchtet werden. Auf technischer Seite sind die Integration der Weboberfläche in ein Portal der Universität Karlsruhe, der Ausbau von Management-Funktionen sowie die Realisierung einer Java-Version des Studenten-Clients geplant.

Referenzen

1. E-Kreide. *Elektronische Kreide: Eine Java-Multimedia-Tafel für den Präsenz- und Fernunterricht*, R. Rojas, L. Knipping, Wolf-Ulrich Raffel, Gerald Friedland. <http://www.ekreide.de/>
2. Microsoft Conference XP. <http://www.conferencexp.net>
3. Multimedia Lecture Board mlb, Universität Mannheim. Jürgen Vogel, Algorithms and Protocols for the mlb - a Shared Whiteboard. Presentation, Invited Presentation at PARC (Palo Alto Research Center), Palo Alto, CA, USA, May 2003. <http://www.informatik.uni-mannheim.de/informatik/pi4/projects/mlb/>
4. SOAP: Simple Object Access Protocol. <http://www.w3c.org/2000/xp/Group/>
5. BMBF-Projekt Notebook Universität Karlsruhe. <http://www.nukath.uni-karlsruhe.de/>
6. Microsoft Netmeeting. <http://www.microsoft.com/windows/netmeeting/default.asp>
7. Web Service Description Language (WSDL) 1.1. <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
8. WILD@Mannheim. Martin Mauve, Nicolai Scheele, Werner Geyer. *Enhancing Synchronous Distance Education with Pervasive Devices*. http://www.lecturelab.de/UCE_1.html
9. Das .NET Framework. <http://msdn.microsoft.com/netframework/technologyinfo/>
10. Tandberg Videokonferenzsystem 6000. <http://www.tandberg-videokonferenz.de/pdf/TANDBERG%206000.pdf>