

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Diplomarbeit

**Entwurf und Implementierung eines
Systems zur Durchführung von
Vorlesungsumfragen**

Boris Lohner

Eingereicht bei: Prof. Dr. Martin Hofmann

Betreuer: Dipl.-Math. Hans Dietmar Jäger
Dr. Martin Lange

Abgabetermin: 14. Dezember 2005

INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



Diplomarbeit

**Entwurf und Implementierung eines
Systems zur Durchführung von
Vorlesungsumfragen**

Boris Lohner

Eingereicht bei: Prof. Dr. Martin Hofmann

Betreuer: Dipl.-Math. Hans Dietmar Jäger
Dr. Martin Lange

Abgabetermin: 14. Dezember 2005

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. Dezember 2005

.....

(Unterschrift des Kandidaten)

Zusammenfassung

Im Rahmen dieser Diplomarbeit wird ein elektronisches System zur Durchführung von Vorlesungsumfragen erstellt.

Hierzu wird zunächst das in der Praxis bewährte papierbasierte Umfragesystem analysiert, nachdem einige Grundlagen zur Theorie von Umfragen aus dem Gebiet der empirischen Sozialforschung betrachtet wurden. Die Problemanalyse befasst sich sodann mit einem umfassenden Vergleich von elektronischen und papierbasierten Umfragetechniken und wird durch einen kurzen Blick auf einige weitere existierende Systeme abgerundet.

Zu Beginn der Entwurfsphase wägen wir Vor- und Nachteile verschiedener elektronischer Ansätze ab und entscheiden uns für ein Web-basiertes Umfragesystem. Die in der Problemanalyse erforschte Vielschichtigkeit einiger Aspekte wird in der Entwurfsphase durch die Konzeption eines modularen Frameworks gewürdigt. Das Zusammenspiel der Module wird durch ein ausge dehntes Beispiel vertiefend erklärt, bevor wir das Datenbankmodell beschreiben, das dem System zugrundeliegen soll.

Das vierte Kapitel beschreibt die Implementierung des Systems in Java und stellt die verschiedenen Systemkomponenten dar. Nach der Beschreibung des Framework-Kerns und der Web-Anwendung, die auf dem Framework basiert, erläutern wir kurz zwei verwendete Design-Pattern: das Visitor-Pattern und Data Access Objects. Eine Beschreibung aller implementierten Module beendet das Kapitel.

Im letzten Kapitel analysieren wir schließlich unsere Erfahrungen mit dem Prototyp des Systems.

Inhaltsverzeichnis

Inhaltsverzeichnis	i
Abbildungsverzeichnis	iv
1 Einleitung	1
2 Problemanalyse	3
2.1 Erkenntnisse aus der empirischen Sozialforschung	3
2.1.1 (Keine) Stichproben	4
2.1.2 Fragebogenumfang	4
2.1.3 Fragetypen	5
2.1.4 Gütekriterien einer Messung	8
2.2 Beschreibung des existierenden Systems	9
2.3 Vergleich papierbasierter und elektronischer Systeme	10
2.3.1 Fragebogengestaltung	11
2.3.2 Fragebogendarstellung	11
2.3.3 Internationalisierung	13
2.3.4 Handschrifterkennung und Anonymität	14
2.3.5 Persönliche Daten und Anonymität	14
2.3.6 Vorzeitige Auswertung	14
2.3.7 Kontinuierliche Evaluation	16
2.3.8 Automatisierte Stimmabgabe	16
2.3.9 Unplausible Stimmabgaben	19
2.3.10 Stimmabgabe durch unberechtigte Personen	20
2.3.11 Mehrfache Stimmabgabe	22
2.3.12 Rücklaufquote	24
2.3.13 Organisatorische Aspekte	26
2.3.14 Rechtliche Fragen	26
2.4 Erkenntnisse aus dem Vergleich verschiedener Umfragen	27
2.4.1 Vergleichende Betrachtung verschiedener Umfragen	27
2.4.2 Zusammenführende Behandlung gleichartiger Fragen	30
2.4.3 Richtlinien für eine einheitliche Formulierungsstruktur	33

3	Entwurf	34
3.1	Denkbare elektronische Realisierungen	34
3.2	Modulares Framework	36
3.3	Datenbankschema	43
3.3.1	Fragenkataloge	43
3.3.2	Internationalisierung	43
3.3.3	Ratingskalen	43
3.3.4	Pseudofragen	46
4	Implementierung	47
4.1	Verwendete Technologien	47
4.1.1	Betrachtung möglicher Implementierungssprachen	47
4.1.2	JAAS als Framework-Basis?	48
4.1.3	Template-Engine für Web-basierte Lösung	48
4.2	Implementierung des Framework-Kerns	49
4.3	Transferstruktur	50
4.4	Implementierung des Webinterfaces	50
4.5	Verwendung des Visitor-Patterns	51
4.6	Verwendung von Data Access Objects (DAO)	54
4.7	Implementierte Module	55
4.7.1	DatenbankHandler	56
4.7.2	MinimalzeitHandler	58
4.7.3	ChainHandler	59
4.7.4	DelayHandler	60
4.7.5	FalseHandler	61
4.7.6	TrueHandler	62
4.7.7	WebPresentationHandler	63
4.7.8	WebAuswertungHandler	64
4.7.9	WebReportHandler	65
4.7.10	WebAntwortbogenHandler	66
4.7.11	PasswordHandler	67
4.7.12	BonbonHandler	68
4.7.13	PseudoFrageAusfuelldauer	69
4.7.14	PseudoFrageSourceIP	70
4.7.15	ZeitfensterHandler	71
4.7.16	MessageHandler	72
4.7.17	AuthHandler	73
5	Abschliessende Betrachtung	74
5.1	Erfahrungen aus den ersten Testläufen	74
5.1.1	Ergebnis der Systembeurteilung	74
5.1.2	Vorzug von Radiobuttons gegenüber Auswahllisten	75
5.1.3	Messung der Ausfülldauer durch eine PseudoFrage	75

5.2	Fazit	76
6	Danksagung	82
A	Management und Erweiterung des Systems	83
A.1	Systemmanagement durch GUI-Applikation	83
A.2	Schnittstellen zur Erweiterung	83
A.3	Verwendete Entwicklungswerkzeuge	84
	Literaturverzeichnis	85

Abbildungsverzeichnis

2.1	Verzerrung durch Intervallvorgabe	7
2.2	Beispiel-Captcha	17
2.3	Umfragebeispiel: Münchner Bäder	29
2.4	Umfragebeispiel: Perspektive Deutschland	30
2.5	Umfragebeispiel: Fakultät für Physik	31
2.6	Mapping-Problem unterschiedlich dimensionierter Skalen	33
3.1	Übersicht der fünf System-Aufgaben	36
3.2	Sequenzdiagramm eines möglichen Ablaufs	40
3.3	Framework-Entwurf mit Beispielhandlern	42
3.4	Entity-Relationship-Diagramm	44
3.5	Datenbankschema aus Sicht von DBmodeller	45
4.1	Interaktion von Servlet und Visitor	52
5.1	Geschätzte vs. gemessene Ausfülldauer	76
5.2	Fragen zum Umfragesystem	77
5.3	Ergebnis der Systembewertung	78
5.4	Fragebogen das SIGNAL-Projekts	79
5.5	Fragebogen das SIGNAL-Projekts, Seite 2	80
5.6	Fragebogen unseres elektronischen Systems	81

Kapitel 1

Einleitung

Der Bewertung der Lehre durch die Studenten wird an der LMU seit vielen Jahren ein hoher Stellenwert eingeräumt. Hierzu werden am Ende des Semesters Fragebögen verteilt und ausgewertet. Seit einigen Jahren wird eine weitere Umfrage mitten im Semester durchgeführt, um noch während der laufenden Vorlesung Feedback zu erhalten und eventuelle Probleme zwischen Studenten und Lehrkörper beheben zu können. Vor allem der Umstand, daß die Hörerzahl während des Semesters in der Regel sinkt, bewirkt ein etwas verfälschtes Bild der Abschlussumfrage, da an dieser nur noch Studierende teilnehmen, die die Vorlesung bis zum Schluss besuchten und daher keine allzu negative Bewertung abgeben, wohingegen besonders die Gründe der Aussteiger interessant wären um die laufende Vorlesung gegebenenfalls rechtzeitig verändern zu können, oder Konsequenzen für künftige Lehrveranstaltungen zu ziehen.

Dem Informationsgewinn dieser Umfragen steht jedoch der hohe Aufwand der Auswertung der papierbasierten Fragebögen entgegen. Ziel dieser Diplomarbeit ist es, ein elektronisches Vorlesungsbewertungssystem zu erstellen, dessen Auswertung automatisch geschieht und nicht nur den Aufwand des Auszählens von Abstimmungsbögen reduziert, sondern neue Wege ermöglicht, wie beispielsweise die in Abschnitt 2.3.7 dargestellte kontinuierliche Evaluation. Das Ergebnis soll dabei gegenüber dem heutigen papierbasierten System keine wesentlichen Nachteile aufweisen.

Im Rahmen dieser Diplomarbeit wollen wir ein elektronisches System zur Durchführung von Vorlesungsumfragen erstellen und mit der existierenden papierbasierten Umfrage vergleichen.

Wir beginnen mit einem kurzen Ausflug in die empirische Sozialforschung und informieren uns über einige Grundlagen zu verschiedenen Fragetypen. Als nächstes schwenken wir zur Betrachtung des existierenden Umfragesystems und beschreiben den Ablauf einer Vorlesungsumfrage in ihrer aktuellen Form. Der Schwerpunkt der Problemanalyse liegt in der vergleichenden Betrachtung von elektronischen Systemen mit papierbasierten Umfragetechniken. Hierbei streifen wir eine Vielzahl von Themenbereichen, wie z.B. die Regelung des Zugangs zum Umfragesystem, Mittel zur Steigerung der Teilnahmebereitschaft oder die Erkennung nicht ernst gemeinter Stimmabgaben. Eine Betrachtung existierender Systeme rundet die Problemanalyse ab und lie-

fert uns einige weitere Anregungen zum Einsatz des Systems in der Praxis.

In der Entwurfsphase entscheiden wir uns nach einer Abwägung von Für und Wider für eine Web-basierte Realisierung unseres zu erstellenden Umfragesystems. Durch die in der Problem-analyse erkannte Vielgestaltigkeit einiger Problemlösungen müssen wir das System als modulares Framework konzipieren, und verdeutlichen die Zweckmäßigkeit und Funktionsweise unseres Entwurfs. Die Beschreibung des Datenbankmodells und die Vorstellung des Konzepts unserer sogenannten *Pseudofragen* beschließen das Kapitel.

In der Implementierungsphase treffen wir als erstes die Wahl der zu verwendenden Programmiersprache und entscheiden uns für Java. Wir beschreiben die Kernstruktur des Frameworks und das zentrale Web-Interface samt seiner Hilfsklassen. Den Rest des Kapitels verwenden wir zur Beschreibung der im Rahmen dieser Arbeit bereitgestellten Module, durch die das Framework erst einsatzfähig wird.

In einem kurzen Schlusskapitel werten wir die Ergebnisse der durchgeführten Testläufe unseres System aus und treffen eine abschließende Bewertung.

Kapitel 2

Problemanalyse

Ziel dieser Diplomarbeit ist die Erstellung und Bewertung eines elektronischen Vorlesungsbewertungsframeworks, das dem existierenden papierbasierten System mindestens ebenbürtig ist.

Zunächst klären wir daher die wesentlichen Funktionen des existierenden Systems, deren Nachmodellierung notwendig ist um die geforderte Ebenbürtigkeit zu erreichen, bevor wir weitere vor- und nachteilige Eigenschaften papierbasierter und elektronischer Systeme vergleichen.

2.1 Erkenntnisse aus der empirischen Sozialforschung

Die elektronische Erfassung eines Meinungsbildes erfreut sich in verschiedensten Bereichen großer Beliebtheit. So werden mittlerweile diverse Fragebögen im Internet zum Ausfüllen angeboten (z.B. die von McKinsey gesponsorte, nach eigenen Angaben "weltweit größte gesellschaftspolitische Online-Umfrage" Perspektive Deutschland). Sogar wichtige politische Entscheidungen wie die Bestimmung des Präsidenten der USA werden teilweise durch elektronische Wahlmaschinen entschieden und Community-basierte Open-Source Projekte wie z.B. Debian Linux sind aufgrund weltweiter Zusammenarbeit bei ihrer Entscheidungsfindung über die Organisation und Ziele des Projektes geradezu auf Online-Abstimmungen angewiesen.

Zur Durchführung von Meinungs-Umfragen bedienen wir uns der reichhaltigen Erkenntnisse und Techniken der empirischen Sozialforschung, die zur Überprüfung von Hypothesen neben anderen Techniken, wie z.B. Beobachtungen von Verhalten und Inhalten, zu einem guten Teil durch die Durchführung von Interviews zu neuen Erkenntnissen gelangt. Nach [Diekmann 95, Seite 371] wird das "Interview gern als Königsweg der Sozialforschung bezeichnet".

Auch wenn unsere Umfrage nicht durch einen menschlichen Interviewer durchgeführt wird, sondern die befragte Person selbst den Fragebogen ausfüllt, können wir dennoch Erkenntnisse über die Fragebogenkonstruktion übernehmen und in das von uns zu erstellende System einfließen lassen.

2.1.1 (Keine) Stichproben

Bei der Durchführung von Umfragen steht die empirische Sozialforschung oft vor dem Problem, eine repräsentative Teilmenge auszuwählen, da eine allumfassende Umfrage zu aufwendig wäre. Die Befragung einer Teilmenge kann bei deutlich niedrigeren Kosten zum gleichen Ergebnis führen wie die Befragung aller Probanden, wenn die Teilmenge gemäß einiger Regeln gebildet wird.

Ursache für die hohen Kosten liegen zum einen in manchen Umfragetechniken, wie z.B. persönliche Interviews, die für jede zu erfassende Meinung ein gewisses Arbeitszeitkontingent erfordern. Doch auch das direkte Ausfüllen eines Fragebogens erfordert, daß der Fragebogen dem Probanden zur Verfügung steht; dieses z.B. durch postalische Zustellung zu erreichen ist wiederum mit Kosten für jeden Fragebogen verbunden.

Im hier zu lösenden Szenario einer Vorlesungsumfrage ist die Konstruktion einer möglichst repräsentativen Teilmenge aus mehreren Gründen nicht notwendig:

- Die Vorlesungshörer sind in der Vorlesung versammelt an einem Ort erreichbar, so daß Informationen über die Umfrage (und im Falle eines papierbasierten Systems auch die Fragebögen selbst) effizient an alle Hörer übermittelt werden können. Die durch eine Einschränkung der Umfrageteilnehmer auf eine Teilmenge erzielbare Kostenersparnis wäre in Form einiger eingesparter Blatt Papier und weniger Minuten bei der Auszählung äußerst gering und steht in keinem Verhältnis zum Aufwand, der zur Konstruktion einer derartigen Teilmenge erforderlich wäre.
- Die Anzahl der Hörer einer Vorlesung bewegt sich im Bereich von einem halben Dutzend (z.B. Vorlesung Compilerbau im WS 2001/02) bis einige hundert (z.B. Vorlesung Analysis1) und ist somit nicht sehr groß im Vergleich mit der Anzahl der Einwohner einer Großstadt, die z.B. über ihre Zufriedenheit mit der Kommunalpolitik befragt werden sollen. Die Beschränkung auf eine repräsentative Teilmenge ist daher nicht notwendig, da die Gesamtmenge der Hörer klein genug ist. Auch dürfte sich die Bildung einer *repräsentativen* Teilmenge aus einer derart kleinen Gesamtmenge äußerst schwierig – wenn nicht sogar unmöglich – gestalten.

Eine Beschränkung der Umfrage auf eine repräsentative Teilmenge ist daher weder notwendig um die Datenmenge auf ein handhabbares Maß zu reduzieren, noch ist sie möglich, da aufgrund der kleinen Probandenmenge eine Verfälschung des Umfrageergebnisses nicht ausgeschlossen werden kann.

2.1.2 Fragebogenumfang

Ein persönliches Interview wird in der Regel, wenn es begonnen hat, auch zu Ende geführt. Im Gegensatz dazu kann das direkte Ausfüllen eines Fragebogens ohne menschlichen Interviewpartner als Mittelsmann jederzeit abgebrochen werden, wenn der Umfrageteilnehmer die Lust

verliert.

Eine Untersuchung des Zusammenhangs zwischen Fragebogenlänge und Abbruchrate ergab nach [Welker 05], daß “weniger die Zahl der Einzelfragen als vielmehr die Gesamtlänge der Befragung relevant war. Ab zehn Minuten Länge steigt die Abbrecherquote deutlich an.” ([Welker 05, Seite 79])

Das bedeutet, daß sich nur durch eine Messung der zum Ausfüllen benötigten Zeit ermitteln lässt, ob der Fragebogen zu umfangreich geraten ist, und nicht durch ein Abzählen der Fragen. Zur Messung der Befragungsdauer im Rahmen dieser Arbeit verweisen wir auf die Abschnitte 4.7.13 und 5.1.3.

2.1.3 Fragetypen

Die empirische Sozialforschung unterscheidet die erfassten Daten durch verschiedene aufeinander aufbauende *Skalenniveaus*[Scholl 03a]. Dabei wird vorausgesetzt, daß zu Zwecken der Erfassung und Kommunikation, die Messdaten auf eine Zahlenskala abgebildet werden.

Nominalskala Die verschiedenen Ziffern der Mess-Skala dürfen nur auf (Un)gleichheit untersucht werden und stellen keine Rangfolge dar.

Beispiel: Geschlecht

Fragetyp: Multiple-Choice-Frage

Ordinalskala Zusätzlich darf eine Rangordnung durch einen Vergleich auf größer/kleiner in die Skala interpretiert werden.

Beispiel: Schulabschluss

Fragetyp: Ratingskala

Intervallskala Nicht nur die Reihenfolge, sondern auch die Abstände der Werte sind von Bedeutung.

Beispiel: Temperatur

Fragetyp: Ratingskala

Ratioskala Zusätzlich ist der Nullpunkt der Skala von Bedeutung und gestattet Aussagen der Form “ist dreimal so gross wie”.

Beispiel: Jahresgehalt

Fragetyp: Zahlfrage

Eine vertiefende Einführung in die Bedeutung verschiedener Skalenniveaus findet sich bei[Scholl 03a, Seite 214ff]. Wir betrachten im folgenden verschiedene Fragetypen und ihr Zusammenspiel mit den eben beschriebenen Skalenniveaus.

Offene oder freie Fragen

Bei freien Fragen wird keine Einschränkung der Antwortmenge vorgegeben. Da sie nicht auf eine Skala abbildbar sind entsprechen sie auch keinem der dargestellten Skalenniveaus. Freie Fragen werden bei unseren Umfragen vor allem zum Erfassen detaillierter Kritik, Wünschen und Verbesserungsvorschlägen eingesetzt. Eine automatische Auswertung von freien Fragen ist nicht möglich, so daß sie nur eingesetzt werden sollten, wenn der erzielte Informationsgewinn den hohen Auswertungsaufwand einer manuellen Durchsicht jeder einzelnen Antwort rechtfertigt.

Multiple-Choice-Fragen

Multiple-Choice-Fragen werden für Messungen auf *Nominalniveau* verwendet. Ein typisches Beispiel für diesen Fragetyp ist die Frage nach dem Geschlecht der Testperson, oder nach der Lieblingsfarbe.

Zahlfragen

Einige Fragen dienen der Erfassung eines Zahlwertes auf *Rationiveau*. Beispielhafte Vertreter dieser Kategorie sind die folgenden Fragen:

- Wie alt sind Sie?
- Wieviele Stunden haben Sie durchschnittlich für die Bearbeitung eines Übungsblattes benötigt?
- An wieviel Prozent der Übungen haben Sie teilgenommen?

Derartige Fragen lassen sich auch im Multiple-Choice-Stil formulieren. Allerdings erfordert die Frage nach dem Alter eine sehr große Menge möglicher Antworten. Auf die anderen beide Beispielfragen sind theoretisch sogar unendlich viele mögliche Antworten denkbar, da auch nicht-ganzzahlige Antworten sinnvoll sind.

Man könnte daher unter Verlust des sehr hohen *Rationiveaus* eine begrenzte Menge an Antwortmöglichkeiten vorgeben, indem man vordefinierte Intervalle zu jeweils einer Alternative zusammenfasst (z.B. 0-1h, 1h-1,5h, ... mehr als 5h). Dies würde die Gestaltung des Abstimmungs-systems vereinfachen, da anstatt eines weiteren Fragetyps auch hierfür die ohnehin benötigten Multiple-Choice Fragen verwendet werden könnten.

Allerdings interpretieren nach [Scholl 03b, Kapitel 7.2] die Befragten das vorgegebene Antwortspektrum als "Indiz für die tatsächliche Verteilung in der Bevölkerung" und schätzen anhand des hierdurch induzierten Mittelpunktes ihr eigenes Verhalten ein, was zu einer Verzerrung des Ergebnisses führen kann. Dies zeigt sich besonders deutlich in einem laut [Scholl 03b] häufig

zitierten Experiment, bei dem mit der Split-Ballot-Technik zwei unterschiedliche Antwortskalen vorgegeben wurden, auf denen die Befragten ihr wöchentliches Fernsehverhalten einordnen sollten. Das Ergebnis in Abbildung 2.1.3 belegt den beschriebenen Effekt.

Antwortskala A		Antwortskala B
bis 1/2 Stunde		
1/2 bis 1 Stunde		
1 bis 1 1/2 Stunden		
1 1/2 bis 2 Stunden		
2 bis 2 1/2 Stunden	\sum 83,8% vs. 62,5%	bis 2 1/2 Stunden
mehr als 2 1/2 Stunden	16,2% vs. \sum 37,5%	2 1/2 bis 3 Stunden
		3 bis 3 1/2 Stunden
		3 1/2 bis 4 Stunden
		4 bis 4 1/2 Stunden
		mehr als 4 1/2 Stunden

Vgl. [Scholl 03b, Kapitel 7.2, Seite 203])

Abbildung 2.1: Verzerrung durch Intervallvorgabe

Ratingskalen

Der größte Anteil an Fragen der im Rahmen dieser Arbeit zu erstellenden Fragebögen betrifft die "Erhebung von Einschätzungen", "Meinungen" und "Verhaltensweisen". Nach [Scholl 03b, Kapitel 5.6] werden für derartige Fragen oft "Ratingskalen benutzt, die Ordinaldaten- oder Intervalldatenniveau haben". Derartige Skalen werden in [Diekmann 95] und [Noelle-Naumann 96] auch als *Likert-Skalen* bezeichnet. Die Entscheidung bei der Gestaltung des Fragebogens über die Anzahl der Stufen einer Ratingskala erfordert eine Güterabwägung zwischen Genauigkeit und Zuverlässigkeit des Fragebogens.

"Je mehr Stufen vorgegeben werden, desto genauer ist die Skala, allerdings wird die Wahl einer bestimmten Stufe unzuverlässiger und zufälliger, weil der Befragte zu viele Auswahlmöglichkeiten hat." ([Scholl 03b, Kapitel 5.6])

In der Praxis finden sich oftmals Ratingskalen mit vier bis sieben Stufen, doch auch deutlich feinere Skalen werden gelegentlich unter Inkaufnahme der beschriebenen Nachteile einer reduzierten Zuverlässigkeit eingesetzt.

Ein weiterer wichtiger Aspekt bei der Gestaltung einer Ratingskala ist der Unterschied zwischen einer *geraden* oder *ungeraden* Anzahl von Stufen. Eine ungerade Anzahl an Stufen impliziert die Existenz einer "Mittelkategorie", die nach [Scholl 03b] den Vorteil birgt, daß der Befragte eine unentschiedene Meinung ausdrücken kann, aber mit dem Nachteil einhergeht, daß diese Mittelkategorie auch dann gewählt wird, wenn der Befragte die Frage gar nicht beantworten will oder kann. Dieser Nachteil lässt sich durch das Anbieten einer Enthaltungsmöglichkeit umgehen.

Demgegenüber erzwingt eine gerade Anzahl von Stufen eine gewisse Tendenz der Meinung, wobei diese Tendenz aber auch künstlich erschaffen wird, da die Befragten keine neutrale Meinung ausdrücken können. Nach [Scholl 03b] ist es auch denkbar, daß der Befragte sich "in seinem Antwortverhalten eingeengt fühlt und daß er deshalb häufiger die Beantwortung verweigert". Bei einer in diesem Fall angebotenen Enthaltungsmöglichkeit besteht dann umgekehrt die Gefahr, daß diese von den Testpersonen als Mittelkategorie fehlinterpretiert wird.

Sortieraufgaben

Eine weitere Fragetechnik, die in der empirischen Sozialforschung oft eingesetzt wird, besteht darin, eine Menge von Aussagen nach dem Grad der Zustimmung zu ordnen. Ein derartiges Verfahren muß zunächst in Testbefragungen eingemessen werden, um festzustellen, ob die angebotenen Möglichkeiten konsistent sortiert werden und damit eine Messung auf *Ordinalniveau* gestatten [Diekmann 95, Seite 237ff].

Diese Technik wird oft eingesetzt, wenn die *Programmfrage*, die der eigentliche Forschungsgegenstand ist, z.B. aufgrund ihrer sozialen Brisanz nicht direkt im Interview gestellt werden kann, sondern durch entsprechend codierte *Testfragen* Rückschlüsse auf das eigentlich Forschungsobjekt möglich sind. Die in [Noelle-Naumann 96] näher beschriebene Problematik bei der Konstruktion geeigneter Testfragen für eine Programmfrage (die sogenannte Codierung), ist für unser Umfragethema von wenig Belang, da unsere Programmfragen direkt als Testfragen stellbar sind. Dieser Umstand wirkt sich positiv auf die im nächsten Abschnitt beschriebene Güte unserer Messungen aus.

2.1.4 Gütekriterien einer Messung

Laut [Diekmann 95, Seite 216] soll eine Messung "objektiv, zuverlässig und gültig" sein. Wir untersuchen daher die Durchführung von Vorlesungsumfragen in Hinblick auf die geforderten Gütekriterien.

Objektivität

Die Objektivität einer Umfrage wird bei [Diekmann 95] differenziert in die *Durchführungsobjektivität* und die *Auswertungsobjektivität*.

Bei unserem Forschungsgegenstand der Vorlesungsbewertung ist das Erzielen einer hohen Durchführungsobjektivität leicht erreichbar, da unsere Umfragetechnik keinen menschlichen Interviewer einsetzt. Mögliche Gefahren für die Durchführungsobjektivität liegen in einer unzureichenden Schulung des Interviewers, der durch tendenziöse Artikulation der Fragen oder unterschwellige Signale an die Testperson das Umfrageergebnis verfälschen könnte. Lediglich

eine suggestive Frageformulierung könnte in unserem Fall die Durchführungsobjektivität beeinträchtigen.

Auch in Hinblick auf die Auswertungsobjektivität sehen wir keine großen Gefahren, da die Formulierung unserer Fragen und Antworten aufgrund der in Abschnitt 2.1.3 beschriebenen großen Nähe zwischen *Programmfragen* und *Testfragen* nur wenig Interpretationsspielraum lässt.

Reliabilität

Die Reliabilität ist nach [Diekmann 95] ein stärkeres Kriterium als die Objektivität. Sie stellt ein "Maß für die Reproduzierbarkeit von Meßergebnissen"[Diekmann 95, Seite 217] dar. Für uns ist in Hinblick auf die Reliabilität der Umfragen vor allem die Darstellung der Fragebögen interessant. In Abschnitt 2.3.2 stoßen wir auf ein paar Probleme, die sich negativ auf die Reliabilität auswirken können.

Validität

Die Gültigkeit unserer Umfrageergebnisse wird von uns nicht weiter in Frage gestellt. Wenn beispielsweise die Studenten das Tafelbild mehrheitlich als unleserlich beurteilten, können wir davon ausgehen, daß das Tafelbild in der Tat verbesserungsfähig ist, wohingegen eine gewisse Skepsis angebracht wäre, wenn sich 80% der Testpersonen für überdurchschnittlich gute Autofahrer halten. Wie bei der Beurteilung der Auswertungsobjektivität ist auch hier die starke Nähe zwischen *Programmfragen* und *Testfragen* äußerst hilfreich. Allerdings sind nach [Diekmann 95] die obengenannte Objektivität und Reliabilität notwendige Voraussetzungen für das Erreichen einer hohen Validität.

2.2 Beschreibung des existierenden Systems

Der Ablauf einer Vorlesungsbewertung gestaltet sich mit dem existierenden papierbasierten System in der Regel wie folgt:

1. Der bereits existierende und fertig gestaltete Fragebogen wird in einer der Hörerzahl der Veranstaltung angemessenen Menge vervielfältigt.
2. Die vervielfältigten Fragebögen werden in der entsprechenden Lehrveranstaltung ausgelegt, mit der Bitte, sie ausgefüllt zurückzubringen.
3. Direkt in der Veranstaltung oder in einer der nächsten Lehrveranstaltungen haben die Hörer Gelegenheit, die ausgefüllten Fragebögen abzugeben.
4. Die abgegebenen Fragebögen werden von einem Beschäftigten des Lehrstuhls gesichtet und entsprechend des Fragetyps wie folgt ausgewertet:

- (a) Die Stimmabgaben der Fragen mit Ratingskalen werden aggregiert. Die aggregierten Werte werden in ein dem Fragebogen entsprechendes Berichterstattungsformular übertragen und an die dem Lehrstuhl übergeordnete Zentralabteilung übergeben.
 - (b) Die Antworten auf Fragen nach persönlichen Daten wie z.B. Geschlecht, Alter, Semesterzahl werden ebenso aggregiert und weitergeleitet.
 - (c) Die Antworten auf freie Fragen werden nicht nach außen kommuniziert sondern nur lehrstuhlintern zur Kenntnis genommen (und entsprechend gewürdigt).
5. Die aggregierte Auswertung wird idealerweise den Hörern in einer der folgenden Lehrveranstaltungen präsentiert, falls es noch weitere Veranstaltungen gibt. Da die Vorlesungsumfragen meistens gegen Ende des Vorlesungszeitraums durchgeführt werden, liegt die Umfrage oftmals so spät, daß keine Ergebnispräsentation mehr im Rahmen der Veranstaltung möglich ist.
 6. Gegebenenfalls kann der Fragebogen aufgrund von entsprechenden Erfahrungen angepasst werden. Die Fragebögen für reguläre Vorlesungen werden allerdings zentral gestaltet, wobei eine Erweiterung um eigene Zusatzfragen zu speziellen Aspekten der Vorlesung nicht vorgesehen ist.

Wir müssen also folgende fünf Arbeitsschritte ermöglichen, wobei wir die veröffentlichbare Ansicht der Umfrageergebnisse stets als *Auswertung* bezeichnen wollen, und die interne, nicht zur Veröffentlichung geeignete Ergebnisansicht als *Report* :

1. *Erstellung* eines Fragebogens und *Konfiguration* des Zeitraums, während dem die Bewertungen abgegeben werden können
2. Ausgabe und Annahme *Fragebögen*
3. Aggregation und Erstellung einer veröffentlichbaren *Auswertung ohne freie Antworten*
4. Erstellung eines *internen Reports nur für befugte Personen*, der auch die freien Antworten enthält
5. Anzeige einzelner ausgefüllter Fragebögen, um spezielle freie Antworten zu verstehen und in den Gesamteindruck des Bewertenden einordnen zu können

Neben diesen geschilderten Arbeitsschritten, die wir auf jeden Fall nachbilden müssen, untersuchen wir nun einige Aspekte, in denen die existierende papierbasierte Lösung sich von elektronischen Lösungen unterscheidet.

2.3 Vergleich papierbasierter und elektronischer Systeme

Unabhängig von der in Kapitel 3.1 zu treffenden Entscheidung für ein Web-basiertes System vergleichen wir grundlegende Eigenschaften elektronischer und papierbasierter Umfragesysteme.

Als einen zentralen Aspekt wollen wir dabei die Anonymität der Umfrage ansehen. Eine ehrliche Stimmabgabe kann möglicherweise mit harscher Kritik am Dozenten einhergehen. Aufgrund des gegebenen Abhängigkeitsverhältnisses der Studierenden vom Dozenten müssen wir eine gewisse Scheu seitens der Studierenden voraussetzen, ihre Ansichten kundzutun, wenn die Gefahr besteht, daß sie mit ihren Äußerungen in Verbindung gebracht werden können. Einige der dabei von uns konstruierten Szenarien mögen weniger wahrscheinlich oder realistisch erscheinen, doch halten wir sie zumindest für denkbar und möchten durch das eine oder andere bewusst extrem konstruierte Beispiel die vielschichtige Komplexität des Themas umreißen.

2.3.1 Fragebogengestaltung

Das heutige papierbasierte System bietet den gesamten Fragenkatalog an, inklusive Fragen zu Übungen, auch wenn zur zu bewertenden Vorlesung gar keine Übung angeboten wird, da der Aufwand maßgeschneiderter Umfragebögen in keinem Verhältnis zum Nutzen steht.

Während auf dem Papier die Struktur gut ersichtlich ist, so daß die nicht passenden Fragen nicht weiter stören, leidet die elektronische Darstellung unter dem begrenzten Platz auf einem Bildschirm.

Der Aufbau des elektronischen Formulars sollte also modular sein, so daß man es aus bestehenden Fragenblöcken entsprechend den Gegebenheiten der zu bewertenden Vorlesung zusammenbauen kann. Zusätzlich sollten einzelne unpassende Fragen aus dem resultierenden Fragebogen entfernt werden können.

Wenn Fragen versehentlich redundant erstellt werden, können die in Abschnitt 2.4.2 beschriebenen Probleme bei einer Auswertung über Vorlesungsgrenzen hinweg auftreten.

2.3.2 Fragebogendarstellung

Während die Darstellungsform des Papierfragebogens einmal festgelegt wurde und keine individuellen Gestaltungsansprüche erfüllen muß, kann von einem elektronischen System erwartet werden, daß es sich in Hinblick auf die Verwendung von Farben und Schriftarten in das universitäre Umfeld integriert. Zusätzlich zu dieser eher kosmetischen Anforderung betrachten wir nun zwei weitere Aspekte der Fragebogendarstellung.

Viele kleine Formulare oder großes Formular?

Der auf einem Bildschirm zur Verfügung stehende Platz ist derart knapp bemessen, daß selbst unter Berücksichtigung der in Abschnitt 2.3.1 motivierten Minimalisierung des Fragebogens, eine komplette Darstellung nicht möglich ist.

Eine verbreitete Technik in Online-Umfragen, die auch in Beispiel 2.4 (Perspektive Deutschland) verwendet wird, spaltet die Umfrage in viele einzelne Frageblöcke auf, die jeweils vollständig dargestellt werden können. Da der Benutzer den Gesamtumfang des Fragebogens nicht ersehen kann, sollte ihm ein Fortschrittsindikator angezeigt werden.

Die Größe von Bildschirmen streut jedoch sehr stark. Displays mit 640x480 Bildpunkten sind wohl nur noch selten anzutreffen, wie eine Vielzahl von Webseiten nahelegt, die “ideal bei 800x600 Pixel” dargestellt werden. Doch sind mittlerweile Bildschirme mit 1600x1200 Punkten sehr verbreitet, während handliche mobile Endgeräte manchmal nur 640x200 Punkte oder weniger aufweisen. Für welche Bildschirmgröße soll die Darstellung also optimiert werden? Jede fixe Entscheidung vergeudet entweder Bedienungskomfort auf großen Bildschirmen oder erschwert die Bedienung auf kleinen Bildschirmen.

Die Empfehlung von [Welker 05], die Bildschirmgröße zu messen und entsprechend zu berücksichtigen, erfordert im Fall eines Web-basierten System die Aktivierung von Javascript im Webbrowser. Umfrageteilnehmer, die aus Sicherheitsbedenken Javascript prinzipiell deaktiviert haben, könnten sich an dieser Anforderung stören und lieber auf eine Bewertungsabgabe verzichten, so daß hier eine Abwägung zwischen besserem Komfort und potentieller Abschreckung von Teilnehmern zu treffen wäre.

Daher werden wir uns gegen eine Aufsplitterung des Fragebogens in viele kleine Formulare entscheiden. Die von uns gewählte Alternative eines großen Formulars bringt in der Bedienung keine wesentlichen Nachteile, und die vom Browser angezeigte Scrollbar dient als Fortschrittsindikator, da der Umfrageteilnehmer an ihr ersehen kann, wie weit er mit der Beantwortung der Fragen bereits fortgeschritten ist. Somit können Besitzer von großen Bildschirmen einen bequemen Überblick über den Fragebogen gewinnen, während wir uns bemühen werden, auch auf kleinen Bildschirmen eine akzeptable Darstellung zu erzielen.

Darstellung von Ratingskalen

Die in Abschnitt 2.1.3 vorgestellten Fragetypen müssen sowohl beim Einsatz eines papierbasierten Umfragesystems als auch bei einer elektronischen Lösung ansprechend und einfach ausfüllbar dargestellt werden.

Während freie Fragen und Fragen nach Zahlangaben durch übliche Eingabefelder erfasst werden können, bieten sich zur Präsentation von Ratingskalen und Multiple-Choice-Fragen mehrere Möglichkeiten an. Die meisten Darstellungsarten lassen sich sowohl elektronisch als auch auf Papier umsetzen.

Eine verbreitete Möglichkeit zur Darstellung von Ratingskalen sehen wir in den Abbildungen 2.3 und 2.4. Diese sogenannten *Matrixfragen* bestechen nach [Welker 05] durch ihr Aussehen und ihre einfache Umsetzbarkeit. Doch wird ebendort auch davor gewarnt, daß “kaum etwas langweiliger auszufüllen” ist “als solch eine Matrixfrage” [Welker 05, Seite 94], und daß spätestens nach dem fünften Item die Konzentration abnimmt [Welker 05, vgl. Seite94]. Eine weitere

Gefahr bei langen Matrixfragen besteht darin, daß der Kopf mit den Antwortmöglichkeiten oben aus dem Fenster herauscrollt und die Zuordnung der Antworten zu den Eingabeknöpfen nicht mehr hergestellt werden kann.

Die ist besonders bedenklich, wenn zusätzlich eine Option "Weiß nicht" angeboten wird, wie dies in 2.4 der Fall ist. Wenn dort die Kopfzeile nicht mehr einsehbar wäre, könnte man in den unteren Zeilen irrtümlich davon ausgehen, daß eine siebenstufige Ratingskala vorliegt. Durch eine geeignete Absetzung der "Weiß nicht"-Knöpfe wäre dieses Problem abschwächbar. Auch ist es denkbar, daß die Bildschirmbreite zur Darstellung nicht ausreicht, so daß vielleicht nur die ersten vier Ratingstufen sichtbar sind und die Skala daher fehlinterpretiert wird.

Voraussetzung für eine Matrixfrage ist eine identische Optionsmenge für alle Fragen. Wenn die Fragen aufgrund ihrer Formulierung verschiedene Antwortformulierungen erfordern, wie dies in Abbildung 5.2 der Fall ist, kann keine Matrixdarstellung gewählt werden. Wenn jede Ratingstufe mit einer Textumschreibung versehen ist, so würde man vor jede Fragezeile eine Kopfzeile mit der Bedeutung der einzelnen Optionen einfügen. Aufeinanderfolgende Fragen mit gleicher Antwortmenge könnten dabei wiederum zu einer kleinen Matrix zusammengefasst werden. Dennoch wird zur Darstellung der Binnenkopfzeilen viel Platz benötigt.

Wenn die Ratingskala nur am Rand mit textuellen Anknüpfen versehen ist, bietet sich eine Darstellung wie in Abbildung 5.2. In 2.5 sehen wir ein Beispiel für eine gefaltete Ratingskala, die am Knickpunkt und den beiden Enden durch Antworttexte verankert ist.

Zusätzlich zu den bisher beschriebenen Darstellungen ist in einem elektronischen System die Präsentation der Ratingskala als Auswahlliste denkbar. Während einige der oben dargestellten Darstellungsmethoden in Ihrer Übersichtlichkeit profitieren, wenn die Ratingskala nur wenig textuell verankert ist, empfiehlt sich bei Verwendung einer Auswahlliste eine textuelle Beschreibung *jeder* Ratingstufe. In Abbildung 5.2 erkennen wir, daß bei Auswahl einer mittleren Bewertung die Textanker am Rand der Skala nicht mehr sichtbar sind und die teilnehmende Person somit nicht erkennen kann, ob die Bewertung "4" eine gute oder schlechte Bedeutung innehat.

Durch die in Abschnitt 4.1.3 beschriebene Technik ist die Darstellungsart leicht wählbar, wobei aufgrund der hier beschriebenen Effekte eine Form gewählt werden sollte, die mit den eingesetzten Antwortmengen verträglich ist.

2.3.3 Internationalisierung

Die bestehenden Papierfragebögen sind ausschließlich in deutscher Sprache verfügbar. Da die meisten Vorlesungen auf deutsch gehalten werden, darf angenommen werden, daß alle Umfrageteilnehmer der deutschen Sprache mächtig sind.

Da seit einigen Jahren vereinzelt Vorlesungen in englischer Sprache angeboten werden, wäre eine englische Fassung zur Bewertung dieser Vorlesungen wünschenswert. Hierzu müsste der Papierfragebogen einmal übersetzt werden und die entsprechende Fassung kopiert und verteilt werden.

Wenn ein elektronisches System auf Mehrsprachigkeit ausgelegt ist, ist zusätzlich auch eine Wahl der Sprache während der Abstimmung denkbar, so daß beispielsweise ein amerikanischer Gaststudent den Fragebogen in seiner Muttersprache beantworten könnte. Wir werden in Abschnitt 3.3.2 unsere Ansätze zur Internationalisierung näher erläutern.

2.3.4 Handschrifterkennung und Anonymität

Die eingesetzten Papierfragebögen enthalten einen kleinen Anteil an freien Fragen. Die handschriftliche Beantwortung freier Fragen gestattet den auswertenden Mitarbeitern des Lehrstuhls unter Umständen Rückschlüsse auf die bewertende Person. Wenn der auswertende Mitarbeiter die Handschrift einiger Studenten durch die Korrektur von Übungsaufgaben kennt, könnte er auch ohne böse Absicht anhand auffälliger Besonderheiten der Handschrift identifizieren, wer den Fragebogen ausgefüllt hat.

Da bei einem elektronischen System keine handschriftlichen Eingaben vorgenommen werden, stellen freie Fragen kein größeres Problem dar als bei einem papierbasierten System. Dennoch ist unter extremen Umständen ein Rückschluss durch freie Antworten auch bei einem elektronischen System denkbar, wenn sich der Bewertende durch für ihn typische Rechtschreibfehler oder Formulierungen zu erkennen gibt; doch sind diese Umstände derart konstruiert, daß wir es bei dem Hinweis belassen, daß ein elektronisches System die Anonymität des Abstimmenden in diesem Aspekt zumindest nicht stärker gefährdet als das papierbasierte System.

2.3.5 Persönliche Daten und Anonymität

Die Beantwortung von Fragen zur Person kann insbesondere bei einem kleinen Hörerkreis geeignet sein, aufgrund von einmaligen Eigenschaftskombinationen Rückschlüsse auf die bewertende Person zu ziehen (z.B. wenn es nur einen männlichen Hörer aus dem 3. Semester gibt).

Die Darstellung persönlicher Daten ist bei der Anzeige ausgefüllter Fragebögen zu Auswertungszwecken nicht unbedingt notwendig, aber u.U. doch wünschenswert. Ein elektronisches System ist in diesem Punkt zumindest nicht schlechter als die existierende Papierlösung und bietet das Potential einer Verbesserung, indem einzelne persönliche Angaben aus der Ansicht ausgeblendet werden könnten.

2.3.6 Vorzeitige Auswertung

Im Gegensatz zum papierbasierten System, das in der Regel erst ausgewertet wird, wenn die Umfrage für beendet erklärt wird, gestattet das elektronische System von den technischen Gegebenheiten her eine Auswertung zu jedem Zeitpunkt, insbesondere noch während der laufenden Umfrage. Hieraus ergeben sich neue Chancen und Risiken, von denen wir einige einer kurzen Betrachtung unterziehen möchten.

Eine Präsentation der Auswertung lässt sich als mögliche Motivation zur Teilnahme einsetzen (siehe Kapitel 2.3.12), indem diese als “Belohnung” für die Teilnahme nach dem Absenden der Umfrage angezeigt wird. Dies erscheint verführerisch, da die “Belohnung” durch den Einsatz des elektronischen Umfragesystems frei Haus ohne weiteren Aufwand existiert, während der Einsatz anderer Belohnungen mit einem Bereitstellungsaufwand verbunden ist und obendrein mögliche rechtliche Risiken in sich birgt.

Allerdings kann eine Überlappung der Zeiträume, während denen die Umfrage durchgeführt wird und die Auswertung präsentiert wird, zu unerwünschten Rückkopplungseffekten führen und stellt zudem eine Gefahr für die Anonymität der Umfrage dar:

Verzerrende Rückkopplung

Eine ergebnisverzerrende Rückkopplung ist denkbar, da das eigene Abstimmverhalten durch ein bereits vorab bekanntgewordenes Zwischenergebnis beeinflusst werden kann. Wenn der Abstimmende beispielsweise eine durchschnittliche Bewertung abgeben möchte, aber bereits weiß, daß etliche gute Bewertungen abgegeben wurden, so unterliegt der Abstimmende der Versuchung, seine (eigentlich als durchschnittlich gedachte) Bewertung als schlecht einzustufen, um das Endresultat in die von ihm gewünschte Richtung zu verschieben. Er kann also quasi sein Stimmgewicht erhöhen und dadurch das Umfrageergebnis verfälschen. Der Abstimmende könnte ohne entsprechend starke Zugangskontrolle so viele negative oder positive Stimmen abgeben bis ihm das Ergebnis gefällt, und im äußersten Extremfall bestimmt der letzte Abstimmende das Ergebnis. Die Unterbindung mehrfacher Stimmabgaben bewirkt allerdings keinen vollständigen Schutz vor einem derartigen Rückkopplungseffekt, da es denkbar ist, daß mehrere Studenten zusammen die Vorlesungsumfrage ausfüllen. Nach der Stimmabgabe des ersten Studenten können nun alle am Computer anwesenden Kommilitonen das Zwischenergebnis betrachten und ihre eigenen Bewertungen entsprechend anpassen um das Ergebnis mehr in die gewünschte Richtung zu biegen.

Gefährdung der Anonymität

Angenommen Student A betrachtet den Zwischenstand der Auswertung und sieht, daß Student B gerade an der Umfrage teilnimmt. Durch ein erneutes Abrufen des Zwischenergebnisses nach der Abgabe von Student B kann Student A aus den Unterschieden der beiden Zwischenauswertungen Rückschlüsse auf die Bewertung von Student B ziehen. Somit bedeutet die Bereitstellung von *aktuellen* dynamisch erzeugten Zwischenergebnissen eine Gefährdung der Anonymität der Umfrage.

Durch eine Erzeugung von Zwischenergebnissen z.B. regelmäßig jede Nacht oder zu zufälligen Zeitpunkten, ließen sich beide dargestellten Gefahren mildern.

2.3.7 Kontinuierliche Evaluation

Ein weiterer Vorteil von elektronischen Abstimmungssystemen gegenüber papierbasierten Umfragen besteht in der Möglichkeit, kontinuierlich Umfragen durchführen zu können, ohne daß der damit verbundene Arbeitsaufwand entsprechend mitsteigt. Wie wir bereits in der Einleitung dargelegt hatten, sind gerade die Ansichten der Studierenden von Interesse, die die Veranstaltung *nicht* bis zum Ende besuchen. Daher wurde vor einigen Jahren damit begonnen, eine zweite Umfrage in der Mitte des Semesters durchzuführen. Ein Dozent begleitete diesen Schritt mit der Bemerkung, daß man dieser Logik folgend die Umfrage gleich in der ersten Stunde durchführen müsse, um alle Hörer anzutreffen. Dieser nicht ganz ernst gemeinte Kommentar bezieht sich allerdings auf die Absurdität, durch eine aufwendige Papierumfrage die Zufriedenheit mit einer Veranstaltung zu erfassen, über die sich die Besucher der Veranstaltung noch gar kein richtiges Bild machen konnten.

Da beim Einsatz eines elektronischen Umfragesystems die Auswertung automatisch geschieht und auch die Erstellung eines Fragebogens idealerweise ohne großen Aufwand durchgeführt werden kann, erscheint eine kontinuierliche Evaluation von der ersten Vorlesungstag an jedoch als sinnvoll durchführbar.

Eine Möglichkeit für eine kontinuierliche Evaluation besteht darin, *eine* Umfrage bereits *während* des gesamten Vorlesungszeitraums anzubieten und nicht erst am Ende. Eine andere Möglichkeit wäre die Aufspaltung in mehrere Umfragen: eine für jeden Monat, jede Woche, oder gar jeden Vorlesungstag. Letzterer Ansatz würde auch den Einsatz sehr restriktiver Zugangskontrollmechanismen gestatten, die nur eine Stimmabgabe pro Hörer zu lassen, da bei jeder neuen Umfrage erneut alle legitimierte Personen einmal abstimmen dürfen.

Wenn die Zwischenergebnisse während des Semesters nicht nur intern ausgewertet werden, sondern auch an die Studierenden publiziert werden, sollten die in Abschnitt 2.3.6 beschriebenen Rückkopplungseffekte beachtet werden. Darüberhinaus kann auch eine Rückwirkung auf den Besuch der Veranstaltung nicht ausgeschlossen werden: So könnte sich ein Studierender im Falle einer Terminkollision zweier Veranstaltungen aufgrund der veröffentlichten Bewertungen für den Besuch einer Veranstaltung entscheiden. Dieser an sich positive erscheinende Effekt könnte auch negative Nebenwirkungen bewirken, wenn z.B. die "sehr gut" bewertete Veranstaltung überlaufen wird und dadurch ihre Qualität einbüßt. Ob derartige Rückwirkungen in der Praxis auffällig werden, und ob die positiven oder negativen Effekte überwiegen, lässt sich wohl erst bei einem Einsatz der kontinuierlichen Evaluation in großem Rahmen herausfinden. Daher beschränken wir uns an dieser Stelle auf die Andeutung möglicher Auswirkungen.

2.3.8 Automatisierte Stimmabgabe

Im Gegensatz zum papierbasierten System besteht bei einem elektronischen Systemen die Gefahr, daß ein böswillig erstelltes Programm automatisch Bewertungen generiert und in das Umfragesystem einspeist. Ein Weg, dieses Problem zu Umgehen besteht darin, mehrfache Stimm-

abgaben zu unterbinden. Da dies allerdings, wie in Abschnitt 2.3.11 dargelegt, unerwünscht sein kann, betrachten wir einige Möglichkeiten, wie maschinengenerierte Stimmabgaben ausgefiltert werden können.

CAPTCHA

Eine verbreitete Abhilfe gegen automatisierte Zugriffe auf Ressourcen, wie z.B. Anmeldeformulare zu Web-basierten Diensten, besteht im Einsatz sogenannter CAPTCHAs (Completely Automated Public Turing-Test to Tell Computers and Humans Apart, siehe hierzu das CAPTCHATMProject der Carnegie Mellon Universität auf <http://www.captcha.net/>), die dem Benutzer eine Aufgabe stellen, die von einem Menschen sehr leicht gelöst werden kann, aber für ein Programm unlösbar oder nur mit unverhältnismäßig großem Aufwand zu bewältigen ist. In Abbildung 2.2 sehen wir ein typisches Captcha, das einen verzerrt dargestellten Text enthält, der vom Benutzer abgetippt werden muss.

Eine generelle Gefahr beim Aufstellen von Zugangshürden liegt darin, daß auch legitime Benutzer ausgesperrt werden. So wird beim Einsatz von CAPTCHAs oftmals davon ausgegangen, daß die visuelle Darstellung des im Bild verschlüsselten Passwortes einsehbar ist. Weder Blinde noch Anwender mit Text-basierten Browsern wie z.B. lynx können jedoch das dargestellte Passwort erkennen. Für Sehbehinderte sollte daher immer eine akustische Alternative zur Verfügung stehen, indem der Zugangscode vorgelesen wird. Bei kombiniertem Auftreten von Seh- und Hörbehinderungen wird eine diskriminierungsfreie Lösung schwierig. Da die Zielgruppe der jeweiligen Vorlesungshörer jedoch bekannt ist, kann gegebenenfalls auf den Einsatz barriere-induzierender Techniken verzichtet werden.



Abbildung 2.2: Beispiel-Captcha

Auswerten von Netzwerkadressen

Durch das Mitprotokollieren der Kommunikation mit einem elektronischen Abstimmungssystem lässt sich zwar nicht erkennen, ob ein Mensch oder eine Maschine mit dem Abstimmungssystem interagiert, aber wenn man voraussetzt, daß der Angreifer seine automatisierte Bewertungsflut von nur *einem* Computer aus durchführt, eröffnen sich Möglichkeiten der Abwehr. Daß die hier vorgestellten Abwehrtechniken gegen einen verteilten Angriff von einem Netz vieler verschiedener Computer nichts ausrichten können, sehen wir nicht als eklatanten Mangel an, da uns ein

derartig aufwendiger Angriff auf ein eher belangloses Ziel wie ein Vorlesungsumfragesystem unwahrscheinlich erscheint und durch die Antwort auf andere Probleme, wie z.B. die in Abschnitt 2.3.10 näher beschriebene Zugangskontrolle, mitgelöst werden kann.

Eine Möglichkeit, automatisierte Stimmabgaben zu unterbinden, besteht darin, die Netzwerkadresse der Gegenstelle zu erfassen und von jeder Gegenstelle höchstens eine Abstimmung entgegenzunehmen. Dieses Verfahren hat den Vorteil, daß die datenschutzrechtlich brisante Clientadresse nicht in Verbindung mit den Abstimmungsergebnissen gespeichert werden muss, sondern in einer getrennten Datenbank als "benutzt" vermerkt werden kann. Allerdings werden hierdurch mehrere Stimmabgaben des gleichen Computers unterbunden, selbst wenn diese durch mehrere verschiedene legitime Personen durchgeführt würden. Verschärft wird dieses Problem durch den Umstand, daß der Einsatz von Proxies oder Network Address Translation (NAT) dazu führt, daß verschiedene Computer unter der gleichen Adresse erscheinen können. Beispielsweise erscheint es uns nicht wünschenswert, mehreren Hörern derselben Veranstaltung nur eine Stimme zu gewähren, nur weil diese in einer Wohngemeinschaft leben und sich ihre Internetanbindung teilen.

Man könnte nun vorschlagen, die Grenze der erlaubten Stimmanzahl höher festzusetzen und z.B. fünf Stimmen pro Computer zulassen, doch ist nicht auszuschliessen, daß im Extremfall sogar alle abgegebenen Stimmen legitim von ein und demselben Computer auszugehen scheinen, wenn z.B. alle Studenten vom CIP-Pool aus über den zentralen Proxy an der Umfrage teilnehmen.

Daher können wir von einem strikten Ausschlussverfahren auf Basis von Netzwerkadressen nur abraten.

Eine andere Möglichkeit, die Verbindungsinformationen gewinnbringend auszuwerten, erfordert eine Speicherung *in Verbindung* mit den Abstimmungsdaten, gefolgt von einer *manuellen Durchsicht* auf Auffälligkeiten. Dabei werden beliebig viele Stimmen pro Netzwerkadresse zunächst prinzipiell zugelassen, und erst nach der Abstimmung wird untersucht, ob eine Auffällige Häufung einzelner Adressen auftritt und die damit verbundenen Bewertungen eine gewisse Tendenz aufweisen.

Im Gegensatz zu Umfragen, die sich an einen großen Ausschnitt der Bevölkerung richten, ist in unserem Umfrageszenario die Anzahl der Vorlesungshörer ungefähr bekannt, so daß, selbst wenn die z.B. 5000 abgegebenen Stimmen genau der erwarteten Stimmverteilung entsprechen und keine Tendenzen einer Ergebnisbeeinflussung zeigen, klar ist, daß nicht alle diese Stimmen legitim sind. Die Problematik, Stimmen manuell auszufiltern, wird uns in Abschnitt 2.3.9 nochmals begegnen.

Der Ansatz, Verbindungsdaten zum Zwecke einer späteren Analyse von potentiell unberechtigt abgegebenen Bewertungen mitabzuspeichern, ist datenschutzrechtlich bedenklich und erfordert darüberhinaus Arbeitsaufwand für die Analyse und wirkt somit unserer Absicht entgegen, den Auswertungsprozess zu vereinfachen. Dennoch stellen wir mit dem in Abschnitt 4.7.14 vorgestellten Modul eine technische Umsetzung zur Verfügung.

Auch die in Abschnitt 2.3.9 dargestellte Technik, die zum Ausfüllen benötigte Zeit zu erfassen

kann gegen eine automatisierte Stimmabgabe eingesetzt werden, solange das angreifende Programm keine Anstrengungen unternimmt, ein realistisches Zeitverhalten der Stimmabgabe zu simulieren.

2.3.9 Unplausible Stimmabgaben

Die in Abschnitt 2.3.12 angeregten Wege, die Teilnahmequote an elektronisch durchgeführten Umfragen zu steigern, führen ein gewisses Zwangsmoment ein, so daß die Teilnahme an der Umfrage nicht mehr ganz freiwillig erfolgt, sondern einige Studierende nur wegen der gegebenen Motivationsanreize eine Bewertung abgeben werden. Eine derartige Bewertung wird wahrscheinlich weniger tief überlegt sein und stellt uns vor das Problem, ernst gemeinte Bewertungen von weniger ernst gemeinten zu trennen.

Auch das papierbasierte System kann unter diesem Effekt leiden, wenn den Studierenden z.B. bei geeigneter Gelegenheit, wie der Scheinausgabe, nahegelegt wird, den Fragebogen auszufüllen.

Plausibilitätskontrolle durch Kontrollfragen

Ein sowohl für papierbasierte als auch elektronische Umfragen gangbarer Weg, unplausible Abgaben zu erkennen, liegt in der Verwendung von Kontrollfragen in den Fragebogen, die zweimal einen möglichst ähnlichen Sachverhalt abfragen und bei einer ernstgemeinten Bewertung entsprechend konsistent beantwortet werden sollten. Idealerweise würde man die beiden zu korrelierenden Fragen einmal positiv und einmal negativ formulieren, so daß ein Ankreuzen der Antwort "ganz links" nicht zu einem scheinbar plausiblen Ergebnis führt.

Möglich Probleme beim Einsatz von Kontrollfragen bestehen einerseits in ihrem Widerspruch zur Prämisse, den Fragebogen möglichst kurz zu fassen (vgl. [Welker 05], [Noelle-Naumann 96]), und andererseits in der Gefahr, einen schlecht durchdachten Eindruck zu hinterlassen, wenn erkennbar ist, daß der Fragebogen zweimal das gleiche fragt. Insbesondere durch die anzustrebende Kürze des Fragebogens steigt die Gefahr, daß die absichtlich eingebaute Redundanz auffällig wird, während bei einem 20-seitigen Fragebogenbuch eine Kontrollfrage leichter versteckbar ist.

Plausibilitätskontrolle über die Ausfülldauer

Ein anderer Ansatz, die Ernsthaftigkeit zu erfassen, mit der ein Fragebogen ausgefüllt wurde, wird durch die technischen Gegebenheiten einer elektronischen Umfrage ermöglicht, indem man die Zeit stoppt, die zum Ausfüllen des Fragebogens benötigt wurde.

Die gewonnene Zeitinformation kann direkt zum Verwerfen offensichtlich unplausibler Bewertungen verwendet werden. Eine mögliche Implementierung dieses Ansatzes werden wir in Abschnitt 4.7.2 vorstellen. Allerdings läuft man dabei Gefahr, durch eine zu hoch angesetzte Schran-

ke legitime Bewertungen zu verwerfen, wenn man z.B. davon ausgeht, daß ein Verständnis des Fragebogen vier Minuten erfordert, aber eine ganze Gruppe von Studierenden den Fragebogen zusammen ausfüllt, so daß nach der ersten Abgabe die weiteren Teilnehmer bereits mit den Fragen vertraut sind, sich ihre Gedanken gemacht haben und dennoch *ihren* Fragebogen in weniger als zwei Minuten ausfüllen können.

Besonders verheerend wäre eine zu hoch festgesetzte Mindest-Dauer, sei es durch falsche Abschätzung oder durch einen Konfigurationsfehler, da im Extremfall alle abgegebenen Bewertungen unwiederbringlich verworfen würden.

Schlecht programmierte automatische Bewertungsprogramme (vgl. Abschnitt 2.3.8), die zum “Ausfüllen” nur wenige Sekunden benötigen ließen sich allerdings durch eine entsprechend konservativ konfigurierte Kontrolle der Ausfülldauer ausfiltern.

Zum Zwecke der Plausibilitätskontrolle beschreiben wir aufgrund der dargelegten Probleme eine sanftere Alternative, indem die gemessene Zeit, anstatt sie sofort zum Aussieben zu verwenden, zusammen mit dem Fragebogen abgespeichert wird. Auffällig kurze Abgaben können durch eine (allerdings arbeitsaufwendige) manuelle Durchsicht auf Plausibilität untersucht werden, indem man z.B. beurteilt, ob ein differenziertes (also nicht nur “ganz links angekreuztes”) Meinungsbild vorliegt, das dennoch konsistent erscheint (und daher wohl nicht wahllos zufällig verteilt angekreuzt wurde). Für eine Umsetzung dieser Strategie verweisen wir auf Abschnitt 4.7.13.

Ein manuelles Aussieben von einzelnen Bewertungen birgt in sich allerdings die Gefahr, das Ergebnis absichtlich oder unterbewusst in eine Richtung verfälschen zu können, so daß sich die auswertenden Personen unter extremen Umständen mit dem Vorwurf der Manipulation konfrontiert sehen können.

Unter sehr konstruierten Umständen kann durch eine einzelne außergewöhnlich lange Ausfülldauer auf die Identität der ausfüllenden Person rückgeschlossen werden, wenn diese z.B. aufgrund einer Behinderung deutlich mehr Zeit zur Bedienung des elektronischen Systems benötigt als andere Umfrageteilnehmer.

2.3.10 Stimmabgabe durch unberechtigte Personen

Da die Papierfragebögen in der Vorlesung ausgegeben werden stehen diese Personen außerhalb des Universitätsumfeldes nicht zur Verfügung, wohl aber Hörern anderer Vorlesungen, die z.B. die liegen gebliebenen Fragebögen im Hörsaal finden. Ein elektronisches System – zumindest unsere anvisierte Web-basierte Fassung – kann sich jedoch im Prinzip von jedermann nutzen lassen.

Zusammen mit den in Abschnitt 2.3.11 beschriebenen Verfahren, die eine noch strengere Zugangsbeschränkung zum System darstellen als in diesem Abschnitt, steht ein breites Spektrum an Möglichkeiten zur Verfügung, den Zugang zum Abstimmungssystem zu regeln. Die genaue Bewertung der möglichen Vor- und Nachteile jeder denkbaren Lösung hängt von einer Vielzahl von Faktoren ab und wird letztlich vom jeweiligen Veranstalter entschieden.

In diesem Abschnitt befassen wir uns nur mit Verfahren zur Unterscheidung zwischen legitimen und nicht legitimen Personen, ohne Betrachtung individueller Unterscheidungsmerkmale der Personen innerhalb der beiden Gruppen, so daß hier keine Gefährdung der Anonymität auftritt.

Einschränkung der abstimmungsberechtigten Netzwerkadressen

Eine Beschränkung der Abstimmung z.B. auf das Universitätsrechnernetz wäre leicht realisierbar und würde den Zugang für universitätsfremde Personen wirksam unterbinden. Allerdings stellt eine derartige Beschränkung auch ein Hindernis für abstimmungswillige Vorlesungshörer dar, die in Ruhe von zu Hause aus an der Umfrage teilnehmen möchten. Allerdings ist es heutzutage aufgrund der technischen Entwicklung durchaus anzunehmen, daß der Student sich von zu Hause aus z.B. über eine VPN-Lösung oder über `ssh` Zugang zum Institutsnetz verschafft und somit trotz Einschränkung der Umfrage auf das Universitätsnetz an der Vorlesungsbewertung teilnehmen kann.

Ein Nachteil liegt in der unter Umständen sehr komplexen Netzwerkstruktur der Universität, die eine Vielzahl verschiedener freizuschaltender Adressblöcke umfassen kann. Einige Lehrstühle setzen eine derartige Beschränkung für alle ihre Lehrinhalte ein. Beim Einsatz in einem derartigen Szenario (z.B. wenn das Umfragesystem auf einem zugangsbeschränkten Lehrstuhlserver läuft), ist die notwendige Information über erlaubte Adressen bereits an anderer Stelle vorhanden und muß nicht innerhalb des Vorlesungsumfragesystems vorgenommen werden.

Geheimhalten des Zugangsweges

Die Information, wie man das Vorlesungsbewertungssystem erreichen kann, muß nicht öffentlich publiziert werden. Es reicht, wenn in der Vorlesung die notwendige Information kommuniziert wird, so daß nur eingeweihte Vorlesungshörer den Zugang zur Abstimmung finden ("security by obscurity"). Das Risiko, daß die relevante Information nach außen diffundiert, ist durchaus gegeben (z.B. indem der Webbrowser die angesurfte URL des Systems gespeichert hat und vorlesungsfremden Mitbenutzern des Computers zugänglich macht), doch sehen wir die hieraus erwachsenden Gefahren als eher gering an. Vorteile dieser Lösung liegen in der engen Beschränkung auf Vorlesungshörer bei gleichzeitig keinerlei technischem Aufwand.

Vorlesungspasswort

Eine weitere anonyme Form der Zugangskontrolle besteht darin, allen Hörern einer Veranstaltung ein gemeinsames Passwort mitzuteilen, durch das sie den Zugang zum Abstimmungssystem erhalten. Eine Umsetzung dieses Ansatzes liegt mit dem in Abschnitt 4.7.11 vorgestellten Modul vor.

Vorlesungszertifikat

Im Rahmen der in Abschnitt 2.3.7 beschriebenen kontinuierlichen Evaluation ist es für die Teilnehmer möglicherweise ärgerlich, sich daß Zugangspasswort über einen längeren Zeitraum zu merken. Auch ein Notizzettel kann verloren gehen oder vergessen werden. Durch das Ausstellen eines Zertifikats, das alle Vorlesungshörer auf ihren Computern installieren würden, könnten diese als abstimmungsberechtigt erkannt werden, ohne jedesmal das Passwort erneut eingeben zu müssen.

Derselbe Effekt lässt sich beim Einsatz eine Vorlesungspasswortes erzielen, wenn die verwendete Clientsoftware in der Lage ist, das einmal eingegebene Passwort zu speichern.

2.3.11 Mehrfache Stimmabgabe

Beim papierbasierten System wird eine strikte Ausgabe von genau einem Fragebogen an einen Studenten nicht kontrolliert, da dies einen hohen Aufwand erfordern würde und die Motivation zur Teilnahme an der Vorlesungsbewertung reduziert würde, wenn es nicht möglich wäre, für einen verhinderten Kommilitonen einen Fragebogen mitzunehmen. Wenn ein Vorlesungshörer jedoch gleich einen dicken Stapel Fragebögen an sich nehmen würde, kann davon ausgegangen werden, daß dies wahrscheinlich auffallen würde.

Eine mehrfache Abgabe einer Bewertung ist bei dem papierbasierten System somit möglich, allerdings nur in einem relativ geringen Rahmen (z.B. sind fünf abgegebene Bewertungen des gleichen Hörers möglich, aber wohl kaum 50).

Das elektronische System dagegen steht mehrfacher Stimmabgabe gegenüber zunächst komplett offen. Auch wenn die in Abschnitt 2.3.10 vorgestellten Mittel zum Einsatz kommen und der Zugang auf legitimierte Personen beschränkt ist, können diese dennoch beliebig viele Bewertungen abgeben. Wir beurteilen nun einige Techniken, diesen Makel zu beseitigen.

Identifikation über die Netzwerkadresse

Eine einfacher Ansatz, mehrfache Stimmabgaben zu unterbinden, besteht darin, von jedem Computer nur eine Bewertung anzunehmen. Diese Technik wird von uns bei der Betrachtung von automatisierten Stimmabgaben in Abschnitt 2.3.8 detailliert erörtert und ist aufgrund der dort beschriebenen Effekte ungeeignet, eine mehrfache Stimmabgabe zu unterbinden.

Identifikation über die Matrikelnummer

An manchen Lehrstühlen werden Klausurergebnisse mit dazugehörigen Matrikelnummern als Aushang veröffentlicht. Hierdurch sind Matrikelnummern nicht mehr als geheim anzusehen, und daher zur Identifikation ungeeignet, da ein und dieselbe Matrikelnummer mehreren Studierenden

bekannt ist. Dieser Ansatz wird von uns daher als ungeeignet betrachtet, da fremde Matrikelnummern verwendet werden könnten um eine mehrfache Bewertungsabgabe durchzuführen, während gleichzeitig andere legitime Umfrageteilnehmer ausgesperrt würden, wenn ihre Matrikelnummer bereits “verbrannt” wurde.

Persönliches Passwort

Durch die Ausgabe von persönlichen Passwörtern kann eine mehrfache Bewertung durch dieselbe Person unterbunden werden.

Die Anonymität der Umfrageteilnehmer ist aber nicht mehr offensichtlich gegeben, so daß unter Umständen dem Umfragesystem ein gewisses Misstrauen entgegengebracht wird..

Technisch bietet es sich an, die Zugangspasswörter in einer separaten Datenbank außerhalb des Evaluationssystems zu speichern. Allerdings eröffnet eine gekoppelte Speicherung die Möglichkeit, die eigene Meinung nachträglich ändern zu können. Dies wäre in Zusammenhang mit der in Abschnitt 2.3.7 vorgeschlagenen kontinuierlichen Evaluation hilfreich, da es den Vorlesungshörern auf diese Weise erspart bliebe, jede Woche erneut den Fragebögen mit der immernoch unveränderten Meinung über die Veranstaltung auszufüllen.

Anonymisierte Passwörter

Alle Vorzüge von persönlichen Passwörtern lassen sich auch erreichen, ohne daß eine Zuordnung von Passwort zu Person möglich ist. Wenn die Passwörter auf eine anonyme Weise verteilt werden, sollten auch keine Bedenken mehr seitens der Studierenden bestehen. Diese Passwörter liessen sich ähnlich wie eine Transaktionsnummer (**TAN**) einmalig verwenden, oder auch permanent zum Zwecke einer kontinuierlichen Evaluation.

Mehrere verschieden aufwendige Ausgabeverfahren sind denkbar. Im einfachsten Fall druckt man eine Liste mit den Passwörtern aus und lässt jeden Umfrageteilnehmer ein Passwort abreißen. Da die Passwörter dabei einsehbar sind, empfiehlt es sich, eher schwer merkbare Symbolkombinationen zu generieren, wie z.B. “b7fwb83iwdk”. Ein aufwendigeres Ausgabeverfahren erfordert eine Vorbereitung der einzelnen Passwörter durch Ausschneiden und Falten. Da die generierten Passwörter hierbei nicht einsehbar sind, können auch leichter merkbare Zugangscodes erzeugt werden und somit der Bedienkomfort für die Teilnehmer gesteigert werden (man vergleiche die Merkbarkeit von “Waikiki-Hamburg-Knödel” mit “b7fwb83iwdk”).

Persönliches Zertifikat

Die am Ende von Abschnitt 2.3.10 getroffenen Feststellungen treffen auch zu, wenn für jeden Teilnehmer ein eigenes Zugangs-Zertifikat erstellt wird. Die Zertifikate lassen sich im Prinzip

anonym verteilen, werden aber wohl aufgrund der Datenmenge nicht von Hand abgetippt werden. Wenn eine Verteilung von Datenträgern auf Grund des Erstellungsaufwandes ausscheidet verbleibt das Herunterladen der ausgestellten Zertifikate, wodurch das anonyme Zugangsproblem nur verschoben wird.

Interessant ist der Einsatz persönlicher Zertifikate, wenn eine bereits existierende Infrastruktur genutzt werden kann, in der jeder Studierende ein persönliches Zertifikat ausgestellt bekommen hat.

Ein Problem erwächst allerdings gerade aus dem Komfort, daß die Umfrageteilnehmer eben *kein* Passwort eingeben müssen, und dennoch *identifiziert* werden können, so daß sie sich scheinbar anonym fühlen, aber es technisch nicht wirklich in jedem Moment sind. Insbesondere im Falle einer Kompromittierung des Abstimmungssystems können hieraus unerfreuliche Probleme erwachsen.

2.3.12 Rücklaufquote

Die Erfahrung mit existierenden elektronischen Umfragesystemen zeigt ein Problem auf, das den Erfolg eines derartigen Systems – trotz gebührender Adressierung der bereits erwähnten Probleme – gefährdet: Etliche bereits eingesetzte Umfragesysteme leiden unter der sehr schwachen Beteiligung der potentiellen Umfrageteilnehmer. So beobachteten wir, daß beim jährlich stattfindenden Chaos Communications Congress des Chaos Computer Clubs (CCC) nach dem Einsatz elektronischer Fragebögen zur Befragung der Kongressbesucher wieder auf Papierfragebögen zurückgewechselt wurde. Auch im von uns betrachteten universitären Vorlesungsumfeld wurden wir von den mit der Vorlesungsevaluation betrauten Mitarbeitern über das Problem mangelnder Umfragebeteiligung unterrichtet. So wurde uns beispielsweise von der zentralen Universitätsverwaltung mitgeteilt, daß die Rücklaufquote bei Papierfragebögen am Institut für Informatik bei 50% liegt, wohingegen das vor wenigen Jahren eingeführte elektronische System der Fakultät für Physik eine Beteiligung von unter 20% erreicht.

Eine mögliche Deutung dieser Erkenntnis besteht in der Annahme, daß das alte papierbasierte System eine Art "Futterneid" bei der Ausgabe der Blätter bewirkte: Da die anderen Studenten sich ein Blatt nehmen, will man selbst auch ein Blatt ergattern. Eine Nachmodellierung dieses Effekts in elektronischen Systemen ist wohl nicht möglich. Daher ziehen wir andere Wege in Betracht, die Rücklaufquote beim Einsatz eines elektronischen Systems zu erhöhen, indem wir dem Abstimmenden nach erfolgter Abgabe seiner Bewertung eine Belohnung (Bonbon) zukommen lassen.

Ein Problem entsteht, wenn der Bewertende das Bonbon nochmals abrufen möchte. Ohne weitere Vorkehrung wäre er zu einer mehrfachen Bewertungsabgabe gezwungen, die weder ihm zuzumuten ist noch im Interesse der Auswertung liegt. Daher wird dem Bewertenden zusammen mit dem Bonbon idealerweise ein Bypass-Passwort mitgeteilt, das er zum erneuten Abrufen des Belohnungsmaterials angeben kann.

Wir unterscheiden zwei prinzipielle Ausprägungen einer Belohnung: eine einheitlich gleiche Belohnung für alle Teilnehmer und individuell zugeschnittene Belohnungen.

Vorlesungsbonbon

Ein einheitliches Bonbon für alle Hörer der Vorlesung ist vergleichsweise unproblematisch zu realisieren, da es nur einmal pro Vorlesung eingerichtet werden muß.

Unter Umständen ist das Bonbon von allgemeinerem Interesse. In diesem Fall wird ein Zugriffsschutz für den Fragebogen notwendig um keine Phantasie-Bewertungen von vorlesungsfremden Personen zu erhalten, die nur das Bonbon erhalten wollen.

Eine Weitergabe des Bonbons an andere Hörer könnte deren Anreiz zur Abstimmung reduzieren. Auch wenn sich eine Weitergabe des Bonbons nicht vollständig unterbinden lässt versuchen wir in Abschnitt 4.7.12 zumindest den direkten Zugriff auf das Bonbon zu unterbinden.

Als Anreiz ist z.B. die Musterlösung des letzten Übungsblattes denkbar, oder im Falle der in Abschnitt 2.3.7 vorgeschlagenen kontinuierlichen Evaluation die Musterlösung zum jeweils aktuellen Übungsblatt. Vor dem Einsatz von Übungsblättern, Musterlösungen oder ähnlichen Lehrmaterialien als Bonbon empfehlen wir eine rechtliche Klärung etwaiger Probleme, die entstehen können, wenn ein Studierender sich im Zugang zu Lehrmaterial eingeschränkt fühlt, weil er an der Umfrage teilnehmen *muß* um an die Unterlagen zu kommen. Wenn die entsprechenden Materialien auch auf anderem Wege bezogen werden können, z.B. im Sekretariat, lässt sich dieses Problem wohl entschärfen.

Ein anderes Bonbon, das sich geradezu anbietet, um die Teilnahme an der Umfrage zu erhöhen, besteht in dem Ergebnis der Umfrage selbst. Mögliche Implikationen hieraus haben wir in Abschnitt 2.3.7 dargelegt.

Individuelles Bonbon

Im Gegensatz zu einem Vorlesungsbonbon muss ein individuelles Bonbon dynamisch anhand der Benutzeridentifikation des Abstimmenden ermittelt werden.

Dies bedeutet einerseits einen erhöhten Aufwand bei der Erstellung der individuellen Bonbons und andererseits, daß die Identität des Umfrageteilnehmers erfasst werden muß. Hierdurch können die Umfrageteilnehmer die Anonymität der Umfrage in Frage stellen.

Denkbar ist eine Umsetzung, bei der die abstimmende Person sich zumindest entscheiden kann, ob sie das Bonbon erhalten will oder anonym verbleiben möchte, indem erst *nach* der Abgabe der Bewertung die zur Ausgabe des Bonbons notwendigen persönlichen Daten erfragt werden. Im Rahmen der technischen Implikationen behandeln wir diesen Ansatz vertieft in Abschnitt 4.7.1.

Als individuelles Bonbon bietet sich z.B. die Scheinnote des jeweiligen Vorlesungshörers an, noch bevor die Noten offiziell verkündet werden.

Vorteil derartiger individueller Bonbons gegenüber von einheitlichen Vorlesungsbonbons ist, daß ein Verbreiten des erhaltenen Bonbons den Anreiz für weitere Teilnehmer nicht reduziert, da diese ihre *eigene* Belohnung erhalten wollen.

Direkte Bitte um Teilnahme

Eine weitere Möglichkeit zur Steigerung der Beteiligung an der Umfrage lässt sich gleichermaßen mit dem papierbasierten als auch mit einem elektronischen System umsetzen: Wenn die Studierenden am Vorlesungsende den Vorlesungsschein abholen, bittet man sie *persönlich*, den Fragebogen auszufüllen. In diesem Szenario würde sich der Einsatz eines dedizierten Abstimmungsterminals trotz der in Abschnitt 3.1 dargelegten Nachteile anbieten. Allerdings hängt die Umsetzbarkeit dieser Möglichkeit stark von den organisatorischen Begebenheiten der Scheinabgabe am jeweiligen Lehrstuhl ab.

2.3.13 Organisatorische Aspekte

Während die Handhabung des papierbasierten Umfragesystems jedermann offenbar, ist erfordert der Einsatz eines elektronischen Systems doch einen gewissen Mindestaufwand an technischer Einrichtung.

Da die dafür notwendigen Ressourcen oder das erforderliche Know-How nicht an allen Lehrstühlen vorhanden sind (insbesondere außerhalb der Fakultät für Informatik), erscheint der Wunsch einiger Dozenten naheliegend, selbst bei Einführung eines universitätsweiten elektronischen Umfragesystems die eigene Umfrage papierbasiert durchzuführen, da man so für die Auswertung nicht auf externe Ressourcen angewiesen ist.

Wir wollen daher die Möglichkeit, Papierfragebögen aus dem System zu erstellen, bewusst in unseren Entwurf miteinbeziehen.

2.3.14 Rechtliche Fragen

Die existierenden Fragebögen beginnen mit einem Block an Fragen zur Person. Diese persönlichen Daten werden somit bei papierbasierten Umfragen erfasst und wie in 2.2 dargelegt nur aggregiert weitergeleitet. Die individuellen Fragebögen werden idealerweise vernichtet, doch selbst wenn diese archiviert werden, ist der Personenkreis, der Zugang zu den Fragebögen findet, als relativ klein anzusehen.

Im Gegensatz dazu kann sich bei einem elektronischen System der Personenkreis, der Zugang zu den gesammelten Daten hat, stark erweitern:

- Der Systemadministrator gehört nicht unbedingt zum mit der Umfrage betrauten Personenkreis, hat aber dennoch Zugang zu den gesammelten Daten.

- Durch Bedienfehler, wie z.B. nicht ordnungsgemäß abgemeldete Zugänge zum Auswertungssystem, können persönliche Daten nach außen dringen.
- Eindringlinge, die sich aufgrund von Sicherheitslücken der installierten Software widerrechtlich Zugang zum Rechner verschaffen könnten, können die gespeicherten Daten nicht nur einsehen, sondern sogar manipulieren.

Auch die Menge der erfassten personenbezogenen Daten kann grösser sein, als beim papierbasierten Umfragesystem. So ist eine Speicherung der IP-Adresse des Webbrowsers denkbar, durch die unter Umständen Rückschlüsse auf die Identität der Abstimmenden Person gezogen werden können.

2.4 Erkenntnisse aus dem Vergleich verschiedener Umfragen

Wir vergleichen nun existierende Umfragen in elektronischer und nicht-elektronischer Form. Die Vielfalt an Unterschieden in kleinen Details stellt uns vor ein ungelöstes Problem, wenn wir ähnliche Fragen als eine Frage auswerten wollen. Zuletzt geben wir einen Ratschlag, wie dieses Problem umgangen werden kann.

2.4.1 Vergleichende Betrachtung verschiedener Umfragen

Unsere vergleichende Betrachtung von Umfragesystemen beschränkt sich nicht nur auf Vorleistungsbewertungssysteme, sondern bezieht auch allgemeinere Umfragen mit ein, wie z.B. eine Umfrage der Städtischen Münchner Bäder zur Kundenzufriedenheit, oder die Umfrage *Perspektive Deutschland* (nach eigenen Angaben die “weltweit größte gesellschaftspolitische Online-Umfrage”).

Münchner Bäder

Der in Abbildung 2.3 dargestellte Fragebogen verwendet eine Matrix aus fünfstufigen Ratingskalen und ein Freitextfeld für Anregungen. Die “Fragen” der Ratingskalen sind nur stichpunktartig formuliert. Alle Ratingstufen sind durch beschreibende Texte gekennzeichnet. Es werden keine Zahlen zur Beschriftung der Ratingskalen eingesetzt, so daß diese Messung gemäß Abschnitt 2.1.3 auf *Ordinalniveau* stattfindet.

Perspektive Deutschland

Die in Abbildung 2.4 dargestellte Matrixfrage ist Teil einer sehr umfangreichen Online-Umfrage. Die Fragen sind als Aussagen formuliert, denen auf einer sechsstufigen Ratingskala zugestimmt

werden kann. Lediglich die Extrempunkte der Ratingskala sind mit Textbeschreibungen versehen. Alle Stufen sind zusätzlich mit Zahlen gekennzeichnet, so daß hier der Eindruck entstehen kann, die Messung fände auf *Intervallniveau* statt. Aber ist "5" genausoviel besser als "6", wie "1" im Vergleich mit "2"?

Vorlesungsumfrage Physik

Diese in Abbildung 2.5 gezeigte Vorlesungsumfrage findet sowohl in Papierform als auch in elektronischer Form statt. Die Gestaltung der beiden Fragebögen ist sehr ähnlich. Es kommt eine fünfstufige Ratingskala zum Einsatz, die nur an ihren Extrempunkten durch Textbausteine verankert ist. Da die Fragen als echte Fragen ausformuliert sind, gibt es Unterschiede in der textuellen Beschreibung der Antworten an den Extrempunkten der Ratingskalen, so daß keine Matrixdarstellung gewählt werden kann.

Vorlesungsumfrage SIGNAL

Abbildung 5.2 zeigt einen Fragebogen mit einer Matrix aus sechsstufigen Ratingskalen. Teilweise sind die Fragen als Aussagen formuliert, teilweise als Fragen, durch die die Testperson geduzt wird. Die Bedeutung der nur mit Ziffern beschrifteten Antwortmöglichkeiten sind im Kopf des Fragebogens dargelegt.

Fazit der vergleichenden Betrachtung

Art und Formulierung der Fragen unterscheidet sich bei allen vier betrachteten Fragebögen drastisch. Wenn wir kurz annehmen, daß derartig unterschiedliche Fragebögen von verschiedenen Assistenten eines Lehrstuhls erstellt worden sind und man selbst nun einen neuen Fragebogen konstruieren muß, der Teilbereiche zweier unterschiedlicher Fragebögen erfasst, so hätte man seine liebe Not aus der bestehenden Fragenbasis einen neuen Fragebogen zu konstruieren.

Entweder man erzeugt die Hälfte der Fragen in einem angepassten Fragestil neu, oder man beschränkt sich auf die bereits existierenden Fragen und erhält einen Fragebogen der eher uneinheitlich wirkt.

Letzere Lösung mag zwar ihre Vorzüge haben, da der uneinheitliche Fragestil für Abwechslung sorgt und damit die Qualität der Umfrage anheben kann (Nach [Noelle-Naumann 96] können absichtliche Themensprünge oder andere Stilbrüche durch ein "Wachhalten" des Teilnehmers qualitätssteigernd wirken). Allerdings kann eine verwendbare Darstellung eines derartig zusammengewürfelten Fragebogens zu einer Herausforderung ausarten, wenn wir uns an die in Abschnitt 2.3.2 getroffenen Aussagen über verschiedene Präsentationsformen von Ratingskalen erinnern.

Ein stilistisch konsistent entworfener Fragebogen führt dagegen zu redundanten Fragen im elektronischen System, deren weitere Implikation wir nun im folgenden Abschnitt diskutieren.



Ihre Meinung ist uns wichtig

Wir freuen uns, wenn Sie diesen Fragebogen ausfüllen.
Herzlichen Dank!

	sehr zufrieden	eher zufrieden	teils-teils	eher unzufrieden	sehr unzufrieden
Eingang/Kasse					
■ Freundlichkeit der Mitarbeiter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Sauberkeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Badeartikel	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Umkleiden/Duschen/Toiletten					
■ Sauberkeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Schwimmhalle					
■ Kompetenz der Mitarbeiter/innen (Wassergymnastik, Schwimmunterricht usw.)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Freundlichkeit der Mitarbeiter/innen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Sauberkeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sauna					
■ Fachliche Kompetenz der Mitarbeiter (Aufguss)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
■ Sauberkeit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

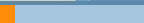
Haben Sie Anregungen?

Geben Sie uns Ihre Daten?

_____ Name	_____ Vorname
_____ Straße/Hausnummer	_____ Telefon
_____ Ort/Postleitzahl	_____ E-Mail

Abbildung 2.3: Umfragebeispiel: Münchner Bäder

perspektive deutschland eine Initiative von McKinsey, stern, ZDF und WEB.DE

Fortschritt: 

Jetzt geht es um einige grundsätzliche Aussagen zu Lebensauffassungen und -ansichten.
Bitte geben Sie an, wieweit Sie den folgenden Aussagen zustimmen.

	Stimmt genau 1	2	3	4	5	Stimmt gar nicht 6	Weiß nicht
Ich setze meine Ziele und Erfolgsvorstellungen in die Tat um.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das Leben zu genießen finde ich wichtiger, als Erfolg im Beruf zu haben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich fühle mich für die Gesellschaft verantwortlich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Auch wenn ich in der Klemme stecke, finde ich immer eine Lösung.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich nutze gerne neue Technik und bin immer auf dem neuesten Stand.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich nehme das Leben lieber, wie es kommt, als alles genau zu planen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bei technischen Problemen schreiben Sie uns bitte eine E-Mail an technik@perspektive-deutschland.de

Abbildung 2.4: Umfragebeispiel: Perspektive Deutschland

2.4.2 Zusammenführende Behandlung gleichartiger Fragen

Idealerweise sollte es im gesamten System nur eine Fassung einer jeden Frage geben, die in allen Fragebögen verwendet wird, in denen die Frage erwünscht ist. Leider gibt es eine Menge von Gründen, die dazu führen, daß zu ein und derselben *beabsichtigten* Fragestellung verschiedene *Formulierungen* existieren:

- Es werden zwei bereits existierende Fragebögen zu einer übergreifenden Auswertung zusammengeführt

Beispiel:

Das Tafelbild war lesbar Ja **Nein**

Konnten Sie das Tafelbild gut lesen? Sehr gut **Überhaupt nicht**

- Versehentlich wurde eine Frage mit identischem oder unterschiedlichem Wortlaut mehrmals eingepflegt

Beispiel:

Das Tafelbild war lesbar Ja **Nein**

Das Tafelbild war leserlich Ja **Nein**

- Dieselbe Frageformulierung wurde mit verschiedenstufigen Abstimmungsskalen angebo-

Fassung SS 2003

Vorlesungsumfrage der Fakultät für Physik

Ziel dieser Umfrage ist es, die Qualität der Lehre an der Fakultät für Physik zu optimieren. Bitte füllen Sie daher diesen Fragebogen fair und differenziert aus.

Dozent: _____ **WS2003/2004**

Vorlesung: _____

1. Vorlesung:

- 1.1 Wie beurteilen Sie den Umfang des dargebotenen Stoffes?
(bei Kursvorlesungen im Vergleich zum Stoffplan)
genau richtig

--	--	--	--	--

 zu viel
zu wenig
- 1.2 War Ihnen der detaillierte Stoffplan im Voraus bekannt?
ja

--	--

 nein
- 1.3 Wie war die didaktische Darstellung des Stoffes?
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.4 Wie beurteilen Sie den Vortrag?
Differenzieren Sie nach mündl. Darstellung = M, schriftl. Darstellung = S, Tempo = T (Eintrag des Buchstabens in gewählte Box)
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.5 Wurden ausreichende Hilfen zur Vorlesung angeboten?
(Eingehen auf Fragen, Literatur, evtl. Skript, Sprechstunden)
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.6 Wurde Material zur Vorlesung im Internet angeboten?
ja

--	--

 nein
Qualität ?
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.7 Wie beurteilen Sie das Engagement des Dozenten?
(Begeisterungsfähigkeit = B, Vorbereitung = V, Anwesenheit = A)
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.8 Wie war Ihr Eindruck von der Vorlesung insgesamt?
(Würden Sie z.B. die Vorlesung an andere Studenten weiterempfehlen?)
hervorragend

--	--	--	--	--

 sehr schlecht
- 1.9 Würden Sie den Dozenten einem jüngeren Studenten weiterempfehlen?
auf jeden Fall

--	--	--	--	--

 auf keinen Fall

2. Übungen (falls angeboten):

- Name des Übungsgruppenleiters: _____
- 2.1 Wie war die Koordination zwischen Vorlesung und Übung?
hervorragend

--	--	--	--	--

 sehr schlecht
- 2.2 Wie beurteilen Sie den Zeitaufwand für die Bearbeitung eines typischen Übungsblattes?
(falls Hausaufgaben vorgesehen waren)
angemessen

--	--	--	--	--

 zu hoch
zu niedrig
- 2.3 Wie beurteilen Sie die didaktischen Fähigkeiten Ihres Übungsbetreuers?
hervorragend

--	--	--	--	--

 sehr schlecht
- 2.4 Wie war der Betreuer auf die Übungen vorbereitet?
hervorragend

--	--	--	--	--

 sehr schlecht
- 2.5 Wie war Ihr Eindruck von der Übung insgesamt?
(Würden Sie z.B. die Übung an andere Studenten weiterempfehlen?)
hervorragend

--	--	--	--	--

 sehr schlecht

Abbildung 2.5: Umfragebeispiel: Fakultät für Physik

ten

Beispiel:						
Das Tafelbild war lesbar	Ja	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Nein
Das Tafelbild war lesbar	Ja	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> Nein

Insbesondere im Falle verschiedenerstufiger Skalen ist nicht klar, wie eine solche Abbildung zu erfolgen hat. Aber auch bei gleicher Skalenstruktur kann eine Abbildung aufgrund leichter semantischer Unterschiede in der Frageformulierung eine Anpassung notwendig machen, wenn sich in einer vergleichenden Testreihe herausstellen sollte, daß die eine Frageformulierung tendenziell positiver beantwortet wird als die andere. Der einfachste Fall läge vor, wenn die beiden Formulierungen sich wie Satz und Kehrsatz verhalten, da in diesem Fall nur die Skala umgedreht werden müsste. Doch selbst in diesem scheinbar trivialen Fall könnte der in der Einleitung von [Kirchhoff 00] beschriebene Effekt, daß Menschen tendenziell lieber “Ja” sagen als “Nein”, zu Problemen bei der Zusammenführung der beiden Antwortskalen führen.

In Abbildung 2.4.2 vergleichen wir eine vier-, fünf- und sechsstufige Ratingskala und bilden die möglichen Antworten auf Werte im Intervall von 0 bis 1 ab. Dabei stehen wir vor dem Problem, ob eine gegebene Antwort w wirklich ein schlechteres Urteil darstellt als eine Antwort E , wenn in beiden Fällen ausgedrückt werden sollte, daß man einer gegebenen These *fast* ganz zustimmt. Lediglich die Extremstimmen lassen sich ohne große Probleme aufeinander abbilden.

Die Ursache des Problems liegt darin, daß Ratingskalen im Sinne der in Abschnitt 2.1.3 definierten *Skalenniveaus* nur auf einer *Ordinalskala* messen, und somit nur Vergleichsoperationen innerhalb der Ratingskala definiert sind, wohingegen eine Messung auf *Rationiveau* einen Skalenübergreifenden Vergleich gestatten würde. Leider wird das hier beschriebene Problem der Zusammenführung von Umfragen in der von uns gesichteten Literatur zur empirischen Sozialforschung nicht behandelt.

Im Rahmen unserer Testläufe verwendeten wir zwei verschiedene Fragebögen: zum einen den Standardfragebogen der Fakultät für Mathematik und Informatik und zum anderen die getrennt entstandenen Fragebögen für die Veranstaltungen des SIGNAL-Projekts. Da SIGNAL jedoch im Jahr 2006 enden wird¹ lohnt weder eine Lösung des Mapping-Problems noch eine Vereinheitlichung der Fragebögen für künftige Umfragen.

Da die sonstigen Vorlesungsumfragen im wesentlichen mit *einem* Fragebogen durchgeführt werden, ist davon auszugehen, daß dieser als Basis für individuelle Anpassungen und Verfeinerungen dienen wird. Deshalb messen wir der Behandlung dieses Problems nur eine untergeordnete Wichtigkeit bei, und verzichten aufgrund der geringen Relevanz und der großen Unklarheiten über eine korrekte Vorgehensweise auf eine Abbildung verschiedener Fragen aufeinander.

Man könnte nun anführen, daß das eigentliche Problem im Anbieten einer diskreten Skala besteht und es doch besser wäre, den Grad der Zustimmung auf einer (quasi)-kontinuierlichen Skala anzugeben. Doch treten dann die in Abschnitt 2.1.4 erläuterten Probleme einer zu feinstufigen

¹<http://www.ifi.lmu.de/Studium/Studiengaenge/SIGNAL/Ueberblick/index.html>

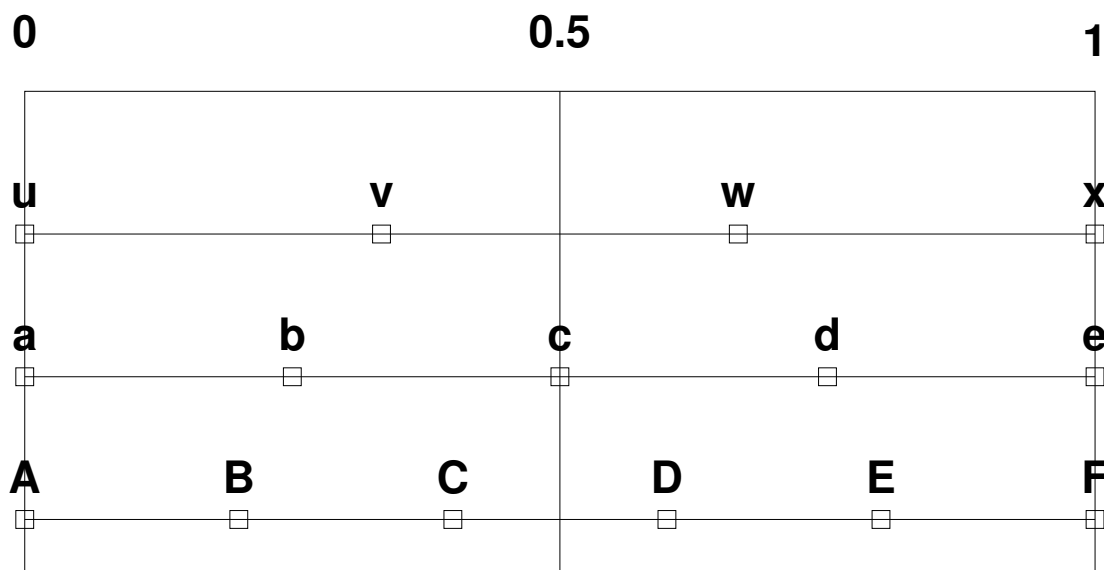


Abbildung 2.6: Mapping-Problem unterschiedlich dimensionierter Skalen

Skala verstärkt auf: Verschiedene Probanden legen ihrer Entscheidung für eine Antwort verschiedene Interpretationen der feinstufigen Skala zu Grunde, so daß beispielsweise ein Proband mit 60% Zustimmung meint ein gutes Urteil abgegeben zu haben, wohingegen ein anderer 65% als eher mittelmäßig ansehen könnte.

2.4.3 Richtlinien für eine einheitliche Formulierungsstruktur

Wir empfehlen vor Einsatz eines elektronischen Vorlesungsbewertungssystems eine Richtlinie zur Formulierung von Fragen und Antworten zu definieren, die insbesondere folgende Punkte klärt:

- Sollen die Testpersonen mit “Du” oder “Sie” angesprochen werden?
- Werden Fragen formuliert, die beantwortet werden müssen, oder Aussagen vorgegeben, denen man zustimmen oder widersprechen kann?
- Werden sich ähnelnde Fragen jedesmal vollständig neu ausformuliert? Oder werden derartige Fragen mit “...” fortgeführt?
- Wieviele Stufen sollen die Ratingskalen aufweisen?
- Wird eine Präsentationsform gewählt, bei der Texte für Zwischenstufen der Ratingskalen eher störend wirken oder eher notwendig sind?

Eine einheitliche Konzeption des Fragebogaufbaus reduziert die in Abschnitt 2.4.2 umrissene Gefahr von Parallelentwicklungen und steigert in Abschnitt 2.1.4 definierte Qualität der Abstimmungsergebnisse.

Kapitel 3

Entwurf

Nach unserer eingehenden Problemanalyse befassen wir uns in diesem Kapitel mit dem Entwurf eines elektronischen Vorlesungsumfragesystems. Hierzu untersuchen wir verschiedene technische Ansätze und treffen die Entscheidung für ein Web-basiertes Umfragesystem. Die Vielschichtigkeit der im vorangehenden Kapitel beleuchteten Probleme erfordert den Entwurf des Systems als *modulares Framework*, dessen Konzeption hier detailliert erklärt wird. Der Vorstellung des für das System entworfenen Datenbankschemas wird abgerundet von einer Einführung in das Konzept unserer *Pseudofragen*, die automatisch von Systemkomponenten “beantwortet” werden.

3.1 Denkbare elektronische Realisierungen

Eine Vielzahl verschiedenartiger elektronischer Umfragesysteme sind denkbar, von denen wir vier Varianten auf ihre Verwendbarkeit in unserem Umfeld untersuchen:

Speziell aufgestellte Abstimmungsterminals Dedizierte Abstimmungsmaschinen benötigen keine Verbindung zu Datennetzen, da die Umfrageteilnehmer direkt an der Maschine selbst die Bewertung abgeben. Daher bietet diese Lösung theoretisch eine sehr hohe Sicherheit gegenüber Angriffen auf das System mit dem Ziel Daten auszuspähen oder zu manipulieren. Allerdings ist die Erfüllung eines derart hohen Sicherheitsniveaus für unseren Einsatzzweck nicht wirklich notwendig. Der hohe Beschaffungs- und Einrichtungsaufwand von Computern extra für die Durchführung von Vorlesungsumfragen lässt uns diese Lösung als äußerst ungeeignet erscheinen.

Programm auf dem Computer des Abstimmenden Diese Lösung erfordert, daß der Abstimmende das Umfrageprogramm auf seinem eigenen Computer ausführt. Man kann annehmen, daß diesem Verfahren eine gewisse Skepsis seitens der Abstimmenden entgegengebracht wird, da sie sich nicht sicher sein können, was das Programm an unerwünschten Effekten mit sich bringt. Die Erstellung eines plattformunabhängigen Programms ist

durch Verwendung verbreiteter Skriptsprachen oder dem Einsatz einer virtuellen Maschinen Plattform wie z.B. Java möglich, erfordert aber auf Seiten des Anwenders die Installation der entsprechenden Laufzeitumgebung, deren Vorhandensein nicht zwingend vorausgesetzt werden kann. Der mit einer Software-Installation verbundene Aufwand wirkt sich wohl zusätzlich motivationshemmend aus und würde die in Kapitel 2.3.12 erwähnten ohnehin bereits niedrigen Rücklaufquoten weiter schmälern.

Über E-Mail versendete Fragebogen-Formulare Etliche im Rahmen dieser Arbeit entworfene Konzepte, lassen sich auf dem Weg, Fragebögen per E-Mail zu versenden, nicht oder nur schwer umsetzen. Die Anonymität kann von Seiten des Abstimmenden angezweifelt werden, da er den Fragebogen an seine E-Mail-Adresse persönlich zugestellt bekommen hat und beispielsweise durch leicht unterschiedliche, individuell generierte Fragebögen identifizierbar würde. Wenn die Abgabe des ausgefüllten Fragebogen auch per E-Mail erfolgt, ist eine Identifikation anhand der Absenderadresse möglich, so daß berechtigte Zweifel an der Anonymität der Umfrage aufkommen könnten. Desweiteren könnte die abstimmende Person den Fragebogen verändern, indem sie beispielsweise eine ihr unklar erscheinende Frageformulierung abändert. Diese in guter Absicht vorgenommene Änderung bewirkt jedoch, daß eine automatische Auswertung dieses veränderten Fragebogens unmöglich wird. Daher ziehen wir eine E-Mail-basierte Lösung nicht weiter in Betracht, da sie keine nennenswerten Vorteile aufweist, die die erwähnten Nachteile auch nur annähernd kompensieren würden.

Web-basiertes Umfrageformular Eine Web-Lösung erreicht durch die starke Verbreitung von Webbrowsern (z.B. sogar auf mobilen Endgeräten) die größte Abdeckung an möglichen Kandidaten bei gleichzeitig minimalem Aufwand für die Kandidaten um an der Abstimmung teilzunehmen (keine Softwareinstallation nötig, außer dem in der Regel bereits vorhandenen Webbrowser). Web-Anwendungen sind eingeschränkter in ihren Möglichkeiten der Benutzerführung, aber für das Ausfüllen eines Fragebogens dennoch ausreichend.

Aus den hier dargelegten Gründen der Einfachheit für den Abstimmenden entscheiden wir uns für eine Web-basierte Lösung, da hierdurch ein guter Kompromiss zwischen Flexibilität des Systems und Komfort für den Abstimmenden erreicht werden kann.

Da mit der Pflege der Fragebögen und dem Management der Systemkonfiguration nur ein kleiner Personenkreis betraut sein wird, greifen die angeführten Argumente hierfür weniger. Für diese Aufgaben ist eine Bedienung wünschenswert, die mehr Komfort bietet, als es durch ein Web-Interface möglich ist. Deshalb werden wir hierfür eine separate grafische Benutzerschnittstelle programmieren. In Abbildung 3.1 veranschaulichen wir, wie unser Entwurf des Gesamtsystems die folgenden fünf Aufgaben erfüllt, die in Abschnitt 2.2 beschrieben wurden:

1. *Fragebogenerstellung und Konfigurationsmanagement* über eine GUI-Anwendung
2. Ausgabe und Annahme von *Fragebögen* über eine offene Web-Schnittstelle
3. Erstellung einer veröffentlichen *Auswertung* über eine offene Web-Schnittstelle
4. Erstellung eines internen *Reports* über eine geschützte Web-Schnittstelle

5. Anzeige einzelner ausgefüllter *Antwortbögen* über eine geschützte Web-Schnittstelle

Die nur für einen kleinen Personenkreis gedachten Report- und Antwortbogen-Ansichten werden wir aufgrund der strukturellen Verwandtschaft mit der Auswertungs- und Fragebogenansicht ebenfalls über die Web-Schnittstelle realisieren, auch wenn hierfür der Weg über eine GUI-Schnittstelle denkbar wäre.

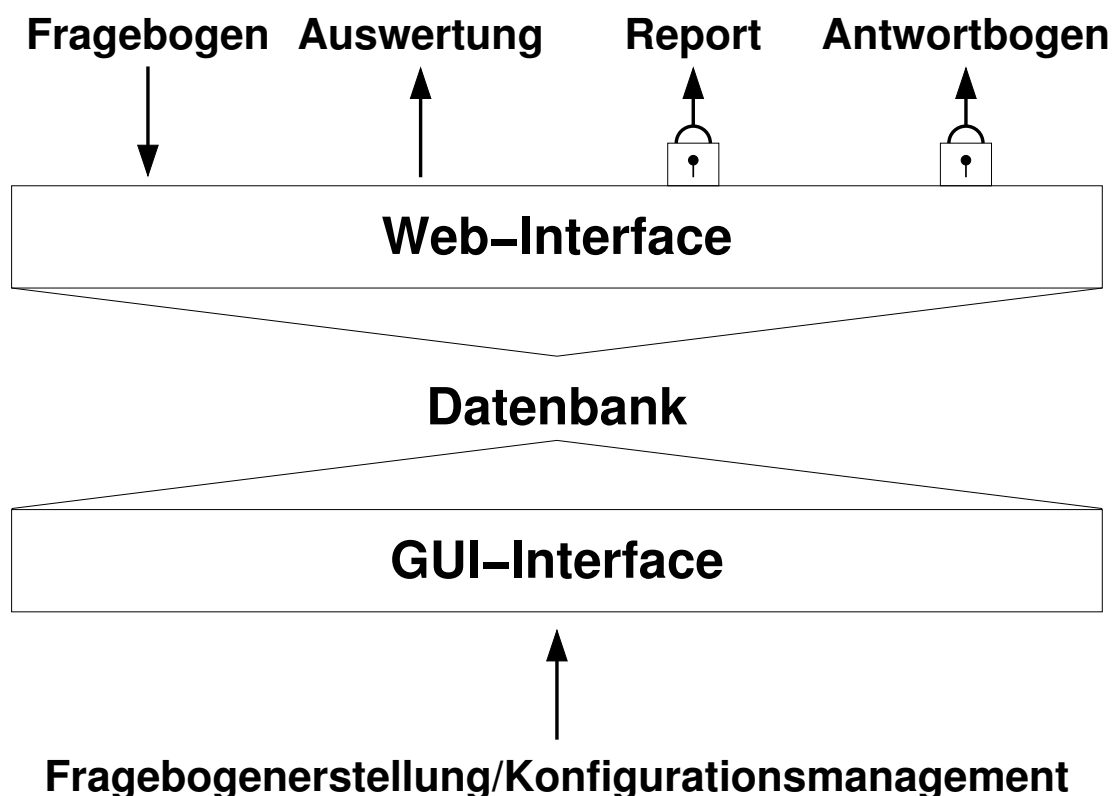


Abbildung 3.1: Übersicht der fünf System-Aufgaben

3.2 Modulares Framework

Durch die unterschiedlichen Möglichkeiten zur Lösung einiger Probleme, vor allem der in Abschnitt 2.3.10 umrissenen Zugangskontrolle, ist eine einheitliche Lösung nicht möglich. Daher entwerfen wir ein modulares Framework, das durch Erstellung entsprechender Module eine Vielzahl möglicher Vorgehensweisen gestatten wird.

Ein grob ähnliches Problem findet sich bei der Benutzeranmeldung auf Unix-Systemen. Die Liste der zur Anmeldung berechtigten Benutzer und ihrer Passwörter können in vielfältiger Form vorliegen: als Datei auf dem lokalen Rechner, in einer Datenbank oder einem Verzeichnisdienst oder sogar in einer Kombination verschiedener Möglichkeiten. Diese Vielzahl möglicher Implementierungen gepaart mit einer Vielzahl an Diensten, die eine Benutzeranmeldung erfordern

(z.B. Anmeldung an der lokalen Konsole, einloggen über das Netzwerk mit `ssh`, Abrufen von E-Mails), legt die Entwicklung einer Schnittstelle nahe, die von Sun Microsystems als Pluggable Authentication Modules (kurz: PAM) etabliert wurde. Diese Schnittstelle kommt z.B. unter Linux, Solaris und anderen Unix-Derivaten zum Einsatz.

Unser System wurde vor allem durch das Konzept der Verkettung verschiedener Module inspiriert, das den Kern der PAM-Architektur darstellt. Im Unterschied zur PAM-Architektur werden wir unsere Modulketten jedoch Stack-artig zweimal durchlaufen, da wir an einigen Stellen einen Vorbereitungsschritt und einen Nachbereitungsschritt benötigen.

Jeder Schritt kann eines von drei möglichen Ergebnissen liefern: `True` für eine erfolgreiche Durchführung dieses Schrittes, `False` für ein Scheitern des Schrittes und `Undef` für ein unklares, noch nicht bekanntes Ergebnis, das u.U. bei einer erneuten Auswertung geklärt werden kann.

Fortan bezeichnen wir die Module unseres Frameworks, die einen Vor- und einen Nachbereitungsschritt ausführen können, und den Erfolg bzw. Misserfolg in der eben beschriebenen Weise melden, als *Handler*.

Wir versehen jeden Handler mit einem Flag, das ihn entweder als “required” oder als “sufficient” gekennzeichnet. Dieses Flag entscheidet über das Vorgehen in den Vorbereitungsschritten.

Wenn der Vorbereitungsschritt eines required-Handler fehlschlägt, wird die gesamte Abarbeitung abgebrochen. Wenn der Vorbereitungsschritt eines required-Handler ein unklares Ergebnis liefert, wird die Verarbeitung unterbrochen und beim nächsten Versuch vom Einsprungpunkt an wieder aufgenommen. Nur wenn der Vorbereitungsschritt eines required-Handler ein erfolgreiches Ergebnis liefert, wird die Verarbeitung mit dem nächsten Handler fortgesetzt und der Einsprungpunkt auf den nächsten Handler gesetzt.

Ist der Handler dagegen als sufficient konfiguriert, so zeigt er folgendes Verhalten: Wenn der Vorbereitungsschritt eines sufficient-Handler fehlschlägt oder ein unklares Ergebnis liefert, wird die Verarbeitung mit dem nächsten Handler fortgesetzt. Wenn der Vorbereitungsschritt eines sufficient-Handler allerdings ein erfolgreiches Ergebnis liefert, wird die Verarbeitung der Vorbereitungsschritte beendet und nur der Stack der bisher verwendeten Handler in umgekehrter Reihenfolge zur Ausführung der Nachbereitungsschritte abgearbeitet.

Bei den Nachbereitungsschritten wird wie folgt vorgegangen:

Ein Scheitern führt zum Abbruch der Abarbeitung. Ein unklares Ergebnis führt zur Unterbrechung der Abarbeitung und erfordert eine erneute Auswertung des Nachbereitungsschritts des aktuellen Handlers. Ein erfolgreiches Ergebnis bewirkt die Fortsetzung der Abarbeitung mit dem nächsten Handler in Nachbereitungsreihenfolge, also dem in der Konfiguration vorhergehenden Handler.

Wenn nach dem Abarbeiten aller Vorbereitungsschritte kein erfolgreicher Übergang zu den Nachbereitungsschritten stattgefunden hat, so wird dies als Erfolg der Vorbereitungsschritte gewertet und es wird zur Nachbereitung übergegangen. Man kann sich einen impliziten sufficient-Handler

an der Spitze des Stacks denken, der immer `True` liefert und somit zur Abarbeitung der Nachbereitungsschritte überleitet.

Zur Kommunikation zwischen der Anwendung und der verschiedenen Handler wird eine *Transferstruktur* an die Vor- und Nachbereitungsschritte übergeben, die neben einer Framework-internen Repräsentation eines Fragebogens auch die aktuellen Handlerkonfiguration enthält.

Am Beispiel der in Abbildung 3.3 dargestellten Handler-Kombination stellen wir das Verhalten einiger möglichen Handler dar. Der Zweck dieser speziellen Kombination besteht darin, die Umfrage durch ein Zugangspasswort zu schützen, die Rücklaufquote durch ein Bonbon zu verbessern und den Zugang zum Bonbon durch ein weiteres Passwort mehrmals zu gestatten, ohne daß dazu eine mehrfache Abgabe des Fragebogens erforderlich ist. Der nochmalige Zugang zum Bonbon wird durch den `sufficient` Passworhandler E erreicht. In Abbildung 3.2 ist ein möglicher Ablauf im Rahmen eines Web-basierten Systems als UML-Sequenzdiagramm dargestellt.

Listing 3.1: Beispielkonfiguration des Frameworks

```
<?xml version="1.0"?>
<chain>

<handler flag="required"
class="de.itfuzzi.diplom.core.handler.DatenbankHandler" />

<handler flag="required"
class="de.itfuzzi.diplom.web.handler.Password"
sheet="xml/xslt/zugang.xsl"
prompt="Bitte_geben_sie_das_Zugangspasswort_ein:"
password="g3h3im"
/>

<handler flag="required"
class="de.itfuzzi.diplom.core.handler.MinimalzeitHandler"
minimumsec="120"
/>

<handler flag="required"
class="de.itfuzzi.diplom.web.handler.BonbonHandler"
sheet="xml/xslt/belohnung.xsl"
text="Danke!_Der_Geheimcode_lautet_1234_und_hier_ist_das_Geschenk:"
url="http://www.google.de"
/>

<handler flag="sufficient"
class="de.itfuzzi.diplom.web.handler.Password"
sheet="xml/xslt/fragebogen.xsl"
prompt="Sie_haben_bereits_teilgenommen?"
```

```

<<<<<<<<Mit_dem_Geheimcode_können_sie_gleich_zur_Belohnung:"
password="1234"
/>

<handler flag="required"
class="de.itfuzzi.diplom.web.handler.WebPresentationHandler"
sheet="xml/xslt/fragebogen.xsl"
/>

</chain>

```

Betrachten wir nun die einzelnen Schritte und ihre Auswirkungen:

Vorbereitungsschritte:

Schritt 1 Handler A Der Datenbankhandler liest die Fragebogenstruktur aus einer Datenbank ein und liefert `True`.

Schritt 2 Handler B Der Passworhandler liefert im ersten Anlauf ein `Undef` und vermerkt den Wunsch eine Eingabeaufforderung für ein Zugangspasswort zu erzeugen. Da dieser Handler `required` ist wird die Abarbeitung an dieser Stelle zunächst unterbrochen und die Anwendung setzt den Wunsch dieses Handlers nach einer Eingabeaufforderung entsprechend um (z.B. durch Darstellung eines Dialogfensters, ein Prompt auf einer Kommandozeile oder Ausliefern einer Webseite, je nach Art der Anwendung).

In einem neuen Anlauf (nach Eingabe des Passwortes) wird die Abarbeitung an dieser Stelle fortgesetzt. Wenn das Passwort korrekt ist, liefert der Passworhandler `True`.

Schritt 3 Handler C Der Auffüllzeithandler merkt sich die aktuelle Uhrzeit und liefert `True`

Schritt 4 Handler D Der Bonbonhandler tut nichts und liefert `True`

Schritt 5 Handler E Ein weiterer Passworhandler vermerkt ebenfalls seinen Wunsch nach einer Passwordeingabe und liefert im ersten Anlauf `Undef`. Da dieser Handler jedoch nicht `required`, sondern `sufficient` ist, kann die Abarbeitung fortgesetzt werden. Dabei wird der Einsprungpunkt zunächst *nicht* weitergeschaltet, so daß dieser Passworhandler nochmals aufgerufen wird.

Wenn dieser Handler die Eingabe eines korrekten Passwortes feststellt, liefert er `True` und da er `sufficient` ist bewirkt dies die Fortsetzung bei Schritt 8 und ein Überspringen der Schritte 6 und 7 des nächsten Handlers.

Schritt 6 Handler F Der Darstellungshandler stellt z.B. die erste Frage des Fragebogens dar und liefert solange `Undef` bis eine passende Antwort eingegeben wurde (z.B. eine Möglichkeit angekreuzt wurde oder auf die Frage nach einer Zahl auch wirklich eine Zahl eingegeben wurde). Sodann fährt er mit der Darstellung der nächsten Frage fort. Solange dieser Handler `Undef` liefert verbleibt der Einsprungpunkt jedoch bei Handler E, so daß dieser Handler jederzeit übersprungen werden kann.

Wenn alle Fragen beantwortet sind liefert er schließlich `True`, und da keine weiteren Handler folgen wird mit den Nachbereitungsschritten fortgefahren.

Nachbereitungsschritte:

Schritt 7 Handler F Der Darstellungshandler vermerkt die gegebenen Antworten in der Transferstruktur und liefert `True`.

Schritt 8 Handler E Der Passworhandler tut nichts und liefert `True`.

Schritt 9 Handler D Der Bonbonhandler vermerkt eine Belohnung für die Teilnahme an der Umfrage und verrät das Passwort für den zweiten Passworhandler, so daß die Belohnung auch zu einem späteren Zeitpunkt nochmal abgeholt werden kann, ohne daß der Fragebogen ein zweites Mal ausgefüllt werden muss.

Schritt 10 Handler C Der Ausfülldauerhandler vergleicht die aktuelle Uhrzeit mit der gemerkten Startzeit. Wenn die zum Ausfüllen benötigte Zeit kürzer ist als die Mindestzeit, die für ein ehrliches Ausfüllen des Fragebogens nötig ist liefert dieser Handler `False`, wodurch die Abarbeitung beendet wird und diese Stimmabgabe nicht mehr berücksichtigt wird da sie nicht mehr zu Schritt 12 gelangen wird. Wenn die Ausfülldauer plausibel erscheint liefert der handler `True`. Da die plausible Mindestausfülldauer vom Umfang des jeweiligen Fragebogens abhängt, ist diese nicht im Handlercode fest verdrahtet, sondern wird aus der für jeden Fragebogen individuell gestaltbaren Konfiguration der Handlerchain bezogen.

Schritt 11 Handler B Der Passworhandler tut nichts und liefert `True`.

Schritt 12 Handler A Der Datenbankhandler trägt die in Schritt 7 vermerkten Antworten in die Datenbank ein und liefert `True`.

Der Kern des Frameworks selbst ist nur für die Ansteuerung der Handler zuständig und somit nicht an unsere Entscheidung für eine Web-basierte Lösung gebunden. Ein Einsatz des Frameworks für andere Realisierungen würde nur die Erstellung geeigneter Handler erfordern, da jegliche Interaktion mit dem Benutzer und anderen Systemkomponenten, wie z.B. Datenbanken, wie in 3.3 durch die Handlerinstanzen ausgeführt wird.

Unser Entwurf unterstützt sowohl eine imperative als auch eine ereignisgesteuerte Vorgehensweise. In einer imperativen Implementierung würde der Handler selbst die Benutzerinteraktion ausführen und erst wenn er sein Ergebnis kennt zurückkehren. Ein `Undef` Ergebnis würde nicht geliefert werden.

Die Entscheidung für eine Web-basierte Implementierung erfordert dagegen eine ereignisgesteuerte Programmierung. Die Handler nehmen dabei selbst keine Benutzerinteraktion vor, sondern vermerken in der Transferstruktur ihre Anforderungen (wie z.B. eine erforderliche Passwor eingabe), die von der Webanwendung durch die Generierung entsprechender HTML-Dokumente umgesetzt werden.

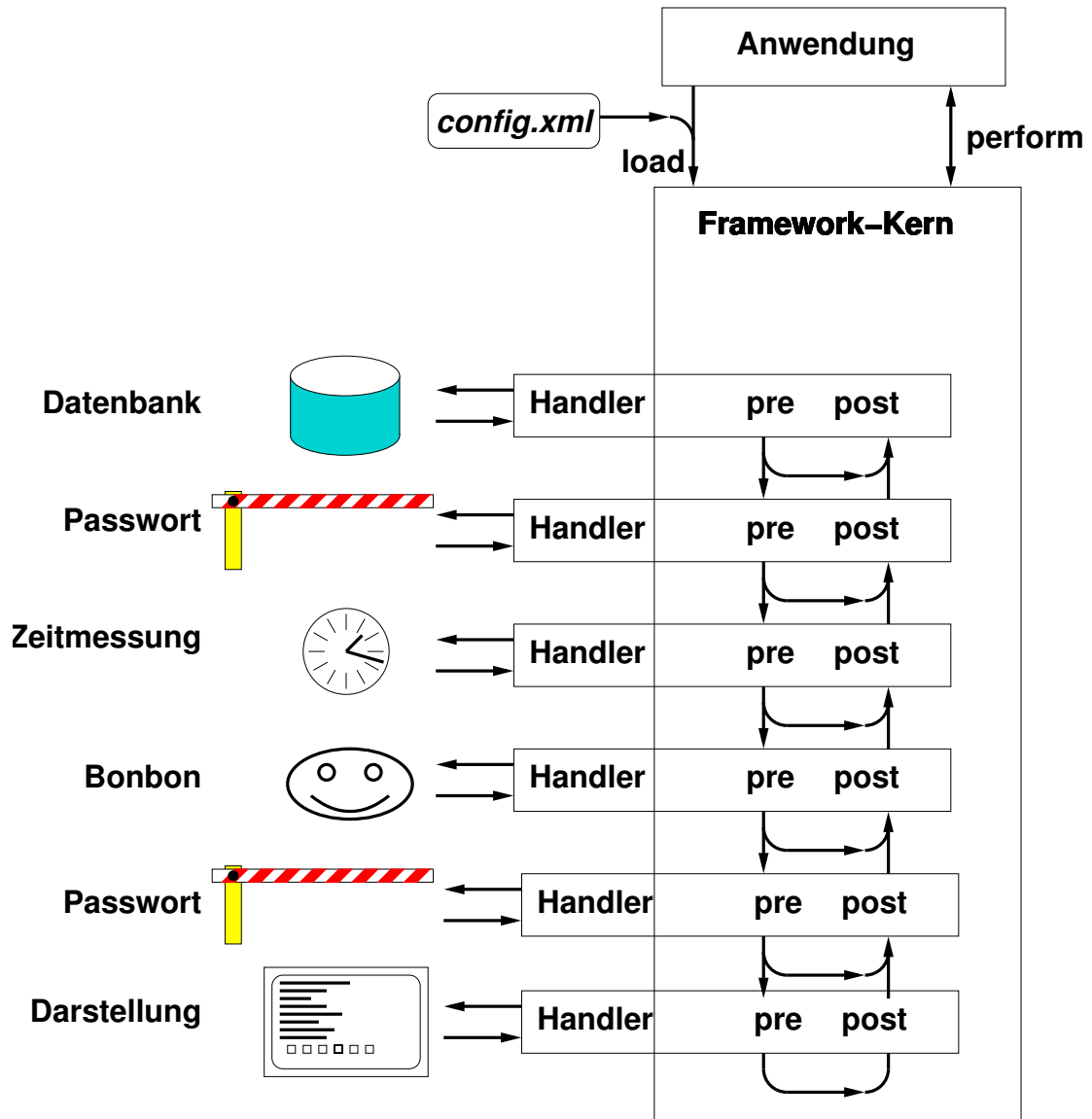


Abbildung 3.3: Framework-Entwurf mit Beispielhandlern

3.3 Datenbankschema

Unser Datenbank-Entwurf ist in Abbildung 3.4 als Entity-Relationship-Diagramm dargestellt. Die resultierende Tabellenstruktur zeigt der Screenshot in Abbildung 3.5. Wir vertiefen nun einige Details unseres Entwurfs.

3.3.1 Fragenkataloge

Um zur besseren Unterstützung der in Abschnitt 2.3.1 angeregten modularen Gestaltung der Fragebögen, sehen wir als Zwischenschicht zwischen den Einzelfragen und dem Fragebogen sogenannte *Fragenkataloge* vor. Diese dienen zur thematische Strukturierung des Fragebogens in Abschnitte wie beispielsweise “Übungsbetrieb”, “Persönliche Angaben” oder “Mathematische Begleitvorlesung”.

3.3.2 Internationalisierung

Im Hinblick auf eine leichtere Umsetzung der in Abschnitt 2.3.3 angeregten Mehrsprachigkeit des Vorlesungsumfragesystems speichern wir alle Texte, die übersetzt werden müssten, in eine Tabelle `i18ndata`. Dort haben wir zwei Spalten `textde` und `texten` vorgesehen für korrespondierende deutsche und englische Texte.

Um etwaige fehlende Übersetzungen insbesondere während der Entwicklungsphase zu überbrücken, wurde eine View `i18nview` definiert, die fehlende englische Texte durch den entsprechenden deutschen Text mit einem vorangestellten Präfix “EN:” ersetzt.

3.3.3 Ratingskalen

Die in Abschnitt 2.1.3 erläuterten *Ratingskalen* verhalten sich bei der Durchführung der Umfrage zunächst ähnlich wie Multiple-Choice-Fragen (vgl. die Beschreibung in Abschnitt 2.3.2). Bei der Auswertung erscheinen sie dahingegen eher analog zu Zahlfragen, da üblicherweise Mittelwerte und andere statistische Berechnungen mit Ratingskalen durchgeführt werden (zumindest wenn diese auf dem in 2.1.3 vorgestellten *Intervallniveau* messen, und nicht nur auf *Ordinalniveau*).

Dieser Zwittercharakter veranlasste uns, Ratingskalen in Bezug auf die Datenspeicherung als Multiple-Choice-Fragen in das Datenmodell mitaufzunehmen. Lediglich ein boolesches Attribut unterscheidet zwischen Ratingskalen und reinen Multiple-Choice-Fragen (die nur auf *Nominalniveau* messen).

Zum Zweck der Auswertung haben wir Views definiert, in denen die Ratingskalen gemeinsam mit den Zahlfragen dargestellt werden. Somit benötigen wir nur *eine* Schnittstelle zum Erstellen statistischer Auswertungen, die sowohl Zahlfragen als auch Ratingskalen abdeckt.

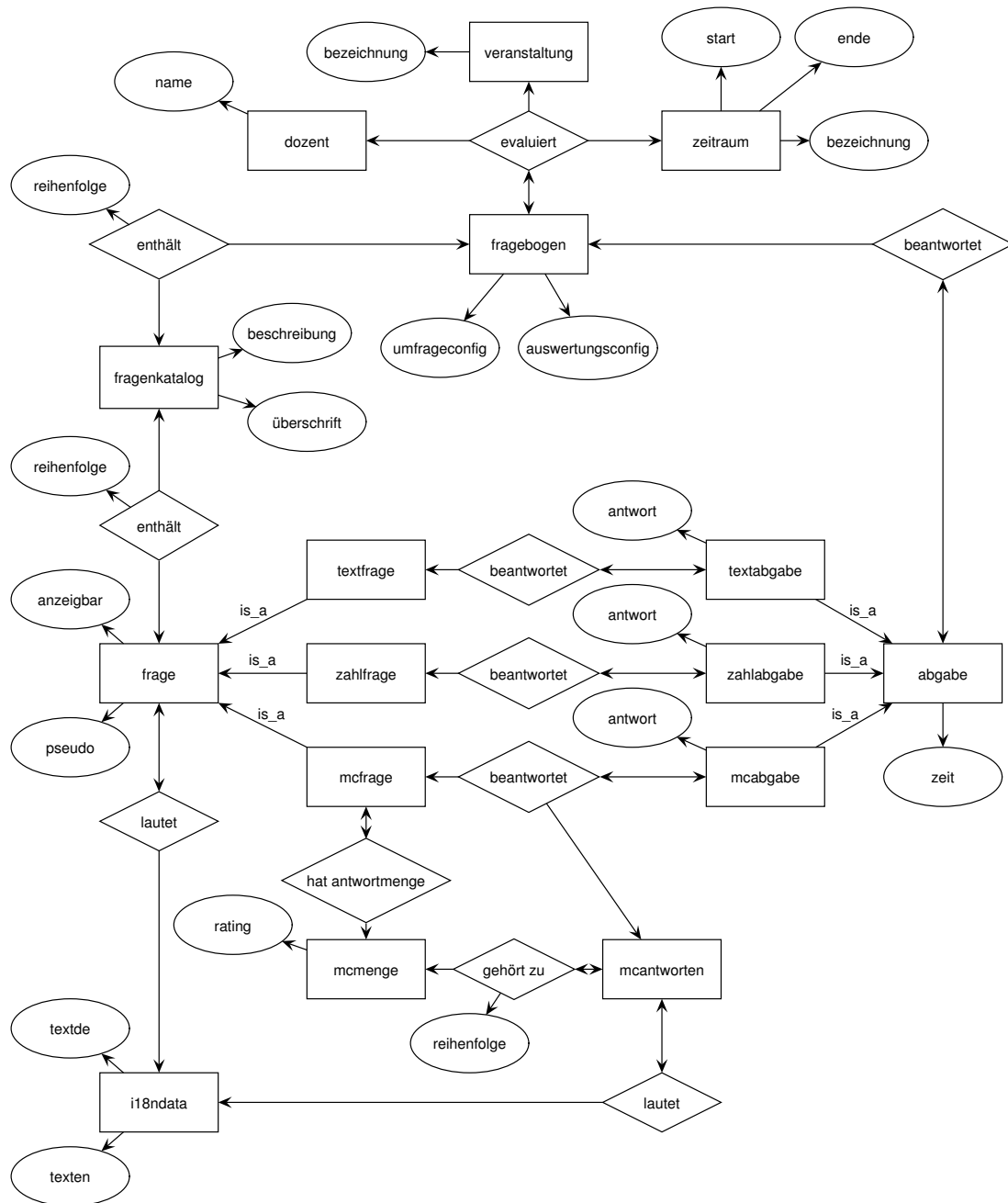


Abbildung 3.4: Entity-Relationship-Diagramm

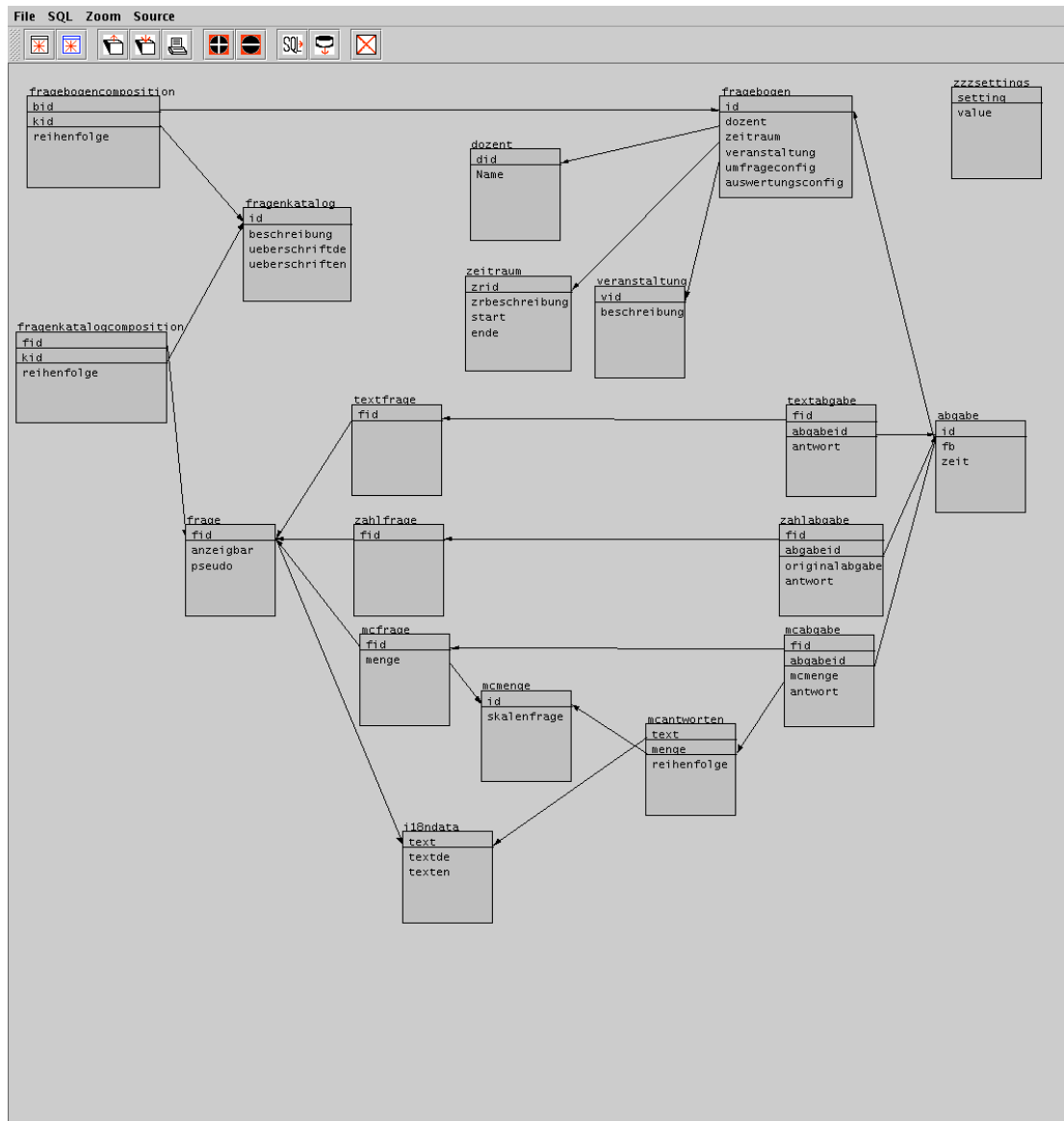


Abbildung 3.5: Datenbankschema aus Sicht von DBmodeler

3.3.4 Pseudofragen

In manchen Einsatzszenarien ist die Speicherung von zusätzlichen Informationen erforderlich, wie z.B. die zum Ausfüllen des Fragebogens benötigte Zeit oder die IP-Adresse des Webbrowsers. Derartige Informationen werden wir durch sogenannte *Pseudofragen* realisieren: "Fragen", die nicht im Fragebogen dargestellt werden und vom Umfrageteilnehmer beantwortet werden, sondern deren Antwort automatisch von einem entsprechenden Handlermodul generiert wird.

Wir werden diese zusätzlichen Informationen auf die gleiche Weise abspeichern und zugänglich machen wie die Antworten auf "echte" Fragen. Somit lässt sich z.B. untersuchen, ob die Teilnehmer den Zeitaufwand für die Teilnahme an der Umfrage richtig einschätzen, wie wir dies im dritten Testlauf 5.1.3 exemplarisch durchführten.

Kapitel 4

Implementierung

Bevor wir die Implementierung beginnen können, müssen wir zunächst eine Entscheidung für die zu verwendenden Programmiersprachen treffen. Nachdem unsere Wahl auf Java und XSLT gefallen ist, beschreiben wir die Details einiger Systemkomponenten. Ausgehend vom Framework-Kern und der Transferstruktur zum Informationsaustausch zwischen Framework-Kern, den Modulen und der Anwendung gehen wir über zu den Details des Web-Interface. Zwei angewandte Design-Pattern verdienen eine besondere Betrachtung: Zunächst erklären wir unsere nach dem *Visitor-Pattern* modellierten Klassen und ihr Zusammenspiel mit anderen Systemkomponenten, gefolgt von unseren automatisch generierten *Data Access Objects*. Schließlich beschreiben wir die von uns bereitgestellten Module, durch die das Framework erst mit Leben erfüllt wird.

4.1 Verwendete Technologien

4.1.1 Betrachtung möglicher Implementierungssprachen

Aufgrund unserer in Abschnitt 3.1 getroffenen Entscheidung, ein Web-basiertes Umfragesystem zu erstellen, kommen als mögliche Programmiersprachen zur Realisierung nur solche in Betracht, die von Haus aus die Programmierung von Web-Anwendungen unterstützen.

Als einige verbreitete plattformunabhängige Sprachen zur Entwicklung Web-basierter Anwendungen betrachten wir PHP, Perl, und Java.

PHP besticht durch seine leichte Erlernbarkeit, die in Hinblick auf den erweiterungsfähigen Charakter unseres Systems zu begrüßen ist. Allerdings fehlt PHP eine Möglichkeit, Variablen explizit deklarieren zu müssen. Alle Variablen werden implizit durch ihre Verwendung eingeführt. Dieses Verhalten ist durchaus in vielen Sprachen verbreitet. So zeigen z.B. auch Perl oder Visual Basic dieses Verhalten; allerdings bieten diese beiden Sprachen die Möglichkeit, eine explizite Deklaration von Variablen zu erzwingen (in Perl mit `use strict;`, in Visual Basic durch `option`

`explicit`). Es gibt zwar in PHP mit `error_reporting(E_NOTICE)` die Möglichkeit, eine Warnung zu erzwingen, wenn eine noch nicht bekannte Variable dereferenziert wird, aber dies deckt nur Lesezugriffe auf nicht bekannte Variablen ab. Ein z.B. durch einen Tippfehler verursachter Schreibzugriff auf eine noch nicht bekannte Variable, wie er in 4.1 dargestellt ist, bewirkt keine Warnung. Da dies der einzige Weg ist, eine neue Variable ins Leben zu rufen wäre eine derartige Warnung auch sinnlos. Daher haben wir uns gegen PHP entschieden, auch wenn PHP trotz (oder gerade wegen?) dieses Mangels weite Verbreitung genießt.

Listing 4.1: Defizit-Demo PHP

```
<?php
    error_reporting(E_NOTICE);
% folgender lesezugriff erzeugt warnung
    echo $lesezugriff;
    $demovvariable=" _Hello_ ";
% folgender schreibzugriff erzeugt keine warnung
    $domovvariable=trim($demovvariable)+"_World";
    echo $demovvariable." \n";
?>
```

Perl kennt mit `use strict;` einen Schutz gegenüber Tippfehlern, die sonst implizit neue Variablen deklarieren würden. Allerdings fehlt dem Objekt-Konzept von Perl eine Kapselung privater Elemente, wie in der Note am Ende der Beschreibung der `perlmodlib` beschrieben ist.

Java wird unseren Anforderungen in Hinblick auf Plattformunabhängigkeit, objektorientierte Programmierung und Absicherung gegen Tippfehler gerecht und ist durch Applicationserver, wie der von uns verwendete `tomcat` bestens geeignet sowohl zur Implementierung des Web-Interfaces als auch für die Gui-Managementschnittstelle, die somit einen großen Teil des Quellcodes miteinander teilen können.

4.1.2 JAAS als Framework-Basis?

Das in 3.2 entworfene modularisierte Handler-Konzept wurde durch das in C implementierte PAM (Pluggable Authentication Modules [VS 96]) inspiriert. Zwar existiert mit JAAS (Java Authentication and Authorization Service) existiert eine JAVA Implementierung des PAM-Konzepts, aber da sich unser Entwurf vom originalen PAM-Konzept unterscheidet (der Handler-Stack muss wie in 3.2 erläutert Vor- und Nachbereitungsschritte abarbeiten) können wir JAAS nicht als Basis für unser Framework verwenden.

4.1.3 Template-Engine für Web-basierte Lösung

Die Einbindung einer Web-basierten Umfrage in eine bestehende Webpräsenz eines Lehrstuhls (oder einer Firma) kann aus Gründen der Ästhetik oder aufgrund von Vorgaben bezüglich des

Erscheinungsbildes der Internetpräsenz (Corporate Identity) eine Anpassung der Darstellung erfordern. Die Darstellungsschicht sollte daher aus Gründen der leichten Anpassbarkeit von der eigentlichen Programmierung des Systems getrennt möglich sein, um tiefgreifendere Änderungen an der Präsentation vornehmen zu können, als dies z.B. durch den Einsatz von Cascading Stylesheets (CSS) möglich ist.

Insbesondere die in Kapitel 2.3.13 motivierte Notwendigkeit, zusätzlich oder anstelle eines Webformulars eine papierbasierten Umfrage durchführen zu können, erfordert eine saubere Trennung zwischen den möglichen Darstellungen und dem Fragebogen-erzeugenden Kern.

Eine derartige Trennung wird oftmals durch den Einsatz sogenannter Template-Engines erreicht, von denen es eine Vielzahl für verschiedene Sprachumgebungen gibt (z.B. Smarty für PHP, Velocity oder JSP für Java). Das Webdokument wird dabei als Schablone erstellt, in die mit Hilfe minimaler Programmlogik des Template-Systems die eigentlichen Daten eingetragen werden. Die Programmierung der Webanwendung ist dadurch getrennt von der Darstellung.

Die Vielzahl existierender Template-Engines zeigt die Vorzüge des Konzepts. Allerdings verwenden die meisten Template-Engines für die Übertragung der Daten zwischen Programm und Engine proprietäre Datenstrukturen, die bei einem Wechsel der Sprache auch einen Wechsel der Template-Engine erforderlich machen. Eine über Sprachgrenzen hinweg standardisierte Template-Engine steht mit XSL Transformations (**XSLT**) zur Verfügung, das als Datenübertragungsformat XML verwendet. XSLT-Transformatoren stehen für etliche Sprachen zur Verfügung, aber durch die Verwendung von XML als Datenschnittstelle wäre das Fehlen einer XSLT-Implementierung für die verwendete Programmiersprache kein Hindernis. So ließe sich z.B. aus einem Perl-Programm heraus das erzeugte XML an einen Java-Prozess übergeben, der die XSL-Transformation durchführt. Moderne Browser können sogar auf Client-Seite XSL-Transformationen durchführen, aber da das XML unter Umständen vertrauliche Daten der Auswertung enthalten kann, die erst durch die Transformation unterdrückt werden, werden wir XSLT nicht auf Clientseite einsetzen.

4.2 Implementierung des Framework-Kerns

Aufgabe des Framework-Kerns ist das Erzeugen des Handlerstacks aus einem XML-Konfigurationsfile und die Abarbeitung des erzeugten Handlerstacks.

Die zu instanzierenden Handlerklassen werden über das Reflection-API geladen.

In einem ersten Ansatz wollten wir eine saubere Definition des Konfigurationsdateiformats in Form einer Document Type Definition (DTD) angeben. Hierzu hätten wir jedem bekannten Handler ein eigenes XML-Tag zugeordnet. Allerdings hätten wir eine Abbildung von Tags auf Handlerklassen benötigt, die durch eine weitere Konfigurationsdatei erfolgt wäre oder komplizierte Mechanismen einer automatischen Registrierung möglicher Klassen erfordert hätte. Beide Wege erschienen uns zu kompliziert und fehleranfällig; daher entschieden wir uns aus Gründen

der Einfachheit nur ein Tag zu verwenden (`handler`) und die zu instanzierende Klasse über das Attribut `class` anzugeben.

Leider können wir somit keine Document Type Definition (**DTD**) mehr angeben, da verschiedene Klassen verschiedene weitere Attribute und gegebenenfalls weitere verschachtelte Elemente erwarten. Die Konfigurationsdateien sind allerdings nicht zum Austausch mit anderen externen Softwarekomponenten vorgesehen. Da eine Validierung von XML-Dateien idealerweise an den Schnittstellen zwischen verschiedenen Komponenten stattfindet, die hier im Falle der Konfigurationsdatei eben nicht vorhanden sind, erscheint der Verlust der Validierbarkeit akzeptabel gegenüber dem Mehraufwand verschiedener Konfigurationstags für verschiedene Handlerklassen.

Das Root-Element einer Framework-Konfiguration ist das `chain`-Element, das keine Attribute enthält und als verschachtelte Elemente nur `handler` Objekte erwartet.

4.3 Transferstruktur

Zur Kommunikation zwischen der Anwendung und den Handlern wird eine Transferstruktur `FWState` an die Aufrufe der `pre-` und `post-`Methoden übergeben. Diese Transferstruktur enthält die interne Fragebogenrepräsentation, deren Struktur aus Abbildung 4.1 hervorgeht und die in ihrem Aufbau dem Datenbankmodell in Abbildung 3.4 entspricht.

Die Unterklasse `XMLWebFWState` ist um folgende weitere Informationen ergänzt, die der Web-basierte Teil des Frameworks zur internen Kommunikation benötigt:

- `HttpServletRequest request`
- `HttpServletResponse response`
- `Document document`
- `String xslt`

Insbesondere durch das `HttpServletResponse` Objekt sind die Handler im Prinzip in der Lage, unter Umgehung des Anwendungsservlets direkt ein Ergebnis an den Webbrowser zu senden falls besondere Umstände dies erforderlich machen.

4.4 Implementierung des Webinterfaces

Die vier Anwendungsfälle aus Abbildung 3.1, die durch das Web-Interface realisiert werden sollen, ähneln sich weitestgehend. Wir haben eine abstrakte Servletklasse `VEFServlet` konstruiert, die die Verwaltung der Sitzungsinformation und der Transferstruktur übernimmt und für jeden Anwendungsfall eine davon abgeleitete Servletklasse erstellt.

Die beiden öffentlichen Servlets `UmfrageServlet` und `AuswertungsServlet` beziehen ihre Handlerkonfiguration für den jeweiligen Fragebogen aus den entsprechenden Datenbankfeldern `umfrageconfig` und `auswertungsconfig`, während die beiden internen Servlets `ReportServlet` und `AntwortbogenServlet` jeweils eine festverdrahtete minimale Konfiguration aus `DatenbankHandler` und `ReportHandler` bzw. `AntwortbogenHandler` zur Initialisierung des Frameworks verwenden.

Nach dem Abarbeiten der Handlerchain wird das in der Transferstruktur von den Handler aufgebaute XML-Dokument durch einen `java.xml.transform.Transformer` mit Hilfe in der Transferstruktur angeforderten Stylesheets zur Antwort an den Webbrowser umgewandelt.

Nach einem Ergebnis der Handlerchain, das nicht `Undef` ist, steht entweder der Erfolg oder der Misserfolg des jeweiligen Anwendungsfalles fest und die Sitzung wird vom `VEFServlet` zerstört.

Die Erzeugung der XML-Fragebogenstruktur wird von einer der vier Handlerklassen `WebPresentationHandler`, `WebAuswertungsHandler`, `WebReportHandler` oder `WebAntwortbogenHandler` durch Anwenden einer Unterklasse des Visitors `FBXMLer` auf die interne Fragebogenrepräsentation bewirkt, wobei die jeweilige Visitor-Implementierung die Fragebogenstruktur mit zusätzlichen Daten anreichert.

Das Zusammenspiel zwischen Servlets, Visitors und der Fragebogenstruktur ist in Abbildung 4.1 näher dargestellt, wobei aus Gründen der Übersichtlichkeit die Interaktion der Servlets mit dem Framework und den Handlern in der Operation `runs_via_handler` zusammengefasst wurden.

4.5 Verwendung des Visitor-Patterns

Wie im vorigen Abschnitt erwähnt, verwenden wir eine Visitor-Klasse um die interne Fragebogenrepräsentation in ein XML-Fragment zu konvertieren. Wir beschreiben nun die Aufgaben aller in Abbildung 4.1 erscheinenden Visitorklassen.

FBVisitor

`FBVisitor` ist die Basisklasse unserer Fragebogenvisitors. Im Zusammenspiel mit den `invite`-Methoden der Fragebogenstruktur werden alle in Abbildung 4.2 aufgeführten Methoden aufgerufen.

Listing 4.2: Beispielsignatur eines Handlers

```
public abstract class FBVisitor {
    protected void preFragebogen(Fragebogen fb) {}
    protected void postFragebogen(Fragebogen fb) {}
    protected void preFragekatalog(FragenKatalog fk) {}
}
```

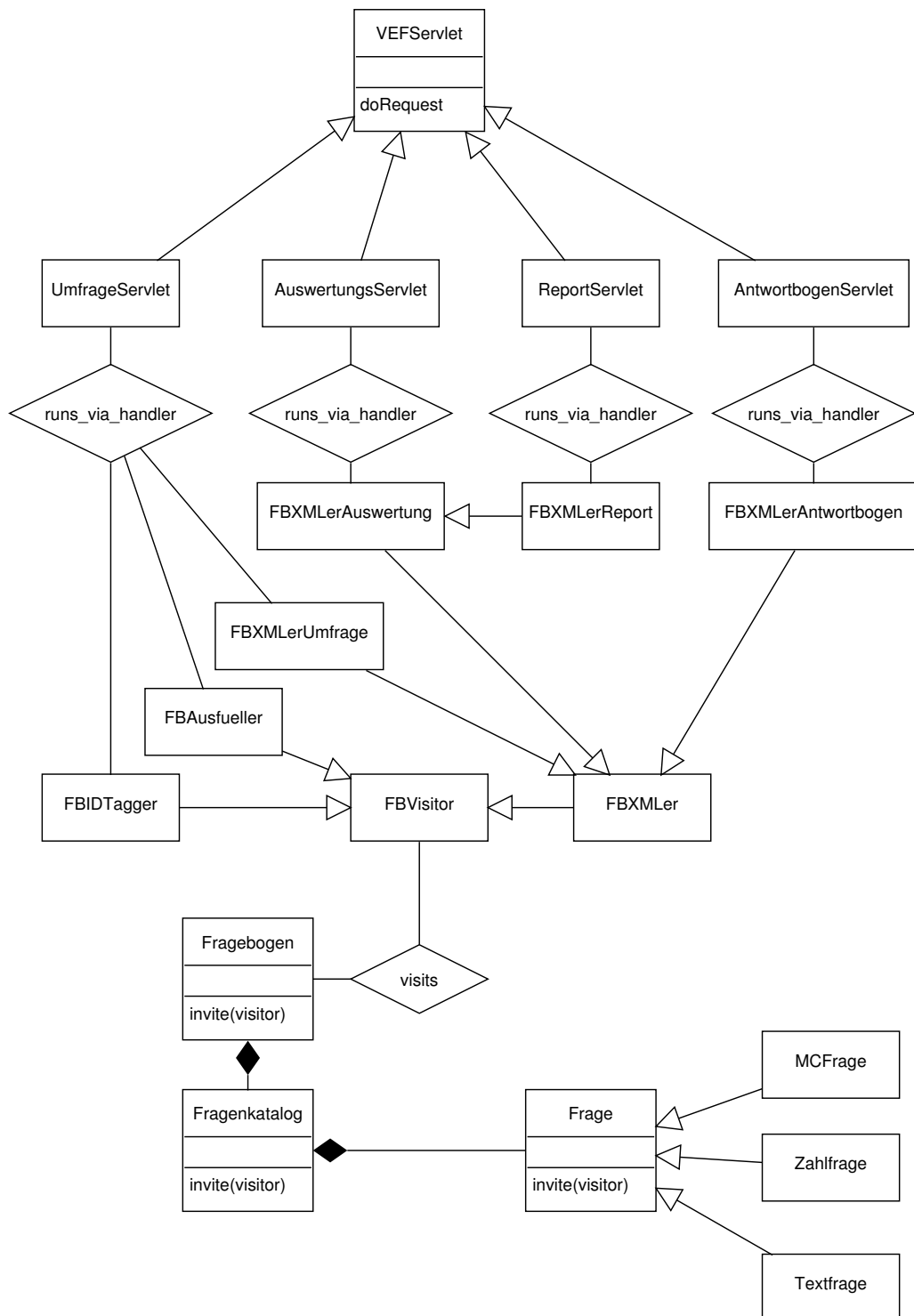


Abbildung 4.1: Interaktion von Servlet und Visitor

```
protected void postFragekatalog(FragenKatalog fk) {}  
protected void visitFrage(FrageInt f) {}  
protected void visitFrage(FrageMC f) {}  
protected void visitFrage(FrageText f) {}  
}
```

FBIDTagger

Der Visitor `FBIDTagger` versieht jede Frage mit einer einmaligen ID. Er wird vom in Abschnitt 4.7.7 vorgestellten `WebPresentationHandler` verwendet.

FBAusfueller

Der Visitor `FBAusfueller` identifiziert anhand der von `FBIDTagger` vergebenen IDs die internen Fragedarstellungen mit den ausgefüllten Formularfeldern des `HTTPRequest`. Wenn eine gültige Antwort erkannt wird, wird diese in der jeweiligen Frage eingetragen, damit der `DatenbankHandler` sie von dort schließlich in die Datenbank überführen kann. Der Visitor `FBAusfueller` wird vom in Abschnitt 4.7.7 vorgestellten `WebPresentationHandler` verwendet.

FBXMLer

Der Visitor `FBXMLer` überführt die interne Fragebogenrepräsentation in ein XML-Fragment. Seine Unterklassen können dieses Fragment mit weiteren Daten anreichern.

FBXMLerUmfrage

Der Visitor `FBXMLerUmfrage` vermerkt die im `HTTPRequest` gegebenen Antworten wiederum im Fragebogen-XML-Fragment, damit bei einem erneuten Laden der Fragebogenansicht im Webbrowser die bisher gegebenen Antworten nicht verloren gehen. Der Visitor `FBXMLerUmfrage` wird vom in Abschnitt 4.7.7 vorgestellten `WebPresentationHandler` verwendet.

FBXMLerAuswertung

Der Visitor `FBXMLerAntwortbogen` lädt die verlangte aggregierte Auswertung aus der Datenbank und reichert das Fragebogen-XML-Fragment mit den entsprechenden statistischen Daten an. Der Visitor `FBXMLerAuswertung` wird vom in Abschnitt 4.7.8 vorgestellten `WebAuswertungHandler` verwendet.

FBXMLerReport

Wie man in Abbildung 4.1 erkennt, ist `FBXMLerReport` eine Unterklasse von `FBXMLerAuswertung`. Während `FBXMLerAuswertung` freie Fragen und Fragen, die nicht zur öffentlichen Anzeige freigegeben sind, unterdrückt, stellt `FBXMLerReport` alle Daten der Abgabe zur Verfügung. Der Visitor `FBXMLerReport` wird vom in Abschnitt 4.7.9 vorgestellten `WebReportHandler` verwendet.

FBXMLerAntwortbogen

Der Visitor `FBXMLerAntwortbogen` lädt die verlangte Bewertungsabgabe aus der Datenbank und reichert das Fragebogen-XML-Fragment mit den abgegebenen Antworten an. Der Visitor `FBXMLerAntwortbogen` wird vom in Abschnitt 4.7.10 vorgestellten `WebAntwortbogenHandler` verwendet.

4.6 Verwendung von Data Access Objects (DAO)

Das Konzept von *Data Access Objects* bringt bei der Programmierung den Vorteil, daß die Strukturen der in der Datenbank gespeicherten Datensätze auf Strukturen der verwendeten Programmiersprache abgebildet werden. Dadurch können Änderungen des Datenbankschemas gegenüber direkten Anfragen über die Datenbankschnittstelle (hier: JDBC) leichter verfolgt werden, da sich veränderte Tabellenstrukturen in veränderten Java-Klassen niederschlagen und ein Zugriff auf z.B. nicht mehr vorhandene Attribute sich in Kompilierfehlern äußert. Bei der direkten Verwendung von SQL-Queries im Java-Code müssen diese manuell angepasst werden, wenn das Datenbankschema geändert wird, und eine vergessenen Anpassung fällt erst beim Ausführen der entsprechenden Datenbankabfrage auf.

In der Regel werden *Data Access Objects* (kurz: DAOs) eingesetzt, um eine Unabhängigkeit von der verwendeten Datenbank zu erzielen, da diese als Zwischenschicht eine saubere Trennung von Anwendungscode und Datenbank bewirken, so daß beispielsweise durch eine Anpassung der DAOs eine relationale Datenbank durch eine XML-Datenbank oder einen Verzeichnisdienst ersetzt werden kann.

Diese Zielsetzung ist für uns weniger interessant, da wir keinen Austausch unserer Datenbankschicht vorhaben. Wir machen uns lediglich die eingangs erwähnte Verlagerung von potentiellen Laufzeitfehlern zu Compilezeitfehlern zu nutze.

Die Beschreibung des DAO-Patterns unter Java geht leider nicht auf Möglichkeiten zur automatischen Generierung der DAO-Klassen ein, sondern sieht eine manuelle Erstellung vor.

Mit dem Werkzeug `DbModeller`¹ existiert ein *automatischer* DAO-Generator, der unter

¹<http://www.horsman.co.nz/story.do?id=24>

anderem über ein Back-End für die von uns eingesetzte PostgreSQL Datenbank verfügt. `DbModeller` ist kostenlos erhältlich und steht unter der GNU General Public License. Der dadurch ebenfalls frei erhältliche Quellcode versetzt uns in die Lage, ein paar Anpassung am DAO-Code-Generator vorzunehmen, und so die erzeugten DAO-Klassen an unsere Vorstellungen anzupassen.

Darüberhinaus verfügt `DbModeller` über ein GUI-Interface zur Darstellung der Tabellen und ihrer Fremdschlüsselbeziehungen, wie in Abbildung 3.5 dargestellt.

4.7 Implementierte Module

Wir beginnen jede Modulbeschreibung mit den benötigten Attributen für das XML-Konfigurationsfragment des jeweiligen Handlers, und gegebenenfalls einer Erwähnung zulässiger verschachtelter Elemente. Das Pflichtattribut `flag`, das die Werte `required` oder `optional` annehmen kann, haben wir dabei nicht jedesmal erneut aufgeführt. Sodann beschreiben wir das Verhalten der beiden Interfacemethoden `pre` und `post`, gefolgt von Anmerkungen zu Besonderheiten in der Implementierung dieses Handlers.

4.7.1 DatenbankHandler

```
class de.itfuzzi.diplom.core.handler.DatenbankHandler
```

Verhalten des *pre*-Schrittes:

Aus der Transferstruktur wird die Nummer des gewünschten Fragebogens ausgelesen. Eine dem Fragebogen entsprechende interne Repräsentation wird in der Transferstruktur erstellt und wenn keine Fehler auftraten ist das Ergebnis `true`.

Verhalten des *post*-Schrittes:

Die Antworten werden aus der internen Fragebogen-Repräsentation der Transferstruktur ausgelesen und in die Datenbank überführt. Wenn dabei kein Fehler auftrat, ist das Ergebnis `True`, sonst `False`.

Anmerkungen:

Der Zugriff auf die Datenbank sollte in einer ersten Konzeption außerhalb des Handlerstacks stattfinden, da hierfür keine freie Konfigurierbarkeit notwendig ist. Aber ein mögliches Einsatzszenario erforderte die Verlegung der Datenbankinteraktion in einen eigenen Handler, da *nach* dem Schreiben des Abstimmungsergebnisses in die Datenbank weitere Handlertätigkeit erforderlich wurde: Wenn wie in Abschnitt 2.3.12 ein personalisierter Bonus als Abstimmungsanreiz ausgegeben werden soll, ist die Abfrage persönlicher Daten notwendig. Allerdings nur, wenn der Abstimmende seinen Bonus abrufen möchte. Man soll also die Abfrage persönlicher Daten bis zum notwendigen Zeitpunkt *nach* der Abgabe des Fragebogens hinauszögern können, so daß dem Abzustimmenden die Wahl offen bleibt zwischen anonymer Stimmabgabe und Erhalt eines persönlichen Bonus.

Zu Beginn einer Bewertung wird die jeweilige Fragebogenkonfiguration aus der Datenbank bezogen. Diese besteht im wesentlichen aus dem für diesen Fragebogen vorgesehenen Handlerstack. Diese Handler werden nun in der vorgegebenen Reihenfolge durchgegangen, solange die *pre*-Methoden der Handler ein Weiterfortschreiten signalisieren.

Nach einem erfolgreichen Abschluss des *pre*-Weges werden die bisher verwendeten Handler in umgekehrter Reihenfolge mit ihrer *post*-Methode aufgerufen. Wenn auch hierbei kein Fehler auftritt, wird die Bewertung vom Datenbankhandler in die Datenbank eingetragen.

Da der Aufruf der Handlerkette an jeder Stelle durch eine Rückfrage an den Benutzer unterbrochen werden kann und der betreffende Handler in diesem Fall durch das Ergebnis `null` anzeigt, daß der Handlerstack später weiterverarbeitet werden soll (sprich: Beim nächsten Formularauf-ruf, damit der Handler seine benötigten Informationen erhält) dürfen die einzelnen Handler nicht erwarten, daß ihre `post`-Methode jemals aufgerufen wird, da der Benutzer sich entschliessen kann den Fragebogen nicht weiter zu bearbeiten und der erwartete Formularaufruf zum Fortsetzen des Handlerstacks nie eintritt.

Würde sich nun der Abstimmende gegen die Preisgabe persönlicher Information zwecks Erhalt des Bonbons entscheiden, so würde der Handlerstack nie zum Abschluss kommen und das Bewertungsergebnis würde nie in die Datenbank eingetragen. Durch die Verlegung des Datenbankzugriffs in einen eigenen Handler wird dieses Problem umgangen, da der Datenbankhandler in einem Nebenzweig (siehe 4.7.3) fertig abschliessen kann und dann erst die für die Bonbonauslieferung notwendige Abfrage persönlicher Information startet. Ein Abbrechen des Vorgangs an dieser Stelle durch den Bewertenden würde ein Abspeichern der Bewertung in der Datenbank nicht mehr verhindern.

4.7.2 MinimalzeitHandler

```
class de.itfuzzi.diplom.core.handler.MinimalzeitHandler  
minimumsec Minimale Zeit in Sekunden, die nicht unterschritten werden darf
```

Verhalten des **pre**-Schrittes:

Die aktuelle Uhrzeit wird gespeichert. Das Ergebnis ist `True`.

Verhalten des **post**-Schrittes:

Die gespeicherte Zeit wird mit der aktuellen Uhrzeit verglichen. Wenn der verstrichene Zeitraum größer ist als durch das Konfigurationsattribut `minimumsec` gefordert, so ist das Ergebnis `True`, ansonsten `False`.

Anmerkungen:

Dieses Modul kann für die in Abschnitt 2.3.9 erwähnte Plausibilitätskontrolle eingesetzt werden. Siehe auch Modul 4.7.13.

4.7.3 ChainHandler

```
class de.itfuzzi.diplom.core.handler.ChainHandler
```

Children Als verschachtelte Elemente ist genau ein `chain-Element` zulässig.

Verhalten des `pre`-Schrittes:

Die verschachtelte Chain wird ausgeführt und ihr Ergebnis zurückgeliefert.

Verhalten des `post`-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler ist notwendig für die in Abschnitt 2.3.12 erwähnte Technik, die persönlichen Daten erst nach dem Abgeben des Fragebogens zu erfassen. Die daraus resultierenden Details sind in 4.7.1 detailliert beschrieben.

4.7.4 DelayHandler

```
class de.itfuzzi.diplom.core.handler.DelayHandler  
delaysec Wartezeit in Sekunden, die der pre-Aufruf pausiert
```

Verhalten des pre-Schrittes:

Nach der konfigurierten Wartezeit wird das Ergebnis `True` geliefert.

Verhalten des post-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieses Modul kann in Verbindung mit einer Beschränkung von gleichzeitigen Verbindung derselben Absenderadresse zum Erschweren von Brute-Force-Angriffen auf ein Zugangspasswort eingesetzt werden.

4.7.5 FalseHandler

```
class de.itfuzzi.diplom.core.handler.FalseHandler
```

Verhalten des pre-Schrittes:

Das Ergebnis ist immer `False`.

Verhalten des post-Schrittes:

Das Ergebnis ist immer `False`.

Anmerkungen:

Dieser Handler kann zum frühzeitigen Terminieren einer Handlerchain verwendet werden.

4.7.6 TrueHandler

```
class de.itfuzzi.diplom.core.handler.TrueHandler
```

Verhalten des pre-Schrittes:

Das Ergebnis ist immer True.

Verhalten des post-Schrittes:

Das Ergebnis ist immer True.

Anmerkungen:

Dieser Handler kann eingebaut werden, um die hinteren Teile einer Handlerchain für Testzwecke zu deaktivieren.

4.7.7 WebPresentationHandler

class `de.itfuzzi.diplom.web.handler.WebPresentationHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

Verhalten des `pre`-Schrittes:

Falls dies der erste Aufruf der `pre` Methode ist, wird durch den Visitor `FBIDTagger` in der internen Fragebogenrepräsentation jeder Frage eine eindeutige ID zugewiesen, die zur Unterscheidung der verschiedenen Eingabefelder im HTML-Formular dienen wird. Durch den Visitor `FBXMLer` wird aus der internen Fragebogenrepräsentation eine XML-Darstellung generiert. Als Ergebnis wird `True` geliefert, wenn alle Antworten auf die Formularfelder gültige Angaben enthalten, ansonsten `Undef`.

Verhalten des `post`-Schrittes:

Die erfassten Antworten werden durch den Visitor `FBAusfueller` in der internen Fragebogenrepräsentation der Transferstruktur eingetragen. Das Ergebnis ist `True`.

Anmerkungen:

Aufgabe dieses Handlers ist, einerseits die Darstellung des Fragebogens zu veranlassen, und andererseits den ausgefüllten Fragebogen anzunehmen und die Antworten auf ihre Gültigkeit (z.B. nur ganze Zahlen auf eine Frage nach dem Alter) hin zu überprüfen. Solange ungültige Antworten gegeben werden, wird dieser Handler durch das Ergebnis `Undef` nachfragen.

Die Darstellung wird aus dem in der Transferstruktur übergebenen Fragebogen durch einen geeigneten Visitor als XML-Teildokument erstellt und vom Anwendungsservlet durch das von diesem Handler gewünschte XSLT-Stylesheet zur Ausgabe an den Webbrowser umgewandelt.

4.7.8 WebAuswertungsHandler

class `de.itfuzzi.diplom.web.handler.WebAuswertungsHandler`
sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

Verhalten des `pre`-Schrittes:

Dieser Handler liest das Auswertungsergebnis aus der Datenbank ein und liefert immer `Undef`.

Verhalten des `post`-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler dient der Präsentation eines Umfrageergebnisses. Da eine Nachbereitung für den Zweck dieses Handlers nicht erforderlich ist, liefert diese Handler stets `Undef` und verursacht daher nie den Übergang zu den Nachbereitungsschritten.

Durch das Ergebnis `Undef` wird die Verarbeitung nicht beendet, so daß die Auswertung beliebig oft neu angefragt werden kann. Dabei wird bei jedem Aufruf der `pre`-Methode erneut die Datenbank konsultiert, so daß ein erneutes Laden der Seite im Webbrowser etwaige Änderungen am Ergebnisstand anzeigt.

4.7.9 WebReportHandler

class `de.itfuzzi.diplom.web.handler.WebReportHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

Verhalten des **pre**-Schrittes:

Dieser Handler liefert `False`, wenn der Benutzer nicht ordnungsgemäß mit der Rolle `vefauswerter` authentifiziert wurde. Der Handler liest dann das Auswertungsergebnis aus der Datenbank ein und liefert immer `Undef`.

Verhalten des **post**-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Die Anmerkungen zu 4.7.8 gelten analog.

Um datenschutzrechtliche Probleme zu vermeiden, überprüft dieser Handler, ob der Benutzer vom Applicationserver authentifiziert und mit der Rolle `vefauswerter` versehen ist. Somit sind die persönlichen Daten und Antworten auf freie Fragen auch dann nur für berechtigte Personen zugänglich, wenn dieser Handler versehentlich in einer öffentlich zugänglichen Ansicht konfiguriert wurde.

4.7.10 WebAntwortbogenHandler

class `de.itfuzzi.diplom.web.handler.WebAntwortbogenHandler`
sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

Verhalten des `pre`-Schrittes:

Dieser Handler liefert `False`, wenn der Benutzer nicht ordnungsgemäß mit der Rolle `vefauswerter` authentifiziert wurde. Ansonsten wird der im HTTP-Request verlangte ausgefüllte Fragebogen generiert und das Ergebnis `Undef` geliefert.

Verhalten des `post`-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Die Anmerkungen zu 4.7.8 und 4.7.9 gelten analog.

4.7.11 PasswordHandler

class `de.itfuzzi.diplom.web.handler.PasswordHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

Verhalten des **pre**-Schrittes:

Der Handler vermerkt im XML-Dokument der Transferstruktur die Anforderung eines Passworts mit der konfigurierten Eingabeaufforderung. Wenn das konfigurierte Passwort im `HTTPRequest` der Transferstruktur erkannt wird, liefert der Handler `True`. Ansonsten liefert er `Undef`.

Verhalten des **post**-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler kann sowohl als `required` als auch als `sufficient` sinnvoll eingesetzt werden, wie dies im Abschnitt 3.2 dargestellt wird.

4.7.12 BonbonHandler

class `de.itfuzzi.diplom.web.handler.BonbonHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

text Textuelle Mitteilung, die als Hinweis ausgegeben wird

url Verweis auf das "Bonbon"

Verhalten des **pre**-Schrittes:

Das Ergebnis ist immer `True`.

Verhalten des **post**-Schrittes:

Ein Verweis auf das Bonbon wird mit dem konfigurierten Text generiert. Das Ergebnis ist immer `True`.

Anmerkungen:

Dieses Modul lässt sich, wie in Abschnitt 2.3.12 motiviert wurde, zur Steigerung der Teilnahmequote einsetzen. Um zu vermeiden, daß der Link zur Belohnungsressource direkt verbreitet wird und somit eine Teilnahme an der Umfrage nicht mehr erforderlich, verwenden wir ein von uns speziell erstelltes `PassThroughServlet`. Hierbei wird die URL des Bonbons vom Bonbonhandler in einer Session-Variable gespeichert und nur ein Verweis auf das `PassThroughServlet` ausgegeben. Dieser Verweis ist ohne Session mit darin gespeicherter Zielinformation nutzlos, da das `PassThroughServlet` sein Ergebnis-Dokument von der in der Sessionvariable gespeicherten Adresse bezieht, und somit außerhalb einer Session keinen Inhalt liefert.

Wenn das Bonbon nicht aus einem externen Dokument, sondern nur aus einer kleinen Textmitteilung besteht, wie z.B. dem Zugangspasswort zum Einsehen der Auswertungsergebnisse, bietet sich der Einsatz von Modul 4.7.16 an.

4.7.13 PseudoFrageAusfuelldauer

```
class de.itfuzzi.diplom.core.handler.PseudoFrageAusfuelldauer
```

Verhalten des **pre**-Schrittes:

Die Pseudo-Zahlfrage mit Fragetext “PSEUDOFRAGE Ausfuelldauer” wird im geladenen Fragebogen erzeugt (und wenn sie in der Datenbank noch nicht existieren sollte auch dort angelegt). Die aktuelle Uhrzeit wird vermerkt. Das Ergebnis ist immer `True`.

Verhalten des **post**-Schrittes:

Die verstrichene Zeit wird als Antwort auf die Pseudofrage “ausgefüllt”. Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler dient zur Erfassung der Zeit, die zum Ausfüllen des Fragebogens benötigt wurde. Damit lassen sich die in Abschnitt 2.3.9 angesprochenen Probleme reduzieren. Dieser Handler ist eine sanftere Alternative zum Einsatz von 4.7.2, da er keine Bewertung ablehnt.

4.7.14 PseudoFrageSourceIP

```
class de.itfuzzi.diplom.web.handler.PseudoFrageSourceIP
```

Verhalten des pre-Schrittes:

Die Pseudo-Textfrage mit Fragetext “PSEUDOFRAGE Source IP” wird im geladenen Fragebogen erzeugt (und wenn sie in der Datenbank noch nicht existieren sollte auch dort angelegt). Aus dem in der Transferstruktur übergebenen HTTP-Request Objekt wird die Client-seitige IP-Adresse ausgelesen und als Antwort auf die Pseudofrage “ausgefüllt”. Das Ergebnis ist immer True.

Verhalten des post-Schrittes:

Das Ergebnis ist immer True.

Anmerkungen:

Der Einsatz dieses Moduls bewirkt die Erfassung von Verbindungsdaten und erfordert daher geeignete datenschutzrechtliche Vorkehrungen, wie z.B. eine Information der Umfrageteilnehmer über die gespeicherten Daten und deren Verwendungszweck.

4.7.15 ZeitfensterHandler

class `de.itfuzzi.diplom.web.handler.ZeitfensterHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

from Start des gültigen Zeitfensters

till Ende des gültigen Zeitfensters

toearlymsg Nachricht, die ausgegeben wird, wenn der Handler zu früh ausgeführt wurde

toolatmsg Nachricht, die ausgegeben wird, wenn der Handler zu spät ausgeführt wurde

Verhalten des **pre**-Schrittes:

Dieser Handler liefert `True`, wenn die aktuelle Zeit im konfigurierten `zeitfenster` liegt. Ansonsten wird die Nachricht `toearlymsg` oder `toolatmsg` im XML-Dokument der Transferstruktur erzeugt und als Ergebnis `False` geliefert.

Verhalten des **post**-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler dient zur zeitlichen Beschränkung von Umfragen oder Auswertungen. Wenn eine der optionalen Zeitfenstergrenzen fehlt gilt der gültige Zeitraum als in diese Richtung unbegrenzt.

4.7.16 MessageHandler

class `de.itfuzzi.diplom.web.handler.MessageHandler`

sheet XSLT-Datei, die zur Darstellung verwendet werden soll.

pretext Mitteilung, die im `pre`-Schritt erzeugt werden soll

posttext Mitteilung, die im `post`-Schritt erzeugt werden soll

Verhalten des `pre`-Schrittes:

Dieser Handler liefert immer `Undef`

Verhalten des `post`-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieser Handler dient zum Erzeugen von Mitteilungen, wie z.B. einer Information über Besonderheiten dieser Umfrage, oder einem Dank für die Teilnahme an der Umfrage.

Da der `pre`-Schritt `stest Undef` liefert, ist dieser Handler vor allem als `sufficient` und direkt vor der Fragebogendarstellung zu verwenden, so daß die Nachricht auch bei erneutem Laden der Web-Seite weiterhin dargestellt wird.

Als Mittel zur in Abschnitt 2.3.12 beschriebenen Motivationssteigerung ist dieser Handler weniger gut geeignet als 4.7.12, da keine Verweise auf externe Dokumente dargestellt werden können.

4.7.17 AuthHandler

```
class de.itfuzzi.diplom.web.handler.AuthHandler
```

```
role Rolle, die der Benutzer innehaben muss
```

Verhalten des pre-Schrittes:

Wenn der Benutzer vom Applicationserver authentifiziert werden konnte und die Rolle `role` innehat, wird `True` als Ergebnis geliefert.

Verhalten des post-Schrittes:

Das Ergebnis ist immer `True`.

Anmerkungen:

Dieses Modul kann im Zusammenspiel mit der Konfiguration des Applicationserver verwendet werden, um die in den Abschnitten 2.3.10 und 2.3.11 beschriebenen Zugangsbeschränkungen auf Basis von Zertifikaten zu realisieren.

Der Handler überprüft lediglich, ob der Benutzer authentifiziert wurde, und lagert die eigentliche Zugangskontrolle an den Applicationserver aus. Somit steht uns die gesamte vom Applicationserver angebotene Palette an Authentifizierungsverfahren zur Verfügung.

Kapitel 5

Abschliessende Betrachtung

5.1 Erfahrungen aus den ersten Testläufen

Erste Tests des Systems wurden im Juli 2005 im Rahmen zweier Veranstaltungen aus unterschiedlichen Bereichen der Fakultät durchgeführt: Zum ersten Test wurde die Vorlesung “Informatik 4” herangezogen, eine Vorlesung des Grundstudiums Informatik, die im Sommersemester 2005 von Prof. Dr. Martin Hofmann gelesen wurde.

Als zweite Veranstaltung bot sich zeitgleich der Kurs “Theoretische Informatik” aus dem SIGNAL-Projekt (Sofortprogramm Informatik am Gymnasium – Nachqualifikation von Lehrkräften) an.

Ein dritter Test erfolgte im November 2005 im Rahmen einer weiteren SIGNAL-Veranstaltung und gestattete eine Erprobung der *Pseudofragen*.

Die beiden existierenden Fragebögen wurden möglichst originalgetreu in die elektronische Darstellung überführt, damit eine Auswertung im Rahmen der existierenden Systeme möglich ist.

Zusätzlich wurde der in Abbildung 5.2 dargestellte Block mit Fragen zum elektronischen Auswertungssystem hinzugefügt.

Zweck der Fragen zum Auswertungssystem war einerseits, die Akzeptanz des elektronischen Systems zu erfassen und andererseits eine Beurteilung der beiden Design-Alternativen Radiobuttons vs. Auswahllisten zu erhalten. Das Ergebnis der drei Testläufe sehen wir in Abbildung 5.3.

5.1.1 Ergebnis der Systembeurteilung

Die Beurteilung des elektronischen Systems lag im Schnitt um die 2 auf einer fünfstufigen Ratingskala.

In der freien Frage nach der Urteilsbegründung wurde das System teilweise als fortschrittlich gelobt, andererseits wurde die komplett offene Konfiguration kritisiert. Hieraus erkennen wir, daß gewisse Zugangsrestriktionen das Vertrauen in das System erhöhen könnten.

5.1.2 Vorzug von Radiobuttons gegenüber Auswahllisten

Die starke Tendenz zu Radiobuttons deckt sich mit dem Hinweis in [Welker 05], Drop-down-Boxen sparsam einzusetzen, da sie durch die versteckten Inhalte die Übersichtlichkeit beeinträchtigen.

Allerdings muß man bedenken, daß unsere Ratingskalen durch fehlende Textanker in den Mittelpositionen ungeeignet für diese Darstellung waren. Dennoch sprach sich ca. ein Drittel der Umfrageteilnehmer für Auswahllisten aus und begründete dies teilweise mit der besseren Bedienbarkeit bei Verwendung der Tastatur zur Navigation.

Die Teilnahmequote lag beim ersten Test mit 69 abgegebenen Stimmen von anfänglich 100 Vorlesungsteilnehmern, erstaunlich hoch, zumal die Hörerzahl gegen Ende der Veranstaltung auf ca. 40 abgenommen hatte. Da das System jedoch in komplett offener Konfiguration eingesetzt wurde, werden wohl ein gut Teil mehrfache Stimmabgaben enthalten sein.

Die unterschiedlichen Vorlieben für elektronische und papierbasierte Umfragen verdeutlichen eine generell geteilte Meinung, die sich auch auf Seiten der Dozenten und Veranstalter findet.

5.1.3 Messung der Ausfülldauer durch eine PseudoFrage

In Abbildung 5.1 haben wir die im dritten Testlauf durchgeführte Messung der Ausfülldauer gegen den von den Testpersonen geschätzten Zeitaufwand für ihre Teilnahme an der Umfrage aufgetragen. Hierbei erkennen wir eine gewisse Streuung zwischen geschätzter und gemessener Zeit, wobei insbesondere die starken Abweichungen in Richtung einer deutlich länger gemessenen Ausfülldauer ein Problem des Messverfahrens aufzeigen: Gemessen wird die Zeit zwischen dem Laden und Absenden des Fragebogens, und nicht die Zeit die von der Testperson zum tatsächlichen Bedienen des Webformulars aufgewendet wird.

Diese Diskrepanz kann beispielsweise in folgendem Szenario entstehen: Eine hypothetische Testperson besucht die Vorlesungswebseite und öffnet den Link zum Fragebogen in einem neuen Fenster, um die Teilnahme nicht zu vergessen. Zunächst widmet sich die Testperson allerdings wichtigeren Aspekten der Vorlesungsseite, wie z.B. Übungsaufgaben oder Klausuranmeldungen. Erst nach Erledigung dieser Aufgaben und dem Schließen des Browserfensters stolpert unsere hypothetische Testperson über den – extra zu diesem Zweck in einem neuen Fenster geladenen – Fragebogen und beginnt mit der Bewertung. Die dann durchgeführte Bewertung mag aus Sicht des Systems eine Stunde gedauert haben, während das tatsächliche Ausfüllen vielleicht nur zwei Minuten in Anspruch nahm.

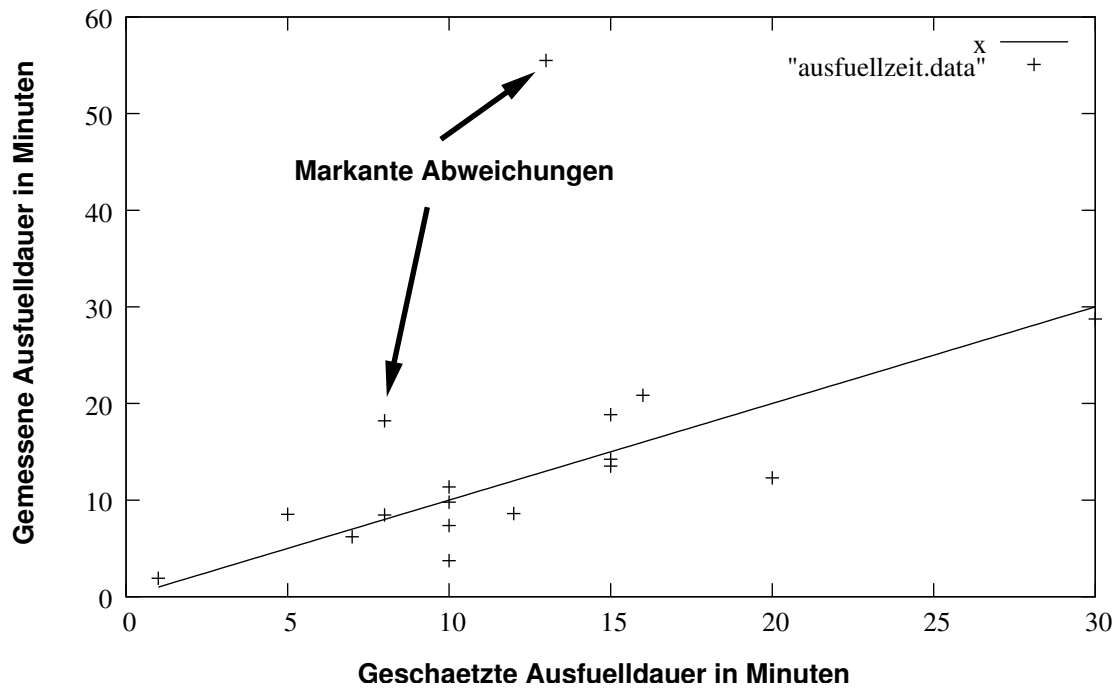


Abbildung 5.1: Geschätzte vs. gemessene Ausfülldauer

5.2 Fazit

Abschließend kann gesagt werden, daß das System die gesetzten Anforderungen erfüllen kann. Es sind keine Nachteile gegenüber dem papierbasierten System auffällig geworden. Die Testläufe sind jedoch nicht mit einem realen Einsatz zu vergleichen, da die Vorlesungshörer extra gebeten wurde, das System zu erproben.

Bewertung des Abstimmungssystems

Wieviele Minuten glauben Sie für das Ausfüllen dieses Fragebogens gebraucht zu haben?

Wie finden Sie das elektronische Bewertungssystem im Vergleich zu den Papierfragebögen?

- 1 hervorragend 2 3 4 5 sehr schlecht
- Ich möchte keine Angaben machen

Warum?

Teilweise konnten Sie zur Beantwortung der Fragen Radiobuttons verwenden und teilweise Auswahllisten. Welche Darstellungsart empfanden Sie als besser geeignet?

- 1 Auswahlliste
- 0 Radiobuttons
- Ich möchte keine Angaben machen

Haben Sie Verbesserungsvorschläge?

Abbildung 5.2: Fragen zum Umfragesystem

Test 1	
min=1 avg=1.82 max=4 dev=0.9	
Wie finden Sie das elektronische Bewertungssystem im Vergleich zu den Papierfragebögen?	1.0 hervorragend 19 *****
	2.0 10 *****
	3.0 10 *****
	4.0 1 *
	5.0 sehr schlecht 0
Warum?	
Teilweise konnten Sie zur Beantwortung der Fragen Radiobuttons verwenden und teilweise Auswahllisten. Welche Darstellungsart empfanden Sie als besser geeignet?	
	min=0 avg=0.3 max=1 dev=0.46
	0.0 Radiobuttons 26 *****
	1.0 Auswahlliste 11 *****
Haben Sie Verbesserungsvorschläge?	
Test 2	
min=1 avg=2.11 max=4 dev=0.96	
Wie finden Sie das elektronische Bewertungssystem im Vergleich zu den Papierfragebögen?	1.0 hervorragend 6 *****
	2.0 5 *****
	3.0 6 *****
	4.0 1 *
	5.0 sehr schlecht 0
Warum?	
Teilweise konnten Sie zur Beantwortung der Fragen Radiobuttons verwenden und teilweise Auswahllisten. Welche Darstellungsart empfanden Sie als besser geeignet?	
	min=0 avg=0.15 max=1 dev=0.38
	0.0 Radiobuttons 11 *****
	1.0 Auswahlliste 2 **
Haben Sie Verbesserungsvorschläge?	
Test 3	
min=1 avg=1.94 max=3 dev=0.77	
Wie finden Sie das elektronische Bewertungssystem im Vergleich zu den Papierfragebögen?	1.0 hervorragend 5 *****
	2.0 7 *****
	3.0 4 ****
	4.0 0
	5.0 sehr schlecht 0
Warum?	
Teilweise konnten Sie zur Beantwortung der Fragen Radiobuttons verwenden und teilweise Auswahllisten. Welche Darstellungsart empfanden Sie als besser geeignet?	
	min=0 avg=0.42 max=1 dev=0.51
	0.0 Radiobuttons 7 *****
	1.0 Auswahlliste 5 *****
Haben Sie Verbesserungsvorschläge?	

Abbildung 5.3: Ergebnis der Systembewertung

Evaluationsbogen für Modul Algorithmus und Datenstrukturen

Bitte füllt den Fragebogen nicht emotionsgeladen aus, das hilft uns nicht viel. Sondern beantwortet die Fragen bitte nach einer kurzen sachlichen Rückschau auf das **letzte Modul**.

Bei eurer Einstufung entspricht 1 = ja/sehr gut/ohne Probleme/einfach/voll und ganz, 2 ... 5 = den entsprechenden Zwischenwerten und 6 = nein/ungenügend/viele Probleme/schwierig/ganz und gar nicht entsprechend den Schulnoten. Oder das Bewertungsschema ist oberhalb der Kästchen explizit angegeben.

0. Eingruppierung

0.1 Ich bin Mathematiker: <input type="checkbox"/>	Ich bin Nicht-Mathematiker: <input type="checkbox"/>
--	--

I. Lehrmaterial in Textform (Folien-Skript)

	1	2	3	4	5	6
I.1 Hat Dich das Folien-Skript für den Stoff motiviert?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.3 Wie beurteilst Du die Verständlichkeit des Inhalts des Folien-Skripts?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.4.1 Wie beurteilst Du das Verhältnis $\frac{\text{Umfang der Materialien}}{\text{Schwierigkeit des Inhalts}}$ des Folien-Skripts?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.4.2 War die Darstellung des Folien-Skripts eher zu knapp oder eher zu ausführlich?	<small>eher knapp</small>	<input type="checkbox"/>	<small>...</small>	<input type="checkbox"/>	<small>eher zu ausführlich</small>	<input type="checkbox"/>

I. Vorlesung

	1	2	3	4	5	6
I.8 Während der Vorlesung habe ich bereits viel verstanden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.9 ... beim Durcharbeiten danach konnte ich alles gut verstehen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.10 Es wurden viele konkrete Beispiele behandelt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.11 Der Dozent/die Dozentin war gut vorbereitet.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.12 ... vermittelt Begeisterung für den Stoff.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.13 ... ermuntert zu Fragen während/nach der Vorlesung.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.14 ... konnte Fragen verständlich beantworten.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.15 Das Tempo der Vorlesung war nicht zu hoch.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.16 Die Vorlesung war übersichtlich gegliedert.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.17 Die Beweisideen bzw. Verfahren wurden klar dargestellt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.18 Es wurden genügend viele konkrete Beispiele dargestellt.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.19 Die Begriffe, Hauptresultate bzw. Verfahren wurden gut motiviert (z. B. durch Anwendungen oder historische Hinweise).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I.20 Die Folien waren hilfreich für den Lernprozeß.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.4: Fragebogen das SIGNAL-Projekts

III. Übungsbetrieb/Allgemein

- III.1 Wie beurteilst Du den gesamten **Aufwand** für das Modul in Stunden (á 60 Min.) pro Woche? (Durchschnittlicher wöchentlicher Aufwand für Skript, Übungsblätter, zusätzliche Informationen, Software, digitale Medien, Vorbereitung auf den Test), exklusiv der Präsenzzeit am Dienstag Stunden á 60 Minuten
- III.2 Hättest Du in diesem Modul lieber ... (Ü=Übungen) ...weniger Ü Ist ok ...mehr Ü
- | | | | | | | |
|--|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 | 6 |
- III.3 Waren die Übungen für Dich gewinnbringend?
- III.4 Wie fandest Du das Tempo der Übungen?
- III.5 Wie warst Du mit der Besprechung der Übungsblätter zufrieden?
- I.5 Wurden die Übungsblätter gut durch den Vorlesungsstoff vorbereitet?
- I.6 Wurde der Test gut durch die Übungsblätter vorbereitet?
-
- III.6 Wie warst Du mit der Betreuung durch den Lehrstuhl von Martin Hofmann zufrieden? (Organisation in München)
- III.7 Findest du das Vortragen der Lösung von einzelnen Aufgaben des Übungsblattes durch Teilnehmer gut?
- III.8 Hast Du sonst noch einen Kommentar zu diesem Modul? Was würdest Du den Tutoren zu diesem Modul empfehlen? (Freitext)

Ganz allgemein: Was hat Dir gut an dem Modul gefallen?

Im Detail: Was fandest Du besonders hilfreich (z.B. Beispiele, Übungen, digitale Medien)?

Ganz allgemein: Was hat Dir überhaupt nicht gefallen?

Im Detail: Was fandest Du nicht besonders hilfreich (z.B. Beispiele, Übungen, digitale Medien)?

Vorlesungsumfrage für die Veranstaltung SIGNALTI im Zeitraum Sommersemester 2005

Dozent: Prof. Kröger

Diese Umfrage soll dazu dienen, die Qualität der Lehre an der LMU München zu verbessern.
Bitte nehmen Sie sich ein paar Minuten Zeit, um diesen Fragebogen fair und differenziert auszufüllen.

Eingruppierung

Sind Sie Mathematiker?	<input type="text" value="1: Ich bin Nicht-Mathematiker"/>
------------------------	--

Lehrmaterial in Textform (Skript)

Hat Dich das Skript für den Stoff motiviert?	<input type="radio"/> 1 Ja <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Nein <input type="radio"/> Ich möchte keine Angaben machen
Wie beurteilst Du die Verständlichkeit des Inhalts des Skripts?	<input checked="" type="radio"/> 1 Sehr gut <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Ungenügend <input type="radio"/> Ich möchte keine Angaben machen
Wie beurteilst Du das Verhältnis von Umfang der Materialien zu Schwierigkeit des Inhalts im Skript?	<input type="radio"/> 1 Sehr gut <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Ungenügend <input type="radio"/> Ich möchte keine Angaben machen
War die Darstellung des Skripts eher zu knapp oder eher zu ausführlich?	<input checked="" type="radio"/> 1 Zu knapp <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 Zu ausführlich <input type="radio"/> Ich möchte keine Angaben machen

Vorlesung

Während der Vorlesung habe ich bereits viel verstanden.	<input type="text" value="1: Ja"/>
... beim Durcharbeiten danach konnte ich alles gut verstehen.	<input type="text" value="3:"/>
Es wurden viele konkrete Beispiele behandelt.	<input type="text" value="6: Nein"/>
Der Dozent/die Dozentin war gut vorbereitet.	<input type="text" value="2:"/>
...vermittelt Begeisterung für den Stoff.	<input type="text" value="4:"/>

Abbildung 5.6: Fragebogen unseres elektronischen Systems

Kapitel 6

Danksagung

Dem Lehrstuhl von Prof. Martin Hofmann gebührt mein großer Dank für die mir gebotene Gelegenheit, diese Diplomarbeit erstellen zu dürfen.

Insbesondere meinen Betreuern Martin Lange und Hans-Dietmar Jäger möchte ich für ihre Geduld und ihre entgegenkommende Betreuung danken.

Meine engen Freunde haben mich nicht nur während dieser Arbeit begleitet, sondern in manch heller Stunde Freude und in manch dunkler Stunde Trost gespendet.

Und schließlich stehe ich tief in der Schuld meiner Eltern, ohne die ich den Weg bis zu dieser Arbeit niemals hätte gehen können.

Danke!

Anhang A

Management und Erweiterung des Systems

A.1 Systemmanagement durch GUI-Applikation

Die Anwendung `de.itfuzzi.diplom.management.Gui` stellt eine grafische Oberfläche zum Erstellen und Bearbeiten von Fragebögen zur Verfügung und gestattet die Konfiguration für die beiden öffentlichen Schnittstellen `Umfrageservlet` und `AuswertungsServlet`.

Durch die Möglichkeit, die für das Web-Interface generierte XML-Darstellung des Fragebogens auch in der GUI-Schnittstelle zu erzeugen und zu exportieren, ist die Erstellung von Papierfragebögen aus dem elektronischen System durch ein geeignetes XSLT-Stylesheet machbar.

A.2 Schnittstellen zur Erweiterung

Die Erweiterung des Frameworks durch zusätzliche Handlerklassen ist aufgrund der in Abschnitt 4.2 dargestellten Mechanismen leicht möglich. Die hypothetische Handlerklasse `fuHand` muss hierzu das Interface `de.itfuzzi.diplom.core.handler.Handler` implementieren und zusätzlich eine statische Methode `getParser` zur Verfügung stellen, die eine Unterklasse von `de.itfuzzi.diplom.core.xmlconfig.HandlerParser` liefert.

Listing A.1: Beispielsignatur eines Handlers

```
public class fuHand implements Handler {  
    public static HandlerParser getParser();  
    publicTriState pre(FrameworkState tt);  
    publicTriState post(FrameworkState tt);  
}
```

A.3 Verwendete Entwicklungswerkzeuge

Java Version j2sdk1.4.1_02

Servlet Container Tomcat 5.5.0

IDE Eclipse 3.0.1

DAO-Generator DbModeller-0.3.2

Literaturverzeichnis

- [BrDa 03] BRITTAİN, JASON und IAN F. DARWIN: *Tomcat: the definitive guide*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, 2003.
- [Brow 02] BROWNELL, DAVID: *SAX2*. O'Reilly & Associates, Inc., 103a Morris Street, Sebastopol, CA 95472, USA, Tel: +1 707 829 0515, and 90 Sherman Street, Cambridge, MA 02140, USA, Tel: +1 617 354 5800, 2002, <http://safari.oreilly.com/0596002378>; <http://www.oreilly.com/catalog/sax2> .
- [Clar 99] CLARK, JAMES: *XSL Transformations (XSLT) Version 1.0*. <http://www.w3.org/TR/xslt>, November 1999.
- [Diekmann 95] DIEKMANN, ANDREAS: *Empirische Sozialforschung*. Rowohlt Taschenbuchverlag, ISBN 3-499-55551-4, 1995.
- [Kirchhoff 00] KIRCHHOFF, SABINE: *Machen wir doch einen Fragebogen*. Opladen : Leske + Budrich, 2000 ISBN 3-8100-2826-6, 2000.
- [Nic 00] NIC, MILOSLAV: *XSLT Tutorial*. <http://www.zvon.org/xxl/XSLTutorial/Output/index.html>, 2000.
- [Noelle-Naumann 96] ELISABETH NOELLE-NEUMANN, THOMAS PETERSEN: *Alle, nicht jeder : Einführung in die Methoden der Demoskopie*. München : Dt. Taschenbuch-Verl., 1996 ISBN 3-423-04688-0, 1996.
- [Scholl 03a] KROMREY, HELMUT: *Empirische Sozialforschung*. Verlag Leske + Budrich, Opladen, ISBN 3-8252-1040-5, 2003.
- [Scholl 03b] SCHOLL, ARMIN: *Die Befragung*. UVK Verlagsgesellschaft mbH, ISBN 3-8252-2413-9, 2003.
- [VS 96] VIPIN SAMAR, CHARLIE LAI: *Making Login Services Independent of Authentication Technologies*. <http://www.sun.com/software/solaris/pam/pam.external.pdf>, March 1996.

- [Welker 05] MARTIN WELKER, ANDREAS WERNER, JOACHIM SCHOLZ: *Online Research*. dpunkt.verlag, ISBN 3-89864-308-5, 2005.
- [Wott 03] WOTTAWA, HEINRICH / THIERAU, HEIKE: *Lehrbuch Evaluation*. Bern ; Göttingen ; Toronto ; Seattle : Hans Huber, 2003 ISBN 3-456-84051-9, 2003.