Studiengang:	Informatik
Prüfer:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Betreuer:	Dr. rer. nat. Cora Burger Dipl.-Inf. Michael Kinateder
begonnen am:	15. Oktober 2003
beendet am:	29. April 2004
CR-Klassifikation:	C.2.4, H.3.3, H.5.3, K.3.1

Studienarbeit Nr. 1912

Kollaborative Weiterentwicklung von Lernmaterial unter Verwendung eines Trust-Management-Systems

Reinhard Kuntz

Institut für
Parallele und Verteilte
Systeme
Abteilung Verteilte Systeme

Zusammenfassung

Untersucht wurden Anforderungen und Konzepte für ein System zur kollaborativen Weiterentwicklung von Lernmaterialien unter der Verwendung des Vertrauensmanagement-Systems UniTEC.

Entsprechend der zusammengestellten Anforderungen wurde nach einer Untersuchung in diesem Bereich bereits bestehender Systeme ein neues System entworfen. Mit diesem ist es Studierenden möglich, digitale Mitschriebe und Annotationen im direkten Bezug auf Dokumente zu erstellen und weiterzugeben.

Hierdurch lässt sich einerseits eine kollaborative Sicht auf die Lernmaterialien erzeugen, aus der jeder einzelne Nutzer eine für ihn optimierte Informationsmenge abrufen und zur Verbesserung seiner persönlichen Lernmaterialienzusammenstellung verwenden kann. Andererseits werden die Materialien auf der Basis – der zu Beginn einer Veranstaltung vom Dozenten zur Verfügung gestellten – Lernmaterialien auf kollaborative Weise weiterentwickelt, indem den Dozenten qualitativ hochwertige Rückmeldung bereitgestellt wird.

Die hierfür erforderlichen Mechanismen wurden unter der Verwendung des Vertrauensmanagement-Systems UniTEC entwickelt.

Inhaltsverzeichnis

1 Einleitung **1**

1.1 Motivation 1
1.2 Aufgabenstellung 2
1.3 Überblick 3
1.4 Anmerkung 3
1.5 Danksagung 4

2 Grundlagen **5**

2.1 Zielgruppe und Umfeld 5
2.2 Szenarien 7
2.2.1 Zusammenstellen des Stoffes 7
2.2.2 Studentische Vorbereitung 9
2.2.3 Lehrveranstaltung 9
2.2.4 Studentische Nachbereitung 10
2.2.5 Feedback 12
2.2.6 Bewältigung der Informationsmenge 13
2.2.7 Auslagern und Löschen von Daten 13
2.3 Informationsinhalt des Systems 14
2.3.1 Lernmaterialien 14
2.3.2 Systemspezifische Informationen 14
2.3.3 Organisatorische Informationen 14
2.3.4 Weiterverwendung von Informationen aus dem System 14

3 Anforderungen **16**

3.1 Bedienungsanforderungen 16
3.2 Verfügbarkeit und Verbreitung des Systems 17
3.3 Leistung und Modularität 17
3.4 Austauschplattform 18
3.5 Handhabung der Datenmenge 18
3.6 Handhabbarkeit von Bewertungen 19
3.7 Haftung für Informationen im System 19
3.8 Funktionale Anforderungen 20

4 Verwandte Arbeiten **28**
4.1 Lernplattformen . . . 28
4.1.1 BSCW, BSCL und Synergeia . . . 29
4.1.2 ILIAS . . . 30
4.1.3 metacoon . . . 31
4.1.4 eclaus . . . 31
4.2 Annotationssysteme . . . 31
4.2.1 ToGather . . . 32
4.2.2 Annotationserweiterung von Targeteam . . . 32
4.2.3 IAS Notizeditor . . . 33
4.3 Textverarbeitung/-visualisierung mit Annotationssystem . . . 33
4.3.1 SPP . . . 34
4.3.2 DoNut . . . 34
4.3.3 Acrobat Writer . . . 35
4.4 Informationsbereitstellungsdienste . . . 36
4.4.1 HTTP . . . 36
4.4.2 WIKI . . . 36
4.4.3 RSS und Blogs . . . 37
4.5 Zusammenfassung und Fazit . . . 37

5 Basissoftware **40**
5.1 Multivalent . . . 40
5.1.1 Architektur . . . 40
5.1.2 Funktionsumfang . . . 41
5.2 UniTEC . . . 43
5.2.1 Empfehlungsdienst . . . 44
5.2.2 Vertrauensmodell . . . 44
5.2.3 Identitätsmanagement . . . 45
5.2.4 Architektur . . . 46

6 Grobentwurf **47**
6.1 Grundlegendes . . . 47
6.2 Funktionale Gliederung . . . 48
6.3 Server . . . 50
6.4 Client . . . 52
6.4.1 Dokumenten-Viewer . . . 52
6.4.2 Bewertungs- und Vertrauens-Management . . . 52
6.4.3 Lokale Datenhaltung . . . 53
6.5 Kommunikation . . . 55
6.6 Ergebnisse . . . 57

7 Feinentwurf **58**
7.1 Dokumenten-Explorer . . . 58
7.2 Globale Eindeutigkeit . . . 59
7.3 Bewertungen . . . 60
7.4 Themenkategorien . . . 63
7.5 Vertrauen . . . 64
7.6 Synchronisation . . . 66
7.7 Filterung . . . 68
7.8 Anonymität der Nutzer . . . 69
7.9 Annotationen . . . 70

8 Realisierung **72**
8.1 Allgemeine Realisierungsaspekte . . . 72
8.2 Realisierung des Servers . . . 73
8.3 Realisierung des Clients . . . 74
8.3.1 Dokumenten-Explorer . . . 74
8.3.2 Dokument Transfer . . . 75
8.3.3 Annotationen . . . 75
8.3.4 Annotationstransfer . . . 78
8.3.5 Bewertungen und Vertrauensupdate . . . 79

9 Zusammenfassung und Ausblick **80**
9.1 Zusammenfassung . . . 80
9.2 Ausblick . . . 81
9.3 Fazit . . . 82

A Projektplan **83**
A.1 Einleitung . . . 83
A.2 Beschreibung der Arbeitspakete . . . 83
A.3 Definition der Arbeitspakete . . . 84
A.3.1 Abhängigkeiten zwischen den Arbeitspaketen . . . 85
A.3.2 Kritische Punkte/Risiken . . . 85
A.4 Zeitplan . . . 86
A.5 Dokumente und Meilensteine . . . 86
A.6 Weitere Anmerkungen . . . 87

B Anmerkungen zum Projektplan **88**

C Rechtliches **89**
C.1 Markenzeichen . . . 89
C.2 Software Lizenzen . . . 89

D Interface - ApplicationServerInt **90**

Kapitel 1

Einleitung

1.1 Motivation

Obwohl seit geraumer Zeit Lernmaterialien zu Vorlesungen im universitären Bereich in immer höherem Maße in digitaler Form zur Verfügung gestellt werden, werden diese meist in traditioneller Form verstanden und verwendet. Skripte und vermehrt auch angebotene Foliensätze werden ausgedruckt, in der Vorlesung handschriftlich mit Anmerkungen versehen und durch Beispiele oder weitere Informationen, die in der Vorlesung vorgetragen werden, ergänzt. Beim Nacharbeiten und in der Prüfungsvorbereitung werden wichtige Passagen markiert, Zusammenschriebe erstellt und Stichwortlisten zusammengestellt. Ebenfalls wird, wenn das Lernmaterial unzureichend ist, in sekundären Quellen recherchiert. Dabei wird oft das Internet herangezogen. Danach werden die Lernmaterialien mit den Ergebnissen der Recherchen ergänzt - zumeist wieder in Form von Ausdrucken, Kopien oder handschriftlichen Ergänzungen. Die so zusammengetragene und strukturierte persönliche Informationsmenge steht, wenn überhaupt, nur einem sehr kleinen Personenkreis zur Verfügung und ist für nachkommende Studierende nicht zugänglich.

Dozenten (siehe 2.1) müssen bisher die Weiterentwicklung von Lernmaterialien alleine bewerkstelligen. Das Weiterentwickeln gestaltet sich zudem oft sehr schwierig, da kaum Möglichkeiten vorhanden sind, Anregungen der Studierenden zu berücksichtigen. Dies liegt in erster Linie daran, dass zur Erweiterung bzw. Verbesserung der von den Dozenten zur Verfügung gestellten Lernmaterialien Feedback weder in ausreichender Granularität noch in erforderlicher Menge erzeugt werden. Außerdem sind diese Informationen, sofern vorhanden, den Dozenten nicht zugänglich.

Des weiteren stellt sich die Frage, inwieweit Studierende, wenn sie solch eine persönliche Informationsmenge von einem anderen Studierenden erhalten, auf die Richtigkeit und Vollständigkeit – also die Qualität – der Informationen vertrauen können. Selbst wenn dies gewährleistet ist, bedeutet das nicht, dass auch die persönlichen Anforderungen an die Informationsmenge erfüllt werden. Beispielsweise versteht der eine Student einen Sachverhalt schon anhand des Skriptes, der andere Student jedoch erst mit Hilfe eines Beispiels.

Der technische Fortschritt der letzten Jahre hat bewirkt, dass ein sehr großer Prozentsatz

der Studierenden einen Laptop oder ein anderes vergleichbares mobiles Endgerät besitzt. Mit diesen kann während der Vorlesung über meist vorhandene WLAN-Anschlüsse auf das Internet zugegriffen werden. Auch verfügen mittlerweile relativ viele Studierende über einen privaten Internetzugang, einige davon sogar über eine Flatrate. Weiter steht allen Studierenden die Möglichkeit offen, in Rechner-Pools auf das Internet zuzugreifen.

Es sind also alle Voraussetzungen gegeben, um den Prozess der Verteilung, Anpassung und Weiterentwicklung von Lernmaterialien rechnergestützt durchzuführen und dadurch zu ermöglichen, dass dieser sowohl ohne Medienbrüche als auch kollaborativ durchgeführt werden kann.

Mit UniTEC (siehe 1.4) steht außerdem ein Vertrauens-Management-System zu Verfügung, das helfen kann, die Frage nach der Qualität der Informationen zu beantworten und zu gewährleisten, dass die Informationsmenge an die persönlichen Anforderungen angepasst ist.

Die technische Ausstattung der Studierenden hat eine weitere Folge. Es entsteht immer häufiger der Wunsch, an unterschiedlichen Orten in kleinen Gruppen oder alleine zu lernen und trotzdem die Möglichkeit zu haben, Gedanken mit nicht anwesenden Kommilitonen oder sogar dem Dozenten auszutauschen, beispielsweise bei der Prüfungsvorbereitung, indem anderen Studierenden Lösungsansätze oder -wege zu Übungsaufgaben zur Verfügung gestellt werden, um darüber zu diskutieren und sie unter Umständen, bezüglich Vollständigkeit und Richtigkeit, gemeinsam zu ergänzen und zu validieren.

In dem geschilderten Bereich gibt es momentan jedoch leider nur sehr wenige Softwaresysteme. Diese befinden sich zudem meist in einem sehr frühen Entwicklungsstadium und sind mit einer anderen Zielsetzung entstanden. Deshalb sind sie im Bezug auf diese Studienarbeit nur sehr bedingt, wenn überhaupt, brauchbar. Außerdem befinden sich kostenpflichtige Systeme darunter.

1.2 Aufgabenstellung

Im Rahmen dieser Studienarbeit sollen die verschiedenen möglichen Formen von Lernmaterialien und Verknüpfungsmöglichkeiten zwischen solchen ermittelt werden. Für den Weiterentwicklungsprozess des Lernmaterials unumgängliche Konzepte sind zu untersuchen. Dabei sind die Möglichkeiten Diskussionen zu führen, Fragen zu stellen und Anmerkungen mit direktem Bezug zur kommentierten Stelle machen zu können, in Betracht zu ziehen.

Es ist weiter zu untersuchen, wie aus den entstehenden persönlichen Informationsmengen eine kollaborative Informationsmenge gebildet werden kann. Dabei soll der Fokus auf dem Modellieren eines Konzeptes liegen, das unter Verwendung des an der Universität Stuttgart entwickelten Vertrauens-Management-Systems UniTEC (siehe 1.4) die Problematik der Vertrauenswürdigkeit bzw. der Qualität der Informationen regelt. Insbesondere ist dafür eine geeignete Methode zur eindeutigen Identifikation des Lernmaterials zu entwickeln.

Außerdem soll natürlich auch darauf eingegangen werden, welche bestehenden Systeme

es schon gibt. Es soll untersucht werden, welche Konzepte oder Teilsysteme bezogen auf die zu erarbeitenden Anforderungen verwendet werden können. Auch ist in Betracht zu ziehen, welche untersuchten Systeme sich als Informationsquelle bezogen auf Lernmaterialien eignen und in wieweit es sinnvoll ist, diese einzubinden.

1.3 Überblick

Nach der Einleitung beschäftigt sich das zweite Kapitel mit den Grundlagen dieser Studienarbeit. Neben einer Betrachtung des Umfeldes wird ein umfassendes Szenario geschildert. Danach geht dieses Kapitel noch auf die Informationen, die in diesem Bereich anfallen, ein. Hierbei wird der Begriff *'Lernmaterial'* im Kontext dieser Arbeit definiert.

Im dritten Kapitel werden die Anforderungen anhand des Szenarios und weiterer Aspekte abgeleitet.

Daraufhin wird im vierten Kapitel untersucht, ob bisher bestehende Systeme den Anforderungen entsprechen oder zumindest zu Teilen verwendet werden können. Es wird sich zeigen, dass dies bei keinem der untersuchten Systeme der Fall ist. Daher wird der Entwurf eines neuen Systems erforderlich.

Deshalb werden im Kapitel 5 die als Basis zur Realisierung des Systems herangezogenen Softwaresysteme Multivalent und UniTEC vorgestellt und genauer beschrieben.

Darauf folgend zeigt das sechste Kapitel den Entwurf der grundlegenden Architektur des Systems. Hierfür werden zu Beginn des Kapitels die Funktionen, die vom System bereitgestellt werden sollen, in Gruppen gegliedert. Welche Funktionen von den im vorhergehenden Kapitel vorgestellten Systemen bereits abgedeckt werden, wird daraufhin ermittelt. Anhand welcher Konzepte das System realisiert werden soll, wird erläutert. Zum Schluss dieses Kapitels wird dann die grundlegende Architektur des Systems vorgestellt, die im Kern auf einer Client-Server-Architektur beruhen wird.

Das Kapitel 7 geht insbesondere auf den Entwurf des Clients ein, da der Server verhältnismäßig einfach zu realisieren sein wird. Dabei orientiert sich die Gliederung des Kapitels an den im vorhergehenden Kapitel gebildeten Funktionsgruppen.

Im vorletzten Kapitel wird geschildert, welche Teile des entworfenen Systems prototypisch realisiert werden. Kompromisse, die hierbei eingegangen werden, werden erläutert, und es wird gezeigt, warum manche Teile im Rahmen dieser Studienarbeit nicht implementiert werden können.

Abschließend werden die Ergebnisse dieser Studienarbeit im letzten Kapitel zusammengefasst. Auch wird darauf eingegangen, welche Maßnahmen noch erforderlich sind, um das System in einen produktreifen Zustand zu bringen. Mögliche Erweiterungen und erweiternde Konzepte werden außerdem angesprochen.

1.4 Anmerkung

Mit dem Thema dieser Arbeit und der Entwickung eines entsprechenden Systems ließe sich ohne Frage ein größeres Team mehrere Jahre beschäftigen. Deshalb ist klar, dass im

Rahmen dieser Studienarbeit und der damit stark begrenzten Zeit nicht auf alle Facetten tief eingegangen werden kann und Schwerpunkte gesetzt werden müssen. Auch darf nicht erwartet werden, dass ein produktreifes Softwaresystem entsteht.

Trotzdem wird in dieser Arbeit versucht, bei der Beschreibung der Szenarien, beim Entwickeln der Anforderungen, den Untersuchungen in 'Verwandte Arbeiten' sowie beim Entwurf des Systems nach Möglichkeit den gesamten Bereich zu betrachten und abzudecken. Erst bei der Realisierung bzw. der prototypischen Implementierung soll ein engerer Fokus auf die Kernfunktionalitäten gesetzt werden.

Dieses Vorgehen wurde gewählt, um den Kontext, in dem die Realisierung gesehen werden muss, besser verdeutlichen zu können. Außerdem soll damit erreicht werden, dass die Anforderungen sowie der Entwurf über das gesamte Gebiet durchgeführt werden, und nicht durch eine zu starke oder zu frühe Fokussierung Barrieren für eine eventuelle spätere Realisierung des gesamten Systems entstehen.

In der Aufgabenstellung dieser Arbeit ist die Verwendung von UniTEC zur Realisierung der Vertrauens-Management-Funktionalitäten nahegelegt. Aus diesem Grund wurde entschieden, keine Untersuchungen bezüglich etwaiger anderer Konzepte und Realisierungen dieser Funktionalitäten durchzuführen, was auch nicht Fokus dieser Arbeit ist. Weitere Informationen zu UniTEC können den Arbeiten [Ter03], [Aic03], [Wid03], [RK03] sowie [KP04] entnommen werden. In vorliegender Arbeit wird in 5.2 nochmals näher auf UniTEC eingegangen.

1.5 Danksagung

An dieser Stelle möchte ich mich bei meinen Betreuern, Dr. Cora Burger und Michael Kinateder bedanken, die mir mit kritischen Fragen, Anregungen, Hilfestellungen und durch Ermutigungen zur Seite standen.

Dr. Cora Burger will ich außerdem dafür danken, dass sie mir die Durchführung einer Studienarbeit in diesem Themengebiet ermöglichte. Sie initiierte die Aufgabenstellung, nachdem ich mit der Basisidee zu dieser Arbeit an sie herantrat.

Vielen Dank an alle, die mich beim Erstellen dieser Studienarbeit durch motivierenden Zuspruch unterstützt haben. Ein dickes Dankeschön auch an Christel Kuntz für das Korrekturlesen.

Kapitel 2

Grundlagen

Um Missverständnissen vorzubeugen, soll an dieser Stelle festgelegt werden, dass im Folgenden unter dem Begriff *'System'* das im Rahmen dieser Studienarbeit zu entwickelnde und zu Teilen prototypisch zu implementierende Softwaresystem (siehe 1.4) verstanden wird. Der Begriff *'System'* beinhaltet also lokal sowie zentral benötigte Softwareelemente.

In diesem Kapitel soll nun zunächst darauf eingegangen werden, wer zur Zielgruppe gehört und welches Umfeld für das System zu erwarten ist. Darauf folgt eine Beschreibung der gängigen Szenarien. Im Anschluss daran wird erläutert, wie der Begriff *'Lernmaterial'* im Kontext dieser Arbeit verstanden wird. Weiter wird darauf eingegangen, welche Daten insgesamt im System verarbeitet und verwaltet werden.

2.1 Zielgruppe und Umfeld

Wer die Zielgruppe des Systems bildet und welche Voraussetzungen für eine sinnvolle Nutzung des Systems bezogen auf die Zielgruppe erforderlich sind, soll in diesem Abschnitt ermittelt werden. Ob diese Anforderungen in ausreichender Weise erfüllt sind bzw. erfüllt werden können, wird im Anschluss daran diskutiert. Hierbei wird das Umfeld des Systems betrachtet.

Das System ist in erster Linie zur Unterstützung von Studierenden beim Durchführen ihrer täglichen Lernaufgaben gedacht, wobei in dieser Arbeit unter *'Studierenden'* die im Folgenden beschriebene Rolle verstanden wird: Studierende sind alle diejenigen, welche lernend an einer Veranstaltung teilnehmen. Dabei kann es sich um eine Vorlesung, eine Übungsveranstaltung oder ähnliches handeln. Dabei sollen bei der Verwendung des Begriffes *'Student'* selbstverständlich auch die weiblichen Studierenden mit inbegriffen sein.

Die Anzahl von Studierenden, die an einer Veranstaltung teilnehmen, variiert sehr stark. Im Falle einer großen Vorlesung kann eine Veranstaltung bis zu 600 Teilnehmer umfassen. Bei Übungsgruppen handelt es sich jedoch meist nur um einige wenige Teilnehmer.

Mit *'Dozent'* werden in dieser Arbeit alle diejenigen Personen bezeichnet, die in der lehrenden Rolle Veranstaltungen durchführen. Also jeder, der eine Veranstaltung, beispielsweise eine Vorlesung, abhält oder eine Übungsveranstaltung durchführt. Die sonst übliche

Unterscheidung zwischen Dozenten und Tutoren wird in dieser Arbeit also bewusst nicht gemacht. Auch soll in dieser Arbeit wie schon bei *'Student'* der Begriff *'Dozent'* der einfacheren Formulierungsmöglichkeiten wegen als geschlechtsneutral angesehen werden.

Eine Unterstützung seitens der Dozenten bezüglich des Systems ist unumgänglich, da nur sie die entsprechenden Informationen bzw. Lernmaterialien im System zur Verfügung stellen können. Damit alle Nutzer auf diese Daten zugreifen können, ist es nötig, dass das System eine Plattform zur Verfügung stellt, mittels der der Austausch von Lernmaterialien möglich ist. Diese Plattform soll im Folgenden *'Austauschplattform'* genannt werden.

Das System soll in einzelnen Fakultäten, aber auch universitätsweit betrieben werden können. Zugang zum System erhalten dann alle zu den entsprechenden Institutionen gehörenden Dozenten sowie die an deren Veranstaltungen teilnehmenden Studierenden. Dies soll über eine entsprechende Authentifizierung gegenüber dem System geschehen. Im Weiteren werden alle, die Zugang zum System haben, als *'Nutzer'* bezeichnet. Unter *'Betreiber'* wird im Folgenden die betreffende, das System zur Verfügung stellende Institution verstanden.

Es sollen nicht etwa nur Informatik Studierende, sondern Studierende aller Fachbereiche als Zielgruppe angesehen werden. Auch soll die Nutzung des Systems nicht auf den universitären Bereich begrenzt sein, da für Studierende anderer Bildungseinrichtungen der Nutzen der gleiche sein dürfte und sehr ähnliche Anforderungen vorliegen werden.

Um das System in einer Basisversion nutzen zu können, muss dem Nutzer mindestens ein Rechner zur Verfügung stehen, mit welchem zeitweise eine Verbindung zur Austauschplattform hergestellt werden kann. Dies kann beispielsweise durch einen PC mit temporärer Internetverbindung geschehen. Wie schon in 1.1 geschildert, werden diese minimalen Voraussetzungen von allen Studierenden erfüllt. Denn, falls die Studierenden keinen eigenen PC mit Internetzugang zur Verfügung haben, können sie zumindest an den Universitäten in Rechnerpools auf entsprechende Gerät mit Internetanbindung zurückgreifen.

Um das System direkt in einer Veranstaltung nutzen zu können, sind höhere technische Voraussetzungen erforderlich. Es wird ein mobiles Endgerät mit einem relativ großen Bildschirm benötigt, das die Lernmaterialien gut lesbar angezeigt. Auch muss das Endgerät eine gute und vor allem schnelle Eingabemöglichkeit aufweisen. Bei dem momentanen Stand der Technik kommen daher nur Laptops oder vergleichbare Geräte in Frage, also keine PDAs oder Mobiltelefone.

Die Verbreitung von Laptops im universitären Bereich, insbesondere in technischen Fächern, aber meist auch in nichttechnischen Fächern, ist mittlerweile relativ hoch und nimmt wahrscheinlich weiter zu. Daher soll hier angenommen werden, dass ausreichend vielen Studierenden Laptops zur Verfügung stehen, um genügend Mitschriebe mittels des Systems während der Veranstaltungen zu erzeugen. Das ist wichtig, da bei der Nutzung des Systems ohne mobilem Endgerät leider weiterhin ein Medienbruch vorhanden ist. Trotzdem soll weiter angenommen werden, dass das System auch für Studierende ohne mobilem Endgerät von hohem Nutzen sein wird, denn sie können mit Hilfe der Mitschriebe ihrer Kommilitonen mit Laptop das System zur Nacharbeit oder zur Prüfungsvorbereitung verwenden.

Leider bereitet beim momentanen Stand der Technik die Kapazität der Geräteakkus Probleme, wenn mehr als zwei Vorlesungen hintereinander besucht werden und in den Räumlichkeiten nicht ausreichend Steckdosen vorhanden sind. Auch stößt man an die Grenzen der Akkukapazitäten, wenn zwischen Veranstaltungen Standortwechsel von den Studierenden vorgenommen werden müssen und die Akkus der Laptops nicht geladen werden können. Dieses Problem kann derzeit nur durch einen oder mehrere Wechselakkus angegangen werden. Möglicherweise wird diese Problematik in Zukunft obsolet.

Während einer Veranstaltung Verbindung zur Austauschplattform - beispielsweise mittels WLAN - herstellen zu können, ist hilfreich und wünschenswert, für die Nutzung der Kernfunktionalitäten des Systems aber nicht erforderlich. Doch selbst in dieser Hinsicht kann angenommen werden, dass fast alle Studierende, die einen Laptop besitzen, auch mit einer WLAN-Karte ausgerüstet sind. An den Universitäten wurde außerdem in den letzten Jahren der Ausbau der WLAN-Netze vorangetrieben, so dass ein Austausch über die Plattform selbst während der Veranstaltungen an immer mehr Veranstaltungsorten erfolgen kann.

Bei der Nutzung des Systems mit einem Laptop entsteht kein Medienbruch. Auch werden die Studierenden durch das System darin unterstützt, sich gegenseitig zu helfen. Außerdem entsteht durch die auftretenden Synergie-Effekte ein relativ großer Mehrwert für die Nutzer des Systems. Deshalb besteht die Hoffnung, dass es akzeptiert werden wird, dass ein Laptop, also ein relativ großes und schweres sowie teures Gerät immer mit in die Veranstaltungen getragen werden muss. Auch kann dem Argument, dass es zu lästig sei, den Laptop immer mit sich führen zu müssen, entgegengehalten werden, dass das Mitnehmen des Laptops anstelle der Unterlagen in Papierform sowie der traditionellen Utensilien geschieht bzw. geschehen kann, und auf diese Weise kein größeres Gepäckvolumen zu erwarten ist.

2.2 Szenarien

Im Folgenden sollen nun die Szenarien geschildert werden. Unter einem *'Dokument'* werden Informationen verstanden, die als Datei vorliegen. Das können beispielsweise Skripte, Foliensätze, Zusammenfassungen, Übungsaufgaben (-blätter) und Lösungsvorschläge in Form von pdf, doc, ppt, txt oder ähnlichen Dateiformaten sein. Unter dem Begriff *'Annotation'* soll eine Notiz, Anmerkung, Erklärung oder Frage verstanden werden. Dabei soll sich diese direkt auf eine Stelle eines Dokumentes beziehen, den ursprünglichen Inhalt des Dokumentes jedoch unverändert lassen.

2.2.1 Zusammenstellen des Stoffes

Der Dozent wird durch das Verwenden des Systems wie in 2.2.5 beschrieben beim Weiterentwickeln der Lernmaterialien unterstützt. Der Aufwand für den Dozenten beim Verwenden des Systems ist ähnlich dem bei traditionellen Methoden, Lernmaterialien den Studiernden zur Verfügung zu stellen. Deshalb erzeugt der Dozent eine neue Instanz im System, wenn er in naher Zukunft eine Veranstaltung durchführen möchte. Das Erzeugen

einer solchen Veranstaltungs-Instanz soll Dozenten vorbehalten sein.

Öffentliche / Allgemeine Daten Damit organisatorische Daten nicht mittels anderer Systeme oder Methoden veröffentlicht werden müssen, soll es möglich sein, diese mittels des Systems zu verwalten. Dazu gibt der Dozent beim Erstellen einer Veranstaltung zunächst Daten, wie den Namen, den Durchführungszeitraum und den Durchführungsort ein. Weiter sind Eingaben wie Hinweise auf Begleitveranstaltungen, beispielsweise Übungen, wer diese durchführt, wo und zu welcher Zeit sie stattfinden.

Außerdem teilt er eine Beschreibung zum Inhalt und Zweck der Veranstaltung mit. Auch welches Angebot besteht, den Stoff prüfen zu lassen und wie man sich dazu anmelden kann, wann und wo die Prüfung stattfindet, wird hier angegeben. Nun muss der Dozent die Veranstaltung noch einer Themenkategorie zuordnen. Näheres dazu unter 2.2.6.

Zugangsbeschränkung Der Dozent kann wählen, ob die Begleit- bzw. Lernmaterialien, die er zu seiner Veranstaltung zur Verfügung stellt, allgemein zugänglich oder nur für seine Hörerschaft erreichbar sein sollen. Das heißt, er kann ein Zugangspasswort für diese Veranstaltung vergeben. Dieses muss er dann in seiner Veranstaltung bekannt geben.

Erstmalige Veranstaltung Führt ein Dozent eine Veranstaltung zu einem Themengebiet zum ersten Mal durch oder hat er dies bisher ohne Verwenden des Systems getan, kann er nicht auf Ergebnisse aus dem System zurückgreifen. Er wird also, sofern vorhanden, Dokumente, wie beispielsweise ein vorlesungsbegleitendes Skript oder Vortragsfolien, Übungsaufgaben sowie alte Prüfungsaufgaben, eventuell mit Musterlösungen, im System veröffentlichen. Möchte er seiner Hörerschaft Literatur passend zur Veranstaltung empfehlen, so kann er dies ebenfalls mit Hilfe des Systems tun, indem er diese in das Literaturverzeichnis einträgt. Näheres hierzu unter 2.2.4, Abschnitt 'Recherche ausserhalb des Systems / Literaturverzeichnis'.

Der Dozent hat jederzeit die Möglichkeit, in regelmässigen Abständen weitere Informationen zur Verfügung zu stellen, beispielsweise fortschreitend die aktualisierten Kapitel des Skriptes oder die Vortragsfolien für den nächsten Termin oder auch Übungsaufgaben. Dies gilt auch während der Veranstaltung, sofern dem Dozenten ein mobiles Endgerät und ein Netzwerkzugang zur Verfügung stehen. So kann der Dozent, wenn er zum Beispiel schneller vorankam als geplant, die Veröffentlichung des weiterführenden Foliensatzes vorziehen.

Wiederholungsveranstaltung / Wiederverwendung Hat der Dozent die Veranstaltung schon einmal mit Verwendung des Systems abgehalten, so kann er den Studierenden die dabei entstandene Informationsmenge wieder zur Verfügung stellen. Hierbei steht ihm frei zu wählen, welche der entstandenen Dokumente und Annotationen oder weiteren Informationen übernommen werden sollen. Möchte er zum Beispiel ein Dokument und einen Teil der dazu entstandenen Annotationen übernehmen, kann er mittels der Bewertungen (siehe 2.2.6) eine Teilmenge definieren und diese in die neue Veranstaltung importieren. Auch hat er natürlich die Möglichkeit mittels der Informationen aus den besonders gut bewerteten Annotationen, Dokumente zu überarbeiten. Diese Neuauflagen der Dokumente

kann der Dozent dann anstelle oder auch zusätzlich zu den alten Fassungen zur Verfügung stellen.

2.2.2 Studentische Vorbereitung

Veranstaltungs- und Stoffübersicht Studierende können sich mit Hilfe der öffentlichen bzw. allgemeinen Daten der Veranstaltungen alle angebotenen Veranstaltungen anzeigen lassen und diejenigen heraussuchen, welche sie besuchen möchten.

Haben die Studierenden eine Auswahl getroffen, werden sie sich einen Überblick über den Stoff der einzelnen Veranstaltungen verschaffen. Deshalb werden sie sich anzeigen lassen, welche Informationen zu der jeweiligen Veranstaltung im System in Form von Dokumenten und Annotationen schon vorhanden sind. Der Studierende lädt also von der Austauschplattform alle zu der relevanten Veranstaltung vorhandenen Informationen herunter. Er sichtet diese Materialien zur Vorbereitung auf die Lehrveranstaltung und macht eventuell erste eigene Annotationen.

Abonnement Die Metadaten von auf der Austauschplattform neu hinzukommenden Dokumenten werden automatisch von der Austauschplattform heruntergeladen und den Nutzern angezeigt. Die Dokumente selbst werden jedoch erst heruntergeladen, wenn sie mit Hilfe der Metadaten manuell ausgewählt werden und das Dokument angefordert wird. Dieser Vorgang kann automatisiert werden. Dies geschieht, indem der Nutzer für eine Vorlesung ein sogenanntes Dokumentenabonnement vereinbart. Ist ein solches vom Nutzer initiiert, wird für die jeweiligen Veranstaltungen zukünftig immer, wenn eine Verbindung zur Austauschplattform besteht, geprüft, ob es neue Dokumente zu dieser gibt. Ist das der Fall, werden diese automatisch heruntergeladen. Das Erstellen eines Dokumentenabonnements geschieht durch Auswählen einer Veranstaltung und darauffolgendes Anwählen der Abonnementenfunktion im Menü der System-Oberfläche. Auf dieselbe Weise können die Studierenden ein solches Abonnement auch wieder auflösen.

Möchte der Studierende Annotationen anderer Nutzer zu einem Dokument mit betrachten, so kann er ein Annotationsabonnement einrichten. Das Annotationsabonnement bewirkt, dass zu dem jeweiligen Dokument, immer wenn eine Verbindung zur Austauschplattform hergestellt ist, lokal noch nicht vorhandene Annotationen geladen werden. Dabei kann gewählt werden, ob alle Annotationen oder eine anhand Filterung (siehe 2.2.6) ermittelte Teilmenge aller Annotationen geladen werden sollen.

2.2.3 Lehrveranstaltung

Mitschreiben / Fragen stellen Studierende mit mobilem Endgerät sollen in Veranstaltungen mit Hilfe des Systems mitschreiben können. Das heißt, sie lassen sich das momentan relevante Dokument, also beispielsweise den aktuellen Foliensatz oder das vorlesungsbegleitende Skriptum anzeigen und verfolgen damit die Vorlesung. Sind sie der Meinung, dass sie sich Informationen oder Erklärungen des Dozenten notieren sollten, fügen sie an den entsprechenden Stellen des Dokumentes eine Annotationen ein. Ist ihnen

etwas unklar, verfahren sie genauso, klassifizieren jedoch die Annotationen zusätzlich als Frage.

Die auf diese Weise neu erstellten Annotationen sind vorerst nur dem Ersteller zugänglich. Ist dieser der Meinung, dass die erstellten Annotationen für andere Nutzer von Bedeutung sein könnten, veröffentlicht er diese. Sobald eine Verbindung zur Austauschplattform besteht, werden veröffentlichte Annotationen, sofern noch nicht geschehen, auf die Plattform geladen.

Skriptstellensuche Steht den Studierenden nur ein vorlesungsbegleitendes Skript zur Verfügung, so kann es nötig sein, die richtige Skriptstelle zum aktuell behandelten Stoff schnell zu suchen. Beispielsweise, wenn die Reihenfolge der Vorlesung nicht mit dem Skript abgestimmt ist. Hierfür kann eine Volltextsuche auf dem Dokument genutzt werden.

2.2.4 Studentische Nachbereitung

Sichten Beim Nacharbeiten gehen die Studierenden ihre Unterlagen noch einmal durch. Sie lesen die Dokumente und bekommen dabei die Annotationen, die sich in der persönlichen Sicht befinden, angezeigt. Unter persönlicher Sicht soll hier ein Darstellungs-Modus verstanden werden. Normalerweise arbeiten Nutzer in diesem Modus. Es wird hierbei eine kleine Auswahl von Annotationen angezeigt, die durch Erstellen eigener oder explizites Hinzufügen fremder Annotationen zustande kommt.

Ist Studierenden etwas unklar, so lassen sie sich die verfügbaren Annotationen anderer Nutzer anzeigen. Dies soll möglich sein, indem der im Folgenden mit *'nicht persönliche Sicht'* bezeichneten Anzeige-Modus gewählt wird. In diesem Modus werden entweder alle lokal verfügbaren Annotationen oder eine mit Hilfe von Filterungsmöglichkeiten bestimmte Teilmenge davon angezeigt. Die Studierenden lesen die Annotationen der anderen Nutzer und wählen die für sie hilfreichen Annotationen aus. Durch das Auswählen werden die Annotationen zur persönlichen Sicht hinzugefügt und dadurch beim zukünftigen Arbeiten immer mit angezeigt.

Selbstverständlich haben die Studierenden auch die Möglichkeit, Beiträge aus der persönlichen Sicht zu entfernen. Dies wird geschehen, wenn beispielsweise festgestellt wird, dass ein Fehler in einer Annotation ist und die korrigierte Information notiert werden soll. Oder wenn ein Nutzer eine besser formulierte Erklärung von einem anderen Studierenden findet.

Wenn ein Beitrag zur persönlichen Sicht hinzugefügt oder aus dieser entfernt wird, bewertet der Studierende diesen. Das Bewerten ist notwendig, damit durch die Bewertungen die Bildung der Vertrauenswerte und somit die Filterungsmöglichkeiten, die oben schon angesprochen wurden, immer effektiver und persönlicher werden. Weiteres zu Bewertungen, Vertrauenswerten und Filterung unter 2.2.6.

Recherche außerhalb des Systems / Literaturverzeichnis Finden Studierende unter den Annotationen der anderen Nutzer keine hilfreichen Informationen, so werden sie mit Recherchen in anderen Quellen versuchen, ihre Fragen zu beantworten. Finden sie

Informationen, die ihre Fragen beantworten, so geben sie diese als Annotation an den fraglichen Dokumentenstellen ein. Um Urheberrechte nicht zu verletzen, müssen sie Quelleninformationen angeben. Diese werden beim Eingeben in das Literaturverzeichnis (siehe 2.2.1 Abschnitt 'Öffentliche / Allgemeine Daten') aufgenommen. Im Literaturverzeichnis kann dementsprechend auch nach externen Informationsquellen zum Thema der Veranstaltung gesucht werden. So ist es jedem Nutzer auch möglich, wenn er externe Literatur empfehlen möchte, unabhängig von Annotationen diese in das Literaturverzeichnis einzutragen.

Fragen stellen und beantworten Finden Studierende zu einer bestimmten Frage keine diese beantwortenden Informationen, so geben sie an der entsprechenden Dokumentenstelle diese Fragen ein (siehe 2.2.3 Abschnitt 'Mitschreiben / Fragen stellen') und veröffentlichen sie.

Ist eine auf diese Weise neu veröffentlichte Frage zu einem Dokument vorhanden, wird diese als Symbol am Rand des Skriptes bei allen Nutzern angezeigt. Hierdurch kann jeder beim Arbeiten am Skript sehen, dass jemand eine Frage zu diesem Teil gestellt hat. Die Studierenden können so prüfen, ob sie die Frage beantworten können. Wenn das der Fall ist, haben sie die Möglichkeit, eine Antwort an die Frage anzuhängen. Stellen die Studierenden jedoch fest, dass sie an dieser Stelle ebenfalls Defizite haben, können sie die Diskussion weiterverfolgen, indem sie die Frage zu ihrer persönlichen Sicht hinzufügen. Denken die Studierenden, dass ihre gestellten Fragen beantwortet sind, bewerten (siehe 2.2.6) sie die jeweils zugehörige *'Diskussion'* und markieren dadurch die entsprechende Frage als für sie beantwortet. Nehmen die Nutzer eine Frage nicht in ihre persönliche Sicht auf, so wird sie, sobald sie vom Ersteller als beantwortet markiert wurde, nicht mehr angezeigt.

Nicht nur im Dokument an der entsprechenden Stelle können die Nutzer auf Fragen und die darauf folgenden *'Diskussionen'* Bezug nehmen. Dies soll auch mit Hilfe des Fragenverzeichnisses möglich sein. Das Fragenverzeichnis stellt hierbei eine hierarchische Auflistung der Fragen und der dazugehörigen Antworten dar. Die Darstellung des Fragenverzeichnisses soll der Visualisierung von im Internet üblichen Diskussionsforen entsprechen. Die Fragen werden also ihrem Status gemäß als gelesen bzw. ungelesen markiert angezeigt. Hierdurch soll den Nutzern eine Übersicht gegeben werden, die es ihnen einfach ermöglicht, nachzusehen, ob zu ihren Fragen neue Antworten gegeben wurden.

Eine Frage kann also beantwortet werden, indem im Fragenverzeichnis oder in einem Dokument eine solche ausgewählt und mit Hilfe einer Annotation dazu Stellung bezogen wird. Auch können die Studierenden anstatt eine Annotation einzugeben, eine Verknüpfung (siehe nächster Abschnitt) zu einem Dokument oder einer Textstelle in einem Dokument angeben. Das ist wichtig, wenn sie beispielsweise die Meinung vertreten, dass die Frage in einem Dokument ausreichend erklärt ist. Auch ist es hilfreich, wenn sie ein Beispiel erstellt und als Dokument im System hinzugefügt und veröffentlicht haben (siehe Abschnitt 'Dokumente veröffentlichen / Übungen'), und auf diese Weise darauf verweisen wollen.

Dokumente verknüpfen Gibt es in einer Veranstaltung ein begleitendes Skript sowie Foliensätze und möchten Studierende in der Veranstaltung an den Folien ihre An-

merkungen machen, so ist es erforderlich, dass Verknüpfungen zwischen den Dokumenten erstellt werden können, um beim Nacharbeiten eine schnelle Navigation zu ermöglich. Also beispielsweise Parallelstellen im Skript zu den korrespondierenden Stellen in den Folien verknüpft werden können.

Verknüpfungen werden auf dem Dokument als Symbol oder durch farbiges Unterlegen eines Textabschnittes aus dem Dokument dargestellt. Die Verknüpfungen sollen benennbar sein. Sie sollen auf Positionen innerhalb des Dokumentes, aber auch auf Positionen in anderen Dokumenten verweisen können. Klickt ein Nutzer eine Verknüpfung an, soll ähnlich wie bei Hyperlinks das entsprechende Ziel der Verknüpfung angezeigt werden. Die Navigation entlang bereits getätigter Verknüpfungssprünge soll vorwärts sowie rückwärts möglich sein.

Dokumente veröffentlichen / Übungen Erstellen Studierende zum Beispiel einen Zusammenschrieb oder eine Formelsammlung, so sollen sie diese Dokumente wie der Dozent jederzeit im System veröffentlichen können. Auch wenn zu Übungen Lösungen erstellt und abgegeben werden sollen, kann dies geschehen, indem die Lösungsabgaben als Dokumente in das System gestellt und veröffentlicht werden.

2.2.5 Feedback

Laufender Wissensstand Wie die Studierenden kann der Dozent sich ebenfalls die Fragen – beispielsweise zu dem beim letzten Veranstaltungstermin behandelten Stoff – ansehen. Auf diese Weise kann er sich ein Bild davon machen, ob der Stoff komplett verstanden wurde oder ob er noch einmal auf Teile davon eingehen sollte. Hat er den Eindruck, dass eine Frage elementar wichtig oder prüfungsrelevant ist, kann er diese beim folgenden Veranstaltungstermin aufgreifen und ansprechen. Geht eine Frage auf einen Bereich vertiefend ein, ist jedoch nicht prüfungsrelevant, so wird der Dozent diese sicherlich nicht in der Veranstaltung ansprechen und für alle verbal erläutern, sondern nur mit Hilfe des Systems beantworten.

Feedback zur beendeten Veranstaltung Wie bereits in 2.2.1 im Abschnitt 'Wiederholungsveranstaltung / Wiederverwendung' geschildert, kann der Dozent die bei der Durchführung einer Veranstaltung entstehenden Informationen nutzen bzw. wiederverwenden. Hierfür kann er sich zum Beispiel Dokumententeile, zu welchen besonders viele hoch bewertete Fragen erstellt wurden, ansehen. Dadurch erhält er ein quantitatives und qualitatives Feedback. Er bekommt implizit Anregungen zum Verbessern seiner Dokumente. Zum Beispiel, wenn Studierende untereinander Fragen anhand von selbst erstellten Beispielen beantworten, oder auch, wenn Sachverhalte mittels Formulierungen erklärt werden, die der Dozent bisher nicht verwendet hat. Das Fragenverzeichnis, wie oben schon beschrieben, kann der Dozent sich in diesem Sinne ebenfalls zu Nutze machen.

2.2.6 Bewältigung der Informationsmenge

Filterung Wie in 2.2.4 im Abschnitt 'Sichten' angesprochen, soll es möglich sein, die Annotationen anderer Nutzer gefiltert anzuzeigen. Das ist erforderlich, damit die Nutzer nicht eine riesige Menge von Annotationen nach den für sie interessanten Informationen durchsuchen müssen. Deshalb soll eine Filterungsmethode verwendet werden, mit deren Hilfe eine Teilmenge aller verfügbaren Annotationen gebildet wird, die möglichst gut auf die Bedürfnisse der einzelnen Nutzer zugeschnitten ist. Hierfür sollen Bewertungen und Vertrauenswerte herangezogen werden (siehe folgende Abschnitte).

Bewertungen abgeben Bewertungen müssen, wie schon beschrieben, beim Hinzufügen und Entfernen von Annotationen zur oder von der persönlichen Sicht abgegeben werden. Auf freiwilliger Basis soll es weiter möglich sein, für jedes Dokument sowie jede Annotation eine Bewertung abzugeben. Gefällt einem eine Zusammenfassung oder eine Diskussion sehr gut, so bewertet der Student diese entsprechend gut. Ist er der Meinung, dass eine Frage beispielsweise völlig fehl am Platz ist, so kann er diese dementsprechend negativ bewerten.

Vertrauensbildung Es kann allerdings sein, dass eine Annotation von manchen Studierenden als sehr gut und von anderen als nicht so gut empfunden wird, obwohl sie korrekt und passend ist. Beispielsweise weil die Annotation von manchen Studierenden als zu knapp formuliert angesehen wird. Andere jedoch der Meinung sind, dass der Sachverhalt ausreichend ausführlich notiert wurde. Bewertungen sind also subjektiv und reichen deshalb für ein auf den einzelnen Nutzer optimiertes Filtern nicht aus.

Aus diesem Grund sollen zusätzlich Vertrauenswerte gegenüber den anderen Nutzern gebildet werden, mittels derer das Filtern auf die persönlichen Bedürfnisse der Nutzer abgestimmt wird. Die Vertrauenswerte sollen mittels Bewertungsvergleichen und mit Hilfe von Algorithmen, welche möglichst 'naturgetreu' Vertrauen abbilden, erzeugt werden. Dabei wird ein Vertrauenswert immer auf eine Themenkategorie und einen Nutzer bezogen erzeugt. Deshalb ist das Zuordnen von Veranstaltungen zu Themenkategorien, wie in 2.2.1 Abschnitt 'Öffentliche / Allgemeine Daten' angesprochen, erforderlich.

2.2.7 Auslagern und Löschen von Daten

Nach Abschluss einer Veranstaltung möchte der Nutzer, da er die Informationen nicht mehr ständig benötigt, den durch diese belegten Festplattenspeicher wieder freigeben. Um jedoch später auf die Informationen wieder zurückgreifen zu können, sollen die Informationsdaten nicht einfach nur gelöscht werden. Es soll möglich sein die Daten auszulagern, beispielsweise indem sie auf eine CD oder DVD gebrannt werden. Auf ausgelagerte Informationen soll dann mittels des Systems jederzeit wieder zugegriffen werden können. Selbstverständlich soll es auch möglich sein, die Informationsdaten einer Veranstaltung auf der Festplatte komplett zu löschen anstatt sie auszulagern.

2.3 Informationsinhalt des Systems

2.3.1 Lernmaterialien

Wie schon im Abschnitt 'Motivation' beschrieben, werden bisher unter dem Begriff 'Lernmaterial' meist nur, sofern vorhanden, das veranstaltungsbegleitende Skript, Foliensätze sowie Übungsaufgaben verstanden. Aus den Szenarien ist ersichtlich, dass der Begriff 'Lernmaterial' in dieser Arbeit auf andere Weise verstanden wird. Zum einen werden Lernmaterialien hier nicht als in traditioneller Form, auf Papier vorliegend, betrachtet, sondern in elektronischer Form, im System befindlich. Zum anderen sollen nicht nur veranstaltungsbegleitende Skripten und Foliensätze als Lernmaterial verstanden werden, sondern auch die mit Hilfe des Systems gemachten Annotationen, also Fragen und Anmerkungen, und die mittels dieser erstellten Diskussionen. Außerdem sollen hier Literaturhinweise, Übungsaufgaben sowie alte Klausuren einschließlich zugehöriger Musterlösungen sowie studentische Lösungsvorschläge und Zusammenschriebe oder Formelsammlungen zum Lernmaterial gehören.

2.3.2 Systemspezifische Informationen

Wie schon in den Szenarien geschildert, soll jeder einzelne Nutzer eine auf ihn persönlich abgestimmte Teilmenge bzw. Sicht der allgemein im System zur Verfügung stehenden Lernmaterialien erhalten. Um dies zu bewerkstelligen sind die systemspezifischen Informationen erforderlich. Zu diesen gehören die Bewertungsinformationen und die Vertrauenswerte sowie die Informationen über die Themenkategorien und ihre Abhängigkeiten untereinander. Auch die Daten, welche Veranstaltung zu welcher Themenkategorie gehört, sollen hier dazu gezählt werden. Nähere Informationen zu Bewertungen, Vertrauenswerten und Themenkategorien finden sich im Abschnitt 'Entwurf'.

2.3.3 Organisatorische Informationen

Neben den Lernmaterialien und den systemspezifischen Daten werden im System organisatorische Daten gehalten. Es handelt sich um organisatorische Informationen wie Veranstaltungstermine und Veranstaltungsorte, dazugehörige Übungsveranstaltungen, Prüfungstermine und Ähnliches wie in 2.2.1 Abschnitt 'Öffentliche / allgemeine Daten' beschrieben. Dies soll bewirken, dass die Studierenden sich einfacher über den Veranstaltungsumfang informieren können. Auch soll damit bezweckt werden, dass alle von den Studierenden benötigten organisatorischen und terminlichen Informationen schnell und einfach aufgefunden werden können und nicht zusätzlich andere Informationssysteme hierfür herangezogen werden müssen.

2.3.4 Weiterverwendung von Informationen aus dem System

Es stellen sich einige rechtliche Fragen bezüglich der Weiterverwendung der kollaborativ entstehenden Informationen im System. Es werden hier einige beispielhafte Fragen gestellt,

um auf diese Problematik aufmerksam zu machen. Jedoch kann im Rahmen dieser Studienarbeit nicht näher auf dieses Thema eingegangen werden, da dies den zeitlichen Rahmen sprengen würde. Außerdem ist dies nicht die Domäne der Informatik und soll den entsprechenden Fachleuten überlassen werden.

Die Fragen:

- Hat der Dozent generell das Recht, die von den anderen Nutzern hinzugefügten Informationen dafür zu verwenden, sein Skriptum zu überarbeiten?
- Möchte der Dozent ein Lehrbuch oder eine andere kommerzielle Publikation veröffentlichen und verwendet er dafür mit Hilfe des Systems über eine längere Zeit zusammengetragene und verbesserte Daten, muss er dann alle Nutzer des Systems bezüglich seiner Veranstaltung als Quelle angeben, oder hat er als Durchführender der Veranstaltung implizit rechtliche Ansprüche auf die Informationen, die begleitend zu seiner Veranstaltung im System entstehen?
- Auch stellen sich ähnliche Fragen, wenn ein Student beispielsweise Informationen aus dem System in einer Veröffentlichung oder beispielsweise einer Studien- oder Diplomarbeit verwenden möchte. Darf er dies und wie muss er die Quelle angeben?
- Können Mitnutzer des Systems rechtliche bzw. finanzielle Ansprüche geltend machen, wenn eine kommerzielle Publikation veröffentlicht wird, die Daten aus dem System zur Grundlage hat?

Kapitel 3

Anforderungen

Im vorangegangenen Kapitel wurden die Szenarien geschildert und die Zielgruppe sowie das Umfeld des Systems beschrieben. Daraus sollen nun in diesem Kapitel die Systemanforderungen erarbeitet werden. Zunächst sollen dabei die Bedienungsanforderungen und Anforderungen, die sich durch die Zielgruppe und das Umfeld des Systems ergeben, behandelt werden. Nach dem Betrachten von Anforderungen an die Leistung des Systems werden zum Schluss dieses Kapitels die funktionalen Anforderungen aufgeführt und deren Ableitung aus den Szenarien verdeutlicht.

3.1 Bedienungsanforderungen

Wie in 2.1 schon beschrieben, ist das System an Studierende aller Fachbereiche gerichtet. Das bedeutet, dass auch mit nicht IT erfahrenen Anwendern zu rechnen ist. Um insbesondere für diese die Hemmschwelle zur Nutzung des Systems möglichst gering zu halten, sollte der Aufwand der Einarbeitung in die Bedienung des Systems möglichst gering ausfallen. Deshalb ist eine sehr einfache, schnelle und intuitive Bedienbarkeit erforderlich.

Dies ist außerdem notwendig, da das System in Veranstaltungen wie beispielsweise Vorlesungen bedient werden soll, wie schon in 2.2.3 beschrieben.

Die Funktionalitäten zum Auswählen, Herunterladen, Veröffentlichen, sowie Bewerten sollen außerdem in Bedienung und Erscheinungsbild in von anderen Softwaresystemen vertrauter Weise zugänglich sein. Hierdurch soll die Bedienung des Systems ebenfalls erleichtert werden und der Angst, mit einem neuen System nicht zurechtzukommen, entgegengetreten werden.

Der Aufwand, Dokumente im System zur Verfügung zu stellen, muss außerdem sehr gering gehalten werden, da das System ansonsten von den Dozenten, wie schon in 2.1 erläutert, nicht angenommen wird. Das bedeutet, dass das Erstellen und Veröffentlichen von Veranstaltungen, Hinzufügen von Dokumenten und Veröffentlichen dieser sehr einfach und schnell bedienbar sein muss. Auch ist hinsichtlich der Dozenten ebenfalls ein Integrieren dieser Funktionalitäten in eine vertraute Umgebung wichtig.

3.2 Verfügbarkeit und Verbreitung des Systems

Die Verfügbarkeit des Systems innerhalb einer Bildungseinrichtung ist sehr wichtig, da nur hohe Synergieeffekte erzielt werden können, wenn das System von möglichst vielen Nutzern verwendet wird. Daher muss darauf geachtet werden, dass nicht von vornherein Teile der potenziellen Nutzer ausgeschlossen werden.

Um dies zu erreichen, muss das System neben den im vorangegangenen Abschnitt geschilderten Anforderungen bezüglich der Bedienung plattformunabhängig sein. Auch sollen Nutzer, die keine Flatrate besitzen und nicht in Rechnerpools arbeiten wollen, nicht über den Kostenfaktor eines normalen Internetzuganges ausgeschlossen werden. Daher ist es erforderlich, dass auch offline das Arbeiten mit dem System möglich ist, selbstverständlich unter zeitweiser Einwahl mit dem Modem, um einen Informationsabgleich mit der Austauschplattform vorzunehmen. Dies ist außerdem erforderlich, weil nicht davon ausgegangen werden kann, dass an jedem Veranstaltungsort eine Verbindung zur Austauschplattform hergestellt werden kann. Die Möglichkeit, offline zu arbeiten, ist außerdem von Vorteil, da dann beispielsweise mit Laptops auch an Orten ohne Internetzugang, wie zum Beispiel unterwegs im Zug, mit dem System gearbeitet werden kann.

Das System sollte kostenlos zur Verfügung stehen, da Studierende kaum gewillt, wenn überhaupt fähig sein werden, das System kostenpflichtig zu erwerben. Es empfiehlt sich daher, das System als Free Ware zur Verfügung zu stellen.

Um eine hohe Verbreitung des Systems zu erreichen, wäre es wünschenswert, das System an anderen Bildungseinrichtungen verwenden und weiterentwickeln bzw. anpassen zu können. Daher ist eine Open Source - Lizenz für das System wünschenswert.

3.3 Leistung und Modularität

Wie in 2.2.3 verdeutlicht, soll das System im laufenden Betrieb von Lehrveranstaltungen verwendet werden. Das impliziert einen sehr hohen Anspruch an die Leistung des Systems. Deshalb soll die Leistung des Systems höchste Priorität haben. Dabei wird eine weniger modulare und dadurch weniger erweiterungs- und anpassungsfähige Architektur in Kauf genommen. Die im Folgenden aufgeführten Punkte stellen hierbei jedoch Ausnahmen dar:

- Unterstützung von Dokumententypen - damit später die Unterstützung neuer Versionen bereits unterstützter Dokumententypen möglich ist. Aber auch das Einbinden neuer Datentypen soll damit ermöglicht werden. So wäre denkbar, multimediale Datentypen wie beispielsweise Video- und Audio-Streams irgendwann einzubinden.

- Annotationen - um ein Integrieren von weiteren, auch nicht textorientierten Annotationen wie beispielsweise Markierungen oder Ähnlichem zu ermöglichen.

- Annotationsklassifizierungen - damit eine Erweiterung um Klassifizierungen wie Hintergrundinformation, Beispiel, Übung oder Ähnlichem realisierbar ist.

- Benutzer- und Rechtemanagement - damit das System in dieser Hinsicht auf die spezifischen Anforderungen der jeweiligen Betreiber angepasst werden kann.

- Datenhaltung / Datenbanksysteme - damit lokal wie auf der Austauschplattform die Datenhaltungsfunktionalitäten, falls erforderlich, ausgetauscht werden können.
- Bewertungs- und Vertrauensbildungssystem - damit die Wunschanforderung, das System als Open Source Software zur Verfügung zu stellen, erfüllt werden kann. Bei der Verwendung von UniTEC, wie in dieser Arbeit vorgesehen, kann dem ansonsten nicht nachgekommen werden, weil UniTEC keine Open Source Software ist.

3.4 Austauschplattform

Um die von anderen Nutzern im System zur Verfügung gestellten Informationen wie in den Szenarien beschrieben nutzen zu können, ist eine Austauschplattform erforderlich, über welche der Informationsaustausch gewährleistet werden kann. Die Austauschplattform muss deshalb Mechanismen bereitstellen, mittels derer Annotationen veröffentlichten Dokumenten global eindeutig zugeordnet werden können. Auch das global eindeutige Zuordnen von Bewertungen zu den Dokumenten bzw. Annotationen muss gewährleistet werden.

Die Kommunikation mit der Austauschplattform muss auch trotz der in universitären Netzen üblichen Fire-Walls stattfinden können.

Der Zugang zur Austauschplattform und den darauf befindlichen Daten soll mittels einer Zugangskontrolle und einem Benutzer- und Rechtemanagement seitens der Austauschplattform geregelt werden können. Die Austauschplattform muss also die dafür erforderlichen Funktionalitäten bereitstellen.

3.5 Handhabung der Datenmenge

Im System müssen lokal wie auch global große Datenmengen bewältigt werden.

Ein erster Schritt, dies lokal unter Kontrolle zu bekommen, ist das Verwenden der Abonnementfunktionen, wie in den Szenarien beschrieben und in 3.8 verdeutlicht.

Eine weitere Eingrenzung soll mittels der Bewertungen und darauf basierender Selektion ermöglicht werden. Jeder Nutzer hat jedoch andere Vorstellungen, wie die Informationen zu bewerten sind. Da nur mit einem Vertrauensmanagement eine persönliche Selektion gewährleistet werden kann, soll ein solches Verwendung finden. Nur mit dem Zuordnen der Informationen zu Kategorien und dem Definieren von Abhängigkeiten dieser Kategorien ist ein solches Vertrauensmanagement differenziert genug möglich. Es muss außerdem dafür gesorgt werden, dass nicht zu viele, bzw. nur die Daten lokal gehalten werden, die wirklich relevant für den einzelnen Nutzer sind. Das Vertrauensmanagement ist hierfür ebenfalls unumgänglich.

Auf globaler Ebene sind entsprechende Mechanismen, natürlich auf die global erforderliche Datenmenge bezogen, ebenfalls wünschenswert. So wäre es möglich, automatisiert Daten anhand von kumulierten Bewertungen und Vertrauenswerten auszusortieren. Dies wäre jedoch sehr rechenintensiv und aufwändig. Deshalb ist fraglich, ob es in der Praxis sinnvoll ist, ein Vertrauensmanagement zum 'Aussortieren' von Daten zu verwenden.

Zudem ist es erheblich einfacher, diese Aufgabe 'von Hand' zu erledigen, indem nach einer gewissen Zeit nach dem Ende einer Veranstaltung alle Daten dieser auf entsprechende Anweisung hin gelöscht werden. Deshalb soll hier 'nur' gefordert werden, dass die entsprechenden Administrationsmechanismen zur Verfügung stehen.

3.6 Handhabbarkeit von Bewertungen

Um Bewertungen zum automatischen Filtern und auch im Zuge der Vertrauensbildung nutzen zu können, ist es erforderlich, dass anhand von immer gleichen Kriterien und jeweils einer immer gleichen Skala bewertet wird.

Außerdem muss eine Bewertung vom Nutzer sehr einfach und schnell getätigt werden können, da jeder Benutzer Bewertungen - wie in den Szenarien beschrieben - sehr oft abgeben muss. So muss bei jedem 'Hinzufügen' sowie 'Entfernen' einer Annotation aus der persönlichen Sicht eine Bewertung erstellt werden. Nebenbei soll das Bewerten auf freiwilliger Basis möglich sein, was jedoch, wenn die Bewertung zu lange dauert, nicht verwendet werden würde.

3.7 Haftung für Informationen im System

Es stellt sich natürlich die Frage, wer für die Inhalte, die die Nutzer im System eingeben, verantwortlich ist.

Man denke dabei beispielsweise an Verletzungen des Urheberrechtes, wenn Nutzer falsch zitieren, oder Quellen von Informationen nicht, oder nicht korrekt angeben. Des weiteren kann es passieren, dass jemand, der es nicht besser weiß oder auch absichtlich falsche Informationen in das System einspeist.

Auch weiterer Missbrauch des Systems soll hier als kritischer Punkt angesprochen werden. So kann es beispielsweise kaum verhindert werden, dass Nutzer andere Nutzer verleumden oder beschimpfen. Auch unsachgemäße oder unschön formulierte Kritik einer Gruppe beispielsweise am Dozenten kann durchaus vorkommen und nur schwerlich verhindert werden.

Sicherlich kann mit den meisten Problemen in dieser Richtung mit dem Publizieren einer Netiquette für das System und gegenseitiger Ermahnung entgegengetreten werden. Durch die Filterungsmechanismen ist mit Hilfe von entsprechendem Bewerten fraglicher Veröffentlichungen ein mächtiges Instument vorhanden. Doch was ist, wenn wirklich einmal eine Klage wegen Inhalten in Annotationen oder Dokumenten vorliegt?

Damit nicht der Betreiber des Systems dafür gerade stehen muss, ist mit Sicherheit nötig, dass zumindest für den Betreiber die Nutzer nicht anonym sind, der Betreiber also im Problemfall die entsprechende Person aus dem System ausschließen und bei rechtlicher Handhabe auf diese verweisen kann. Nutzungsbedingungen, ähnlich denen der universitären Netzzugänge, sollten demzufolge erstellt werden.

3.8 Funktionale Anforderungen

In diesem Abschnitt sollen nun die funktionalen Anforderungen aufgeführt werden. Die hierfür gewählte Darstellungsform lehnt sich an die in Pflichten- bzw. Lastenheften übliche an. Es wurde für jede funktionale Anforderung ein Kürzel 'FA' vergeben, um einfacher Verweise und Querverweise angeben zu können. Die Nummerierung wurde zuerst in Zehnerschritten durchgeführt, so dass während der Anforderungsanalyse weitere Punkte eingefügt werden konnten.
Um die vollzogenen Ableitungsschritte nachvollziebar zu machen, wurden jeweils unter 'siehe auch' die entsprechenden Abschnitte aus den Szenarien oder anderen Bereichen dieser Arbeit angegeben.

Bezeichnung: FA1 - Online/Offline

Beschreibung: Es soll möglich sein, mit dem System zu arbeiten, ohne mit der Austauschplattform dauerhaft verbunden zu sein. Wie in FA50 - 'Veranstaltungs-/Dokumentensynchronisation' und FA190 - 'Annotationssynchronisation' beschrieben, sollen, sobald eine Verbindung zur Austauschplattform hergestellt ist, die Anweisungen des Nutzers, die eine Kommunikation nach sich ziehen, durchgeführt werden. Zur Gewährleistung dieser Vorgänge wird außerdem die in FA210 - 'lokale Datenhaltung' beschriebene Funktionalität benötigt.

siehe auch: 3.2

Bezeichnung: FA10 - Veranstaltungen und Dokumente anzeigen

Beschreibung: Veranstaltungen sowie Dokumente sollen visualisiert werden. Dadurch soll dem Nutzer ermöglicht werden, sich über seine eigenen sowie über die von anderen Nutzern über die Austauschplattform zur Verfügung gestellten Dokumente und Veranstaltungen einen Überblick zu verschaffen. Außerdem soll hierdurch Zugang zu den folgenden Funktionen bereitgestellt werden: FA20 - 'Veranstaltung hinzufügen / veröffentlichen', FA30 - 'Dokumente hinzufügen / veröffentlichen', FA37 - 'Literaturverzeichnis', FA40 - 'Dokumente herunterladen', FA60 - 'Dokumentenabonnement', FA70 - 'Formatunabhängige Dokumentenanzeige', FA135 - 'Annotationsabonnement', sowie FA215 - 'Daten sichern / auslagern' und FA230 - 'Löschen lokaler Daten'.

Bezeichnung: FA20 - Veranstaltung hinzufügen / veröffentlichen

Beschreibung: Ein Nutzer kann Veranstaltungen erstellen. Nach dem Erstellen ist eine solche ohne zu veröffentlichen nur ihm zugänglich. Wenn ein Nutzer die Rechte dazu hat, kann er eine Veranstaltung veröffentlichen (siehe FA200 - 'Rechte-/Benutzerverwaltung und Zugangskontrolle'). Daraufhin sind die öffentlichen Daten der Veranstaltung allen Nutzern zugänglich. Sofern es sich um eine passwortgeschützte Veranstaltung handelt, sind die Dokumente und Annotationen jedoch nur für die entsprechenden Nutzer auffindbar.

Veranstaltungen sollen ermöglichen, mehrere Dokumente zu bündeln und veranstaltungsspezifische Daten wie Veranstaltungsort, Durchführungszeitraum, ... bereitzustellen.

siehe auch: 2.2.1 Abschnitt 'Öffentliche Daten' und Abschnitt 'Zugangsbeschränkung'

Bezeichnung: FA25 - Themenkategorie

Beschreibung: Eine Veranstaltung muss bei Erstellung oder Veröffentlichung einer Themenkategorie zugeordnet werden. Die Zuordnung zu Themenkategorien ist zur Vertrauensbildung und für die Filterungsmechanismen erforderlich. Näheres dazu unter FA180 - 'Vertrauensbildung und -management' und FA190 - 'Annotationssynchronisation'.

siehe auch: 2.2.6 Abschnitt 'Vertrauensbildung; 2.2.1 Abschnitt 'Öffentliche / Allgemeine Daten'.

Bezeichnung: FA30 - Dokumente hinzufügen / veröffentlichen

Beschreibung: Ein Dokument kann einer Veranstaltung zugeordnet und zur lokalen Datenhaltung (FA220 - 'Austauschplattform') des Systems hinzugefügt werden. Ein solches Dokument kann dann veröffentlicht werden und wird dadurch anderen Nutzern zugänglich.

siehe auch: 2.2.4 Abschnitt 'Dokumente veröffentlichen / Übungen'

Bezeichnung: FA33 - Feedback - kritische Dokumentbereiche

Beschreibung: Ein Dokumentbereich, auf welchen sich eine vom Nutzer definierbare Mindestanzahl von Annotationen bezieht, soll als kritischer Dokumentbereich bezeichnet werden. Der Nutzer soll definieren können, wie viele Annotationen, mit welchen Bewertungs- und Vertrauenseigenschaften innerhalb eines Bereiches (beispielsweise einer Seite) vorhanden sein müssen, damit dieser kritisch wird. Diese kritischen Dokumentbereiche sollen aufgelistet und nach Auswahl aus der Liste angezeigt werden können.

siehe auch: 2.2.5; 2.2.1 Abschnitt 'Wiederholungsveranstaltung / Wiederverwendung'

Bezeichnung: FA35 - Feedback - übernehmen von Teilen einer abgeschlossenen Veranstaltung

Beschreibung: Erstellt man eine Veranstaltung, so soll es möglich sein, aus einer bestehenden Veranstaltung Dokumente zu importieren. Ein Übernehmen eines Dokuments ohne die zugehörigen Annotationen soll möglich sein. Es soll aber auch freistehen, die Annotationen zu diesen Dokumenten dabei mitzuübernehmen. Die Annotationsübernahme soll wahlweise ungefiltert, also alle Annotationen, oder, mittels der Bewertungen und der Vertrauenswerte, gefiltert möglich sein.

siehe auch: 2.2.1 Abschnitt 'Wiederholungsveranstaltung / Wiederverwendung'

Bezeichnung: FA37 - Literaturverzeichnis

Beschreibung: Es soll möglich sein, Verweise bzw. Quellenangaben bezüglich externer Literatur einzugeben. Diese Literaturangaben sollen dann mittels des Literaturverzeichnisses gesucht und angezeigt werden können.

siehe auch: 2.2.1 Abschnitt 'Erstmalige Veranstaltung'; 2.2.4 Absatz 'Recherche außerhalb der Systems / Literaturverzeichnis'

Bezeichnung: FA40 - Dokumente herunterladen

Beschreibung: Dokumente, die von anderen Nutzern veröffentlicht wurden, können manuell oder mittels eines Dokumentenabonnementes (FA60) automatisiert heruntergeladen werden. Danach stehen sie dem Nutzer zur Ansicht und zum Annotieren bereit. Siehe auch FA220 - 'Austauschplattform', FA190 - 'Annotationssynchronisation', FA50 - 'Veranstaltungs- /Dokumentensynchronisation' und FA1 - 'Online/Offline'.

siehe auch: 2.2.2

Bezeichnung: FA50 - Veranstaltungs-/Dokumentensynchronisation

Beschreibung: Ist das System bei der Nutzung mit der Austauschplattform verbunden, wird überprüft, welche Veranstaltungen und Dokumente seit der letzten Verbindung veröffentlicht wurden. Die betreffenden Veranstaltungen und Dokumente des Nutzers werden auf die Austauschplattform geladen. Informationen über Dokumente und Veranstaltungen anderer Nutzer werden von der Austauschplattform geladen (siehe FA10 - 'Veranstaltungen und Dokumente anzeigen'). Wurde die Anweisung gegeben, dass Dokumente heruntergeladen werden sollen (siehe FA40 - 'Dokumente herunterladen' und FA60 - 'Dokumentenabonnement'), wird dies durchgeführt.

Die Synchronisation soll bei Systemstart sowie auf explizite Anweisung des Nutzers durchgeführt werden. Periodische Synchronisation soll ebenfalls möglich sein. Die Nutzer sollen während des Synchronisationsvorganges weiter mit dem System arbeiten können.

Bezeichnung: FA60 - Dokumentenabonnement

Beschreibung: Für eine Veranstaltung soll festgelegt werden können, dass alle Dokumente herunter geladen werden. Dies gilt dann auch für neu hinzukommende Dokumente, bezogen auf die Veranstaltung.

siehe auch: 2.2.2

Bezeichnung: FA70 - Formatunabhängige Dokumentenanzeige

Beschreibung: Dokumente im System (siehe FA10 - 'Veranstaltungen und Dokumente anzeigen') unabhängig vom Format (html, pdf, doc, ...) sollen angezeigt werden können.

siehe auch: 2.2.3 Abschnitt 'Mitschreiben / Fragen stellen'

Bezeichnung: FA75 - Skriptstellensuche

Beschreibung: Unabhängig vom Format des vorliegenden Dokumentes soll eine Volltextsuche möglich sein.

siehe auch: 2.2.3 Abschnitt 'Skriptstellensuche'

Bezeichnung: FA80 - Annotationen erstellen

Beschreibung: Es soll möglich sein, Annotationen mit feingranularem Bezug zu einem Dokument unabhängig von dessen Dateiformat zu erstellen. Eine Annotation bezieht sich auf eine genaue Stelle des Dokuments, also auf eine Position (x,y) oder auf eine Seite und eine Position (x,y) auf dieser Seite. Eine neu erstellte Annotation ist nur dem Nutzer selbst zugänglich. Die Annotation besteht aus einem frei definierbaren Fließtext.

siehe auch: 2.2.3 Abschnitt 'Mitschreiben / Fragen stellen'

Bezeichnung: FA90 - Annotationsklassifizierung

Beschreibung: Eine Annotation kann beim Erstellen (FA80) klassifiziert werden. Mögliche Klassifikationen sind 'Frage' sowie 'Anmerkung'.

siehe auch: 2.2.3 Abschnitt 'Mitschreiben / Fragen stellen'; 2.2.4 Abschnitt 'Fragen stellen und beantworten'; 3.3

Bezeichnung: FA100 - Annotationen veröffentlichen

Beschreibung: Eine Annotation kann veröffentlicht und dadurch anderen Nutzern zur Verfügung gestellt werden.

siehe auch: 2.2.3

Bezeichnung: FA110 - Sichten

Beschreibung: Voreinstellung ist die 'persönliche Sicht' (siehe FA120). Vom Nutzer selbst erstellte Annotationen gehören automatisch zu dieser Sicht. Annotationen von anderen Nutzern gehören der 'nicht-persönlichen Sicht' an. Als Fragen klassifizierte Annotationen stellen hier einen Sonderfall dar. Mehr hierzu unter FA125 - 'Fragen'. Annotationen der 'persönlichen Sicht' gehören immer auch der 'nicht-persönlichen Sicht' an (siehe auch FA135 - 'Bewertungen'). Als dritte Sichtweise soll das Darstellen ohne Annotationen möglich sein.

siehe auch: 2.2.4 Abschnitt 'Sichten'; 2.2.5

Bezeichnung: FA120 - Persönliche Sicht

Beschreibung: Befindet man sich in der 'nicht-persönlichen Sicht', ist es möglich, Annotationen der persönlichen Sicht hinzuzufügen (siehe auch FA 110 - 'Sichten'). Dabei soll die Annotation bewertet werden müssen (FA130 - 'Bewertungen'). Annotationen der persönlichen Sicht können aus dieser entfernt werden, und müssen dann ebenfalls bewertet werden.

siehe auch: 2.2.4 Abschnitt 'Sichten'

Bezeichnung: FA125 - Fragen

Beschreibung: Wie schon in FA110 - 'Sichten' angesprochen, stellen als Fragen klassifizierte Annotationen bei der Visualisierung einen Sonderfall dar. Sie werden zwar wie andere Annotationen auch den entsprechenden Sichten gemäß angezeigt, jedoch werden neue, noch nicht gelesene Fragen zusätzlich als Symbol in der persönlichen Sicht angezeigt. Nach einem Klick auf das Symbol wird der Inhalt der Frage dargestellt und danach als gelesen angesehen. Hiermit soll gewährleistet werden, dass Fragen schnell gelesen und beantwortet werden.

Neben der Darstellung von Fragen auf dem Dokument werden diese außerdem im Fragenverzeichnis dargestellt. Es werden zur persönlichen Sicht gehörende sowie ungelesene Fragen dargestellt. Ungelesene Fragen werden als solche gekennzeichnet. Neue Annotationen zu Fragen aus der persönlichen Sicht werden ebenfalls kenntlich gemacht (ähnlich üblicher Darstellungen in Diskussionsforen). Es soll möglich sein, sich mittels Auswahl einer Frage das entsprechende Dokument anzeigen zu lassen.

siehe auch: 2.2.4 Abschnitt 'Fragen stellen und beantworten'; 2.2.5

Beschreibung: FA130 - Bewertungen

Beschreibung: Bewertungen müssen zumindest beim Hinzufügen und Herausnehmen von Annotationen aus der persönlichen Sicht gemacht werden. Außerdem soll es jederzeit möglich sein, freiwillig Annotationen sowie Dokumente zu bewerten.

siehe auch: 2.2.4 Abschnitt 'Sichten'; 2.2.6

Beschreibung: FA135 - Annotationsabonnement

Beschreibung: Ob, welche und wieviele der von anderen Nutzern veröffentlichten Annotationen zu einem Dokument in der nicht-persönlichen Sicht zur Verfügung stehen, soll mittels eines Annotationsabonnements festlegbar sein. Zur Auswahl stehen keine, alle oder eine anhand von Vertrauenswerten (FA180 - 'Vertrauensbildung und -management') gefilterte Teilmenge.

Bezeichnung: FA140 - Verknüpfungen erstellen

Beschreibung: Mit feingranularem Bezug (wie bei Annotationen in FA80 - 'Annotationen erstellen') sollen Verknüpfungen erstellt werden können und benannt werden, und auf eine Stelle in demselben Dokument oder auf eine Stelle in einem anderen Dokument verweisen. Ähnlich Hyperlinks, jedoch unabhängig vom Dateiformat des Dokuments.

siehe auch: 2.2.4 Abschnitt 'Dokumente verknüpfen'

Bezeichnung: FA150 - Verknüpfungen verfolgen

Beschreibung: Nach einem Klick des Nutzers auf eine Verknüpfung sollen die Stelle und das Dokument angezeigt werden, auf welche die Verknüpfung verweist. Mittels bei Browsern üblichen 'vor und zurück'-Schaltflächen soll das Entlanghangeln an bereits getätigten Dokumentsprüngen möglich sein.

siehe auch: 2.2.4 Abschnitt 'Dokumente verknüpfen'

Bezeichnung: FA160 - Verknüpfungen anzeigen

Beschreibung: Verknüpfungen sollen wie Annotationen auf den Dokumenten angezeigt werden, aber auch ausblendbar sein.

siehe auch: 2.2.4 Abschnitt 'Dokumente verknüpfen'

Bezeichnung: FA170 - Diskussion

Beschreibung: Es soll möglich sein, mittels des Systems Diskussionen zu führen. Dies soll dadurch gewährleistet werden, dass an Annotationen weitere angehängt werden können. Auf diese Weise können auch Fragen (FA125) beantwortet werden.

siehe auch: 2.2.4 Abschnitt 'Fragen stellen und beantworten'

Bezeichnung: FA180 - Vertrauensbildung und -management

Beschreibung: Mittels der Bewertungen sollen bezüglich der anderen Nutzer Vertrauenswerte ermittelt werden. Mittels dieser wird bei filterndem Annotationsabonnement (FH135 - 'Annotationsabonnement') die Auswahl bei der Synchronisation (FH190) stattfinden. Die Vertrauenswerte werden jeweils bezogen auf Anwender und Themenkategorien. Gebildet werden sie über Bewertungsvergleiche. Die für die Konfiguration der Vertrauensbildung benötigten Schwell- und Maximalwerte sollen vom Benutzer einstellbar sein.

siehe auch: 3.5; 3.6; 2.2.6

Bezeichnung: FA190 - Annotationssynchronisation

Beschreibung: Ist das System bei der Nutzung mit der Austauschplattform verbunden, wird überprüft, welche Annotationen seit der letzten Verbindung vom

Nutzer veröffentlicht wurden. Die betreffenden Annotationen des Nutzers werden auf die Austauschplattform geladen. Sind Annotationsabonnements eingerichtet, wird überprüft, ob bei den gewählten Einstellungen Annotationen von der Austauschplattform geladen werden sollen. Ist dies der Fall, wird der Austausch durchgeführt.

Die Synchronisation soll bei Systemstart für alle Annotationsabonnements durchgeführt werden. Wurde ein Dokument angezeigt, so wird ebenfalls eine Synchronisation bezüglich dieses Dokumentes durchgeführt. Die Nutzer sollen während des Synchronisationsvorgangs weiter mit dem System arbeiten können.

siehe auch: 2.2.1

Bezeichnung: FA200 - Rechte-/Benutzerverwaltung und Zugangskontrolle

Beschreibung: Mittels der Austauschplattform (FA220) soll eine globale Benutzer und Rechteverwaltung ermöglicht werden. Mittels dieser soll eine generelle Zugriffskontrolle bzw. Beschränkung auf die Austauschplattform möglich sein. Außerdem sollen auf Nutzergruppen bezogene Rechte verwaltet und geprüft werden können. Das wird für die in FA20 - 'Veranstaltung hinzufügen / veröffentlichen' beschriebene Funktionalität bezüglich des Veröffentlichens von Veranstaltungen benötigt.

siehe auch: 2.2.1

Bezeichnung: FA210 - lokale Datenhaltung

Beschreibung: Um die in FA1 - 'Online/Offline' beschriebenen Funktionalitäten realisieren zu können, sind lokal einige Daten erforderlich. Diese Daten müssen gespeichert werden.

siehe auch: 2.1

Bezeichnung: FA215 - Daten sichern / auslagern

Beschreibung: Es soll möglich sein, alle zu einer Veranstaltung gehörenden lokal gehaltenen Daten zu sichern. Ausgelagerte Daten bezüglich einer Veranstaltung sollen dauerhaft oder auch temporär wieder in das System eingebunden werden können, sodass wieder mit ihnen gearbeitet werden kann.

siehe auch: 2.2.7

Bezeichnung: FA220 - Austauschplattform

Beschreibung: Die Austauschplattform soll die in FA200 - 'Rechte-/Benutzerverwaltung und Zugangskontrolle' beschriebenen Funktionalitäten sowie das Austauschen von Dokumenten und Annotationen ermöglichen. Eine entsprechende Speicherung und Verwaltung der Dokumente und Annotationen soll gegeben sein. Benötigte Administrationsfunktionalitäten sollen ebenfalls vorhanden sein.

siehe auch: 2.1

Bezeichnung: FA230 - Löschen lokaler Daten

Beschreibung: Es soll möglich sein, lokal gehaltene Daten zu einer Veranstaltung zu löschen. Auch lokal gehaltene Dokumente und Annotationen sollen einzeln gelöscht werden können. Hierbei wird dafür gesorgt, dass diese bei der Synchronisation nicht erneut geladen werden.

siehe auch: 2.2.7

Bezeichnung: FA240 - Löschen globaler Daten

Beschreibung: Administratoren, also einem Personenkreis mit speziellen Rechten, soll es möglich sein, einzelne Dokumente und Annotationen, aber auch die kompletten Daten einer Veranstaltung aus dem System zu entfernen.

siehe auch: 2.2.7

Kapitel 4

Verwandte Arbeiten

Im Folgenden sollen relevante Arbeiten beziehungsweise Systeme daraufhin untersucht werden, ob sie oder Teile aus ihnen im Hinblick auf die im vorangegangenen Kapitel erarbeiteten Anforderungen verwendet werden können. Auch soll hierbei auf Methoden und Konzepte, die für den Entwurf des Systems oder für eventuelle Erweiterungen dieser Arbeit relevant sind, eingegangen werden.

4.1 Lernplattformen

Die in diesem Abschnitt beschriebenen Systeme sind Plattformen, welche zur Unterstützung von Lernvorgängen entwickelt wurden. Diese Lernplattformen werden in der Lehre schon verwendet und können sich teilweise einer großen Verbreitung an Universitäten erfreuen. Alle betrachteten Plattformen sind webbasierte Systeme. Auf der Clientseite wird zur Bedienung also nur ein üblicher Browser benötigt. Die Daten werden mit einem Webserver zur Verfügung gestellt. Die Funktionalitäten sind mit Hilfe von serverseitigen Skripterweiterungen realisiert. Bei manchen Systemen kommt darüber hinaus noch ein Datenbankserver zum Einsatz. Zur Realisierung der Lernplattformen wurde auf die LAMP-Architektur oder vergleichbare Architekturen zurückgegriffen (LAMP = Linux Apache MySql PHP).

Als Hauptproblem in dieser Systemgruppe stellte sich heraus, dass die Systeme dieser Kategorie nicht über einen Client verfügen, welcher Replikations- und Synchronisationsmechanismen zur Verfügung stellt. Deshalb ist es nicht möglich, sie offline zu verwenden, was aber eine Anforderung dieser Arbeit ist. Aus diesem Grund macht es keinen Sinn, zur Realisierung dieser Studienarbeit Systeme aus dieser Gruppe komplett heranzuziehen bzw. sie zu erweitern. Als weiteres Manko ist anzusehen, dass, sofern es überhaupt möglich ist, Annotationen in den Systemen zu erstellen, diese nur in grobgranularer Weise gemacht werden können. Aus diesem Grund ist es nicht möglich, diese Systeme zum Mitschreiben in Veranstaltungen zu verwenden, was aber eine der Hauptanforderungen dieser Arbeit ist.

Jedoch sollten die in diesem Abschnitt beschriebenen Systeme insbesondere wegen ihrer relativ hohen Verbreitung als mögliche Datenquellen für Lernmaterialien gesehen werden.

4.1.1 BSCW, BSCL und Synergeia

Im Rahmen des Projektes ITACOLE (siehe [*IT]) wurde das System Synergeia entwickelt (siehe auch [App03] und [*SY]).

Synergeia ist für den Einsatz in Schulen gedacht und soll mit Hilfe des darin verwirklichten pädagogischen Modells 'Progressive Inquiry Theorie' kollaboratives Lernen unterstützen (siehe auch [Sta]). Dieses pädagogische Modell beinhaltet beispielsweise das in sogenannten 'Knowledge Building Areas' integrierte Konzept der 'Thinking Types'. Durch diese sollen die Schüler dazu angehalten werden, sich vor Erstellung eines Beitrages darüber klar zu werden, ob sie beispielsweise ein Problem formulieren oder einen Kommentar abgeben möchten. Hierbei stehen Klassifikationen wie 'Problem', 'Arbeitshypothese', 'Kommentar', 'Bitte um Hilfe' zur Auswahl. Die Inhalte der verschiedenen 'Thinking Types' werden anhand von zugeordneten Farben angezeigt und auf diese Weise entsprechend kenntlich gemacht.

In Synergeia werden das synchrone sowie das asynchrone Arbeiten unterstützt. Auf das in Synergeia eingebundene Tool namens 'Map Tool', welches für das synchrone Arbeiten zuständig ist, soll im Folgenden nicht weiter eingegangen werden, da nur die asynchrone Arbeitsweise in dieser Arbeit relevant ist. Die Funktionalität für das asynchrone Arbeiten wird vom in Synergeia integrierten BSCL bereitgestellt. BSCL basiert auf der erweiterten BSCW-Plattform (siehe [*BS]) und ist wie diese mit Hilfe von CGI-Erweiterungen eines Web-Servers verwirklicht.

In BSCL wird insbesondere die Konstruktion von textuellen Inhalten, also von Dokumenten, Bookmarks und Diskussionen unterstützt. Die Diskussionen finden in den schon erwähnten 'Knowledge Building Areas' statt. Als Funktionsumfang werden die Authentifizierung, Versionsverwaltung, Zugriffsrechte-Management, Suchfunktionen, Mehrsprachenunterstützung sowie ein Ereignisdienst angeboten. Dabei wurde insbesondere das Benutzerinterface von BSCW überarbeitet und schulischen Anforderungen angepasst. Im Zuge dessen ist außerdem ein Klassen/Gruppen-Konzept eingeführt worden. Auch die Rollen 'Schüler und Lehrer' wurden mit den dazugehörigen Rechten im System hinzugefügt.

In Synergeia bzw. BSCL sowie ebenfalls in BSCW können Beiträge nur in 'Knowledge Building Areas' (Diskussionsforen) oder mit Bezug auf ein Dokument gemacht werden. Auch können Beiträge hier nicht mit weiteren Dokumenten verknüpft werden und in einem Beitrag kann der Bezug zu einem anderen nur mit Hilfe einer textuellen Beschreibung hergestellt werden.

Außerdem sind auch keinerlei Selektionsmechanismen (Bewertung und Filterung) zur Erzeugung einer Gesamtsicht aus den persönlichen Sichten vorhanden. Das ist nicht verwunderlich, da das pädagogische Modell hinter Synergeia Lernfortschritte durch 'diskursive Wissensbildung' fordert. Unter 'diskursiver Wissensbildung' ist zu verstehen, das durch Betrachten aller Meinungen und Beiträge und anschließendem Auswählen, Zusammenfügen und Ergänzen eine eigene Sicht gebildet wird. Synergeia ist für kleine Gruppen entwickelt worden, bei denen die Informationen noch 'per Hand' gefiltert werden können. In dieser Arbeit ist jedoch als Anforderung die Verwendbarkeit für große Gruppen gestellt und das

Vorhandensein von Selekionsmöglichkeiten gefordert. Auch, wie oben schon für alle Systeme aus dieser Gruppe beschrieben, ist eine Anforderung dieser Arbeit, dass Annotationen feingranular vorgenommen werden können und diese sowie Dokumente sich verknüpfen lassen.

Zur eventuellen Erweiterung dieser Arbeit ist das in Synergeia in den 'Knowledge Building Areas' verwendete Konzept der 'Thinking Types' (siehe oben) sehr interessant. So wäre eventuell zur Erweiterung der bisherigen Klassifizierung von Annotationen die feingranularere Klassifizierung des 'Thinking Type' Konzeptes geeignet, müsste aber in jedem Falle an die andere Umgebung dieses Systems sowie die andere Zielgruppe angepasst werden.

4.1.2 ILIAS

ILIAS ist ein webbasiertes System zur Unterstützung von selbstgesteuertem sowie kooperativem Lernen. Dieses Open-Source-Projekt von der Universität Köln in Zusammenarbeit mit einigen Partnern findet sich zum Download sowie als Testinstallation unter [*IL]. Auch weitere allgemeine Informationen und Erläuterungen zu den dahinterstehenden Konzepten sind dort zu finden.

In ILIAS werden die Rollen 'Lernender und Lehrender' unterschieden. Lehrende haben mehr Rechte als Lernende. Im Gegensatz zu den Lernenden dürfen sie Lerneinheiten anlegen und bearbeiten. In ILIAS wurden die bereitgestellten Editierfunktionen durch heranziehen von im Internet üblichen Oberflächen realisiert. Mit diesen ist es spartanisch möglich, Tabellen, Texte und Bilder zu Lerneinheiten zusammenzuklicken. Auch können Dokumente in die Lerneinheiten eingebunden werden. Den Lehrenden werden Funktionalitäten zum Erstellen von Lückentext- sowie Multiple-Choice-Aufgaben passend zu den Lerneinheiten bereitgestellt.

Die Lernenden können sich die Lerneinheiten ansehen. Mit Hilfe von in das System integrierten Diskussionsforen kann über die Lerneinheiten diskutiert werden. Auch kann je Lerneinheit ein Kommentar angebracht werden, welcher ähnlich wie bei Synergeia (siehe 4.1.1) klassifiziert werden kann. Die Klassifizierung ist hier jedoch nur sehr rudimentär möglich und wird nur durch ein kleines Bildchen visualisiert. Die Kommentare sind nur für einen selbst sichtbar. ILIAS unterstützt außerdem Bookmarkfunktionalitäten ähnlich wie diese zur Verwaltung von Hyperlinks in handelsüblichen Browsern realisiert sind. Die Möglichkeiten, Lerneinheiten zu drucken oder herunterzuladen, ist ebenfalls realisiert.

Als eine interessante Erweiterungsmöglichkeit für diese Arbeit ist die in ILIAS verwirklichte Methode anzusehen, Lückentexte sowie Multiple-Choice-Fragen im System zu integrieren. Auf diese Weise können Übungen oder auch Scheinprüfungen mittels des Systems zumindest zum Teil verwirklicht werden und machen eine Auswertung beziehungsweise Feedback einfacher. Es muss jedoch bedacht werden, das diese 'Prüf-' sowie 'Feedback-Realisierung' sicher nicht für alle Themen- beziehungsweise Stoffgebiete geeignet ist.

4.1.3 metacoon

metacoon (siehe [HTZS03]) ist ein weiterer Vertreter der Lernplattformen und in Funktionsumfang und Realisierung den Systemen Synergeia und ILIAS sehr ähnlich. Allerdings gibt es in metacoon im Gegensatz zu den oben beschriebenen Systemen keine Möglichkeit, Annotationen zu erstellen. Positiv aufgefallen ist bei metacoon die sehr gute Benutzerführung und Bedienbarkeit im Vergleich zu ILIAS.

4.1.4 eclaus

eclaus ist ein System zur Unterstützung der Durchführung von vorlesungsbegleitenden Übungen im universitären Bereich. Das System wurde in der Arbeit [BEH+02] entwickelt und wird momentan in mehreren Vorlesungen an der Universität Stuttgart verwendet (siehe [*ec]) und von einem Supportteam betreut sowie weiterentwickelt.

eclaus unterstützt insbesondere das Erstellen von Übungsaufgaben (Multiple-Choice-, Votieraufgaben, Freitextaufgaben und Programmieraufgaben), deren Verwaltung sowie die entsprechende Abgabe der Lösungen der Studenten. Dabei sind Hilfen für die Korrektur der Aufgaben sowie die Verwaltung der 'Benotung' in Form von Punkten realisiert. Außerdem ist ein komplexes Rollen-, Rechte- sowie Benutzer-Management implementiert. Dieses ist jedoch leider komplett in das System verwoben, was an der Verwendung von Servlets liegt, sodass es für diese Arbeit nicht herangezogen werden kann. Die Abgabe der Studenten zu einem Übungsblatt kann dabei in Gruppen geschehen. Die zu einem Übungsblatt gehörenden Abgaben werden per e-mail als ZIP-Datei den Korrigierenden zugeschickt, damit diese nicht alle Abgaben einzeln herunterladen müssen. Statistiken und ähnliche Funktionalitäten sind ebenfalls realisiert. Die Eingabe der Lösungen geschieht mittels der Weboberfläche, wobei Freitexte, Multiple-Choice-Auswahl sowie Datei-Uploads verwendet werden können. Bei den Programmieraufgaben können die Quellcode-Programme auf lauffähige Übersetzung überprüft werden. Annotationsfunktionen sind leider nicht vorhanden.

4.2 Annotationssysteme

In diesem Abschnitt sollen Systeme beschrieben werden, die in erster Linie zur Erstellung und Verwaltung von Annotationen entwickelt wurden. Ähnlich wie bei den Lernplattformen (siehe 4.1) sind diese leider ebenfalls von webbasierten Technologien dominiert und damit nicht offline benutzbar. Eine Ausnahme stellt ToGather dar, welches eigentlich lokal verwendet werden könnte, durch die Account-Abfrage mittels Java Web Start jedoch auch eine Netzwerkverbindung benötigt. Somit gilt wiederum für alle Systeme aus dieser Gruppe, dass sie zum Offline-Arbeiten nicht herangezogen werden können. Ein weiteres Manko der in diesem Teil beschriebenen Systeme ist, dass es sich um Prototypen handelt und deshalb die Bedienbarkeit bzw. Benutzerführung Mängel aufweist. Auch die Darstellung der Annotationen insbesondere bei dem IAS Notizeditor und der Annotationserweiterung von Targeteam entspricht nicht den in FA80 - 'Annotationen erstellen' geforderten Eigenschaften.

4.2.1 ToGather

Das im Verbund mit weiteren Anwendungen im Projekt namens 'Digitale Hörsäle' entstandene ToGather (siehe [TRB+03]) ist ein Tool, mit dessen Hilfe während der Vorlesung sowie beim Betrachten der Videoaufzeichnungen und beim Durchgehen der einzelnen Folien Annotationen angebracht werden können. Dies ist anhand von einfachen Stift- und Kastenmarkierungen sowie je Folie mit einem textuellen Kommentar möglich. Die Annotationen werden beim Abspielen der Vorlesung zeitgemäß wiedergegeben, sodass nach und nach hinzugefügte Erklärungen des Dozenten Schritt für Schritt nochmals nachvollzogen werden können. Bevor mit ToGather gearbeitet werden kann, muss eine 'Einladung' heruntergeladen und mit ToGather geöffnet werden. Außerdem müssen die Lernmaterialien geladen werden. Diese liegen im tg-Format, einem speziellen Format für ToGather vor. Es werden drei Betrachtungsweisen unterschieden: Life - also während der laufenden Vorlesung, Nachbearbeitung - zum Hinzufügen von Notizen, wobei Folien mit allen bisherigen Annotationen angezeigt werden, und das Playback - die zeitliche Abfolge des Hinzufügens der Notizen wird nochmals angezeigt. Eine Authentifizierung geschieht über Java Web Start, über welches ToGather gestartet werden muss (ToGather kann auf der ToGather Internetseite [*To] geladen und mit den unter [*Te] erhältlichen Daten getestet werden).

ToGather ist im Sinne der in dieser Arbeit definierten Anforderungen nicht geeignet. So ist ein asynchroner Austausch von Annotationen nicht möglich. Annotationen können nur je Folie erstellt werden und die Stiftfunktion macht höchstens zum Unterstreichen oder anderem Markieren Sinn, wenn man keine spezielle Handschrift-Hardware besitzt. Außerdem lässt sich auch nur das spezielle ToGather-eigene Datenformat verwenden, was den Anforderungen dieser Arbeit insofern widerspricht, als gefordert ist, verschiedene Dokumentformate zu unterstützen. Außerdem lässt die Bedienung beim asynchronen Annotieren von Folien zu wünschen übrig, da jede Folie einzeln ausgewählt werden muss, um annotieren zu können und vorher ein entsprechendes Darstellungsfenster von Hand geöffnet werden muss.

4.2.2 Annotationserweiterung von Targeteam

Das System Targeteam (TArgeted Reuse and Generation of TEAching Materials) ist ein Autorensystem, welches seit 1999 an der Technischen Universität München zur Erstellung von Lehrmaterialien eingesetzt wird. Ziel des Systems ist die Unterstützung der kollaborativen Erstellung und Nutzung von Begleitmaterialien zu Lehrveranstaltungen, also von Skripten, Folien sowie Aufgabensammlungen.

In Targeteam werden die Inhalte darstellungsneutral in der XML-Sprache TeachML erfasst und thematisch in Module aufgeteilt. Diese Module können dann zu Lehrveranstaltungen ausgewählt und hierarchisch miteinander vernetzt werden. Dabei ist es möglich, zusätzliche Inhalte wie beispielsweise Überleitungen hinzuzufügen und Teile der Module wegzulassen sowie umzuordnen.

Aus diesen darstellungsneutral zusammengestellten Daten können dann verschiedene

Ansichten generiert werden, wie beispielsweise vernetzte HTML-Seiten als Kurzform oder auch als vollständige Version oder als Skriptum zum Ausdrucken.

In der Arbeit [Zha03] wurde dieses System um Annotationsfunktionalitäten sowie eine Suchfunktion erweitert. Zu einer Seite können beliebig viele Annotationen gemacht werden, wobei der Bezug immer nur die Seite allgemein ist. Das System ist in einer LAMP-Architektur realisiert. Mechanismen zur Replikation und Synchronisation sind nicht vorhanden, weswegen ein Arbeiten offline nicht möglich ist.

4.2.3 IAS Notizeditor

Mit dem in [Sch02] beschriebenen IAS Notizeditor lassen sich Annotationen feingranular erstellen. Es werden zwei verschiedene Zuordnungsarten unterschieden. Rein textuelle (durch word-Dateien generiert), wo die Zuordnug zu Textabsätzen geschieht, und zu Grafiken (über Powerpoint erzeugt), wo die Zuordnung anhand einer XY-Position, also einem Punkt auf der Seite geschieht. Unterstützt werden ausschließlich html-Seiten. Diese können mit Hilfe eines Tools aus ppt bzw. doc Dateien generiert werden. Es wird eine einfache Benutzer-, Rollen- und Rechteverwaltung bereitgestellt. Die Notizen können den Autoren zugeordnet werden. Eine Administrationsschnittstelle zur Verwaltung der Notizen (zum eventuellen Löschen) und für Benutzer (zum Hinzufügen bzw. Löschen solcher) ist vorhanden. Die Notizeditorfunktionen werden dynamisch mittels Java Server Pages eingebunden. Allerdings dürfen die Webseiten nur statische Links enthalten, welche von den JSPs in JSP-Links umgewandelt werden. Die ursprüngliche URL ist dann in den JSP-Links als Parameter enthalten. Diese Umwandlung ist notwendig, da die URL für die Notizen als Referenz verwendet wird. Annotationen sowie die Benutzerdaten sind im System in einer Datenbank auf dem Server hinterlegt. Des weiteren wird zwischen öffentlichen und privaten Annotationen unterschieden. Die privaten Annotationen werden nur beim Ersteller angezeigt, die öffentlichen stehen allen Nutzern zur Verfügung.

Bei diesem System gibt es neben den für die gesamte Kategorie schon beschriebenen Problemen ein weiteres Manko. Es werden ausschließlich html-Dateien, deren Links in JSP-Links umgewandelt wurden, unterstützt.

4.3 Textverarbeitung/-visualisierung mit Annotationssystem

Bei den hier beschriebenen Programmen handelt es sich um Textverarbeitungs- beziehungsweise Visualisierungssoftware, welche Annotationsmöglichkeiten unterstützen.

Problematisch hinsichtlich der Anforderungen dieser Arbeit ist bei den Programmen dieser Gruppe zu sehen, dass sie die Annotationen in den Dokumenten und nicht getrennt von ihnen abspeichern, weswegen eine weitere Verteilung der Annotationen sehr schwierig oder nicht möglich, und eine erweiternde Einbindung von Bewertungs- bzw. Trustmechanismen unmöglich wird. Des weiteren stützen sich die Systeme auf jeweils ein Dokumentenformat und sind in dieser Hinsicht nicht erweiterbar. Erweiterungen dieser Systeme sind außerdem generell nicht oder nur sehr bedingt möglich, da es sich um kommerzielle Soft-

ware Produkte handelt, oder zumindest um Systeme, die auf einem solchen aufbauen. In den Anforderungen ist jedoch gefordert, dass verschiedene Dateiformate verwendbar sind. Auch ist die Nutzung von kommerzieller Software an sich ebenfalls sehr problematisch, da verlangt ist, dass die Software kostenfrei zur Verfügung stehen soll.

4.3.1 SPP

Shared Powerpoint ist im Zuge des NUSS-Projekts [BBW+03] entstanden. Dabei wurde in der Arbeit [Cer03] versucht, eine generische Möglichkeit zu finden, Anwendungen fernzusteuern. Damit sollte ein gemeinsames Arbeiten mit entsprechenden Anwendungen in der Lehre ermöglicht werden, wobei die Rechte- und Rollenverwaltung von NUSS mit zum Einsatz kommen sollte.

Prototypisch wurde dies anhand von Microsoft Powerpoint verwirklicht. Insbesondere wurde dabei das Augenmerk auf synchrone Kommunikation und auf die Anpassung an für die Lehre spezifische Anforderungen gelegt. Annotationen sind anhand der in Powerpoint realisierten Annotationsfunktionen, die erweitert und angepasst wurden, möglich.

Leider wurde es in [Cer03] nicht geschafft, die Fernsteuerung von Anwendungen generisch zu realisieren. Deshalb, und weil in der prototypischen Realisierung Powerpoint verwendet wurde, ist SPP für diese Arbeit nicht brauchbar. Dies liegt insbesondere daran, dass mit SPP nur unter Microsoft-Betriebssystemen gearbeitet werden kann, wobei in den Anforderungen dieser Arbeit Plattformunabhängigkeit gefordert ist. Auch ist nachteilig, dass wie oben auch schon für diese Softwaregruppe beschrieben, nur ein Dateiformat, in diesem Fall ppt, verwendet werden kann.

4.3.2 DoNut

Bei der in der Arbeit [Nag03] entwickelten Software handelt es sich um eine Unterstützung der Kommunikation zwischen Dozent und Studierenden in der Vorlesung (siehe auch [BBPW02]). Vom Dozenten werden Vortragsfolien speziell für den Einsatz mit DoNut vorbereitet, indem die einzelnen Folienseiten in Bereiche eingeteilt werden. Diese Folien werden den Studierenden zugänglich gemacht. Die Studierenden können nun während der Vorlesung auf die Stelle einer Folie klicken, zu der sie eine Frage haben und weitere Informationen zu ihrer Frage eingeben. Dem Dozenten werden diese Fragen auf seinem Laptop angezeigt. Er kann nun entscheiden, ob die Fragen wichtig sind und sofort geklärt werden sollen, oder ob bei der Nacharbeit der Vorlesung per e-mail oder auch zu Beginn der nächsten Vorlesung auf diese eingegangen wird. Auch erhält der Dozent so Feedback und sieht, welche Bereiche seines Vortrages verbessert werden sollten. DoNut ist insbesondere für große Vortragsveranstaltungen gedacht, um dort die Probleme der verbalen Kommunikation in großen Räumen bzw. Vorlesungen zu beheben.

Durch die völlig andere Zielsetzung in der Arbeit zu DoNut ist es im Sinne der in dieser Studienarbeit gestellten Anforderungen nicht geeignet. So ist es zwar möglich, Annotationen

im Kontext zu beispielsweise Folien zu erstellen und diese auch anderen zur Verfügung zu stellen, jedoch sind erhebliche Einschränkungen bezüglich der Positionierung der Beiträge vorhanden, da nur an den vom Dozenten vorgesehenen Stellen annotiert werden kann, was bedeutet, dass kein freier Bezug möglich ist. Auch der Ansatz, Kontextinformationen zu den Annotationen aus Bereichsdefinitionen hinter Folien zu ziehen, ist nicht brauchbar, da dies einen erheblichen Mehraufwand für Dozenten darstellt.

Die Speicherung der Annotationen (Fragen oder Anmerkungen) wird zwar unterstützt, jedoch nur zentral, wodurch keine Offline-Bearbeitung möglich ist. Auch wurde in der Arbeit [Nag03] wie schon in [Cer03] keine generische Lösung aufgetan, so dass eine sehr starke Anwendungsabhängigkeit in der prototypischen Realisierung vorliegt. Diese Abhängigkeit ist insbesondere durch die Integration von DoNut in SPP (siehe 4.3.1) entstanden.

Auch hat diese Integration zur Folge, dass wie schon in SPP nur das ppt-Format und nicht wie in den Anforderungen dieser Arbeit verlangt, auch andere Dateiformate unterstützt werden. Abgesehen davon sind Bewertungs- bzw. Trustmechanismen in DoNut nicht angedacht, da dies für die Kommunikation in der Vorlesung nicht unbedingt Sinn macht. Wie bei SPP ist eine Erweiterung von DoNut, wenn überhaupt, nur sehr schwer möglich.

4.3.3 Acrobat Writer

Nicht zu verwechseln mit dem Adobe Acrobat Reader, der kostenlos zur Verfügung steht, ist der Adobe Acrobat. Er ist das entsprechende Gegenstück, also ein Programm zur Erstellung von PDF-Dokumenten. Adobe Acrobat ist kostenpflichtig und neben weiterführenden Informationen unter [*Ad] erhältlich. Neben den normalen Funktionen einer Textverarbeitungssoftware bietet Acrobat ein relativ ausgefeiltes Annotationssystem für PDF-Dokumente. Es ist möglich, Notizen, Markierungen in Form von Pfeilen, Kreisen, Kästchen und anderen Formen zu machen sowie Textmodifizierungen vorzunehmen. Alle Annotationen dieser verschiedenen Annotationsmethoden lassen sich komplett ausblenden oder anhand von Filterfunktionen, wie beispielsweise einzelnen Personen oder auch anderen Kriterien, anzeigen.

Die Visualisierung der Notizen an sich im Kontext zum Dokument ist in Acrobat sehr gut gelöst. Jedoch kann leider nicht eine Textmarkierung als Grundlage (Anker) einer Notiz oder einer anderen Annotation genommen werden, sondern nur eine Position (ein Punkt). Die Markierungswerkzeuge und insbesondere ihre Visualisierung sind recht gut umgesetzt.

Da Acrobat kostenpflichtig ist, kann er für diese Arbeit nicht verwendet werden. Auch ist es nicht möglich, Acrobat zu erweitern bzw. anzupassen, wodurch eine teilweise Verwendung des Acrobats ebenfalls ausgeschlossen wird.

Es ist aber sinnvoll, sich bei der Realisierung der Annotationsfunktionalitäten in dieser Arbeit an denen des Acrobats zu orientieren. So ist eine ähnliche Umsetzung der Visualisierung von Annotationen und von Markierungen sinnvoll.

Zu überlegen ist, ob Unterstreichungen wie textuelle Annotationen ebenfalls als eine Art von Beiträgen gesehen werden sollten, welche veröffentlicht und ausgetauscht werden sollen. Wenn ja, wäre dies als eine weitere Erweiterungsmöglichkeit dieser Arbeit zu sehen.

Das Gleiche gilt auch für die oben schon beschriebenen Markierungsmöglichkeiten und das 'Editieren' als Annotation.

4.4 Informationsbereitstellungsdienste

In diesem Abschnitt soll auf einfache Protokolle, beziehungsweise Technologien eingegangen werden, die in erster Linie zum Bereitstellen, beziehungsweise zum Veröffentlichen von Informationen geeignet sind. Diese sind im Bereich dieser Arbeit relevant, da sie oft verwendet werden, um Lehrinhalte den Studierenden zur Verfügung zu stellen.

4.4.1 HTTP

Die Verwendung von HTTP, also das Bereitstellen von Lernmaterial und Informationen zu den Veranstaltungen über eine Webseite, sind momentan in der Lehre der de-facto-Standard. HTTP bedarf an dieser Stelle keiner weiteren Erläuterung, da davon ausgegangen wird, dass diese Technik dem Leser bekannt ist.

Das Problem bei Webseiten ist, dass kein Rücklauf von Informationen unterstützt wird und für den Offline-Betrieb nur auf zum Download bereitgestellte Dateien zugegriffen werden kann. Es sollte jedoch beachtet werden, dass durch die Stellung als de-facto-Standard der Webseiten diese auf jeden Fall als mögliche Datenquelle für Dokumente dieses Systems gesehen werden muss.

4.4.2 WIKI

Hinter dem Begriff 'Wiki Wiki' verbirgt sich, wie in [Huh02] nachzulesen ist, ein sehr einfaches, aber gerade dadurch bestechendes Konzept (siehe auch [*Wi]).

Über Wiki-Engines werden spezielle Webseiten zur Verfügung gestellt. Das Besondere daran ist, dass jeder anhand von dafür vorgesehenen Buttons jede Seite editieren, löschen oder ersetzen kann. Auch neue Seiten können problemlos von jedem hinzugefügt werden. Dies alles geschieht anhand einer vereinfachten HTML-Syntax.

Bei früheren Wiki-Engines war es nur möglich, Textinhalte darzustellen, was mit einer sehr begrenzten Darstellungsmöglichkeit einherging. Die heutigen Wiki-Engines erlauben mittlerweile den Einsatz von Style-Sheets und das Einbinden von Grafiken und Bildern. Auch bieten sie teilweise ein einfaches Rechtemanagement an. Interessant ist auch die Übersichtsfunktionalität zur Darstellung von Änderungen, wenn Seiten schon einmal besucht wurden und man sie sich später erneut ansieht.

Das Problem ist, dass nur eine allgemeine Sicht modifiziert wird, und nicht, wie bei dem hier zu entwickelnden System in den Anforderungen verlangt, die persönliche Sicht auf die Inhalte. Das ist aber von großer Bedeutung. So darf es laut den Anforderungen nicht möglich sein, beispielsweise ein Skript zu ändern. Jedoch unterstützt Wiki nicht explizit die Trennung von Annotationen und dem Dokument. Deshalb sind von Dokumenten getrennt abzuspeichernde Annotationen nur über einen Missbrauch der Wiki-Architektur

realisierbar, weswegen der Wiki-Ansatz zur Realisierung dieses Projektes nicht in Frage kommt. Außerdem sind Bewertungskonzepte und Trust-Management-Konzepte weder im Wiki-Konzept angedacht noch integriert. Auch ein Client, der mittels Replikation und Synchronisation das Offline-Arbeiten ermöglicht, ist wie bei den anderen Vertretern dieser Gruppe nicht vorhanden.

Die vereinfachte HTML-Syntax, welche bei Wiki verwendet wird, um den Nutzern das Erstellen und Editieren von HTML-Seiten zu erleichtern, wäre durchaus ein Konzept, welches zum Eingeben von Beiträgen eine einfach und schnell bedienbare Formatierungsunterstützung gewährleisten könnte.

4.4.3 RSS und Blogs

Ähnlich wie bei WIKI-Engines sind RSS und Blogs (Web Logs) ebenfalls Vertreter der Informationsbereitstellungsdienste mit Informationsrücklauf. Auf sie soll hier aber nur kurz hingewiesen werden, da sich dieselbe Problematik wie bei Wiki ergibt und diese Systeme damit nicht für diese Arbeit verwendbar sind. Weitere Informationen zu RSS, Blogs und Wiki finden sich unter [GJ03].

4.5 Zusammenfassung und Fazit

Um einen Überblick über die in diesem Kapitel betrachteten Systeme zu geben, sollen diese in den Abbildung 4.1, 4.2, 4.3 und 4.4 nochmals aufgelistet werden. Die wichtigsten Eigenschaften, mit welchen sie die Anforderungen dieser Arbeit nicht erfüllen, werden aufgeführt.

Dabei werden vorhandene für die Erweiterung oder die Realisierung dieser Arbeit in Frage kommende Konzepte ebenfalls nochmals kurz genannt. Kommen die Systeme generell als Datenquelle für Dokumente für das System in Frage, wurde der Begriff 'Dokumentenserver' notiert.

System	**KO-Kriterien**	**relevante Methoden und Konzepte**
BSCW,BSCL und Synergeia	Webbasierung, kein Open Source, grobgranulare Annotationen	'Thinking Types'; Dokumentenserver
ILIAS	Webbasierung, grobgranulare Annotationen	Lückentexte, Multiple-Choice-Fragen; Dokumentenserver
eclaus	Webbasierung, kein Open Source, keine Annotationen	

Abbildung 4.1: Tabellarische Zusammenfassung 'Verwandte Arbeiten - Lernplattformen'

System	**KO-Kriterien**	**relevante Methoden und Konzepte**
ToGather	nur ToGather-Dateiformat, kein asynchroner Annotationsaustausch	
Annotationserweiterung von Targeteam	Webbasierung, nur HTML-Format, grobgranulare Annotationen	
IAS-Notizeditor	Webbasierung, nur HTML-Format	

Abbildung 4.2: Tabellarische Zusammenfassung 'Verwandte Arbeiten - Annotationssysteme'

System	**KO-Kriterien**	**relevante Methoden und Konzepte**
SPP, DoNut	keine Plattformunabhängigkeit, nur ppt-Format, kein Open Source, keine separate Speicherung der Annotationen	
Acrobat	kein Open Source, nur pdf-Format, keine separate Speicherung der Annotationen	Visualisierung der Annotationen, Annotationstypen

Abbildung 4.3: Tabellarische Zusammenfassung 'Verwandte Arbeiten - Textverarbeitung / -visualisierung mit Annotationssystem'

System	**KO-Kriterien**	**relevante Methoden und Konzepte**
HTTP	webbasiert, kein Informationsrücklauf	
WIKI	webbasiert, keine Trennung von Annotation und Dokument	vereinfachte HTML-Syntax zur Eingabe von Annotationen
RSS und Blogs	webbasiert, keine Trennung von Annotation und Dokument	

Abbildung 4.4: Tabellarische Zusammenfassung 'Verwandte Arbeiten - Informationsbereitstellungsdienste'

Anhand der tabellarischen Zusammenfassung und den Erläuterungen in diesem Kapitel ist verdeutlicht worden, dass bisherige Systeme den Anforderungen dieser Arbeit nicht, oder nur teilweise gerecht werden. Auch sind die Abweichungen von den Anforderungen so stark, dass keines der Systeme komplett herangezogen und angepasst werden kann, um die geforderten Merkmale zu erfüllen.

Deshalb ist klar, dass ein neues System entwickelt werden muss.

Zur Realisierung eines Teiles in einem neu entwickelten System wurde, wie in 6.3 beschrieben, die Verwendung des BSCW-Systems in Betracht gezogen. Ebenso wurde ein FTP Ansatz in 6.3 näher erwogen.

Kapitel 5

Basissoftware

Wie schon im vorangegangenen Kapitel geschildert, erfüllen bisherige Systeme die Anforderungen dieser Arbeit nicht oder nur teilweise. Es ist also klar, dass ein neues System entwickelt werden muss. Zur Verwirklichung des Systems wird der Dokumentviewer Multivalent herangezogen (siehe 6.4 sowie [*MU]). Dieser soll deshalb in diesem Kapitel gesondert vorgestellt werden.

Auch auf das UniTEC-System wird in diesem Kapitel genauer eingegangen. UniTEC soll, wie schon in 1.4 geschildert, zur Realisierung des Systems verwendet werden.

5.1 Multivalent

Bei dem Dokumentviewer Multivalent handelt es sich um eine Open Source Software, welche an der Berkeley Univerity [PW] entstanden ist. Die Weiterentwicklung des Projektes wird über Sourceforge [*SO] koordiniert. Multivalent ist in Java realisiert.

Multivalent ist ein für den 'Einbenutzerbetrieb' entwickelter Dokumentviewer. Er unterstützt die Visualisierung verschiedenster Dokumenttypen. So können in der aktuellen Version unter anderem HTML, PDF sowie ZIP-Dateien angezeigt werden. Eine Erweiterung zur Unterstützung von Microsoft Word- und Microsoft PowerPoint-Dokumenten ist außerdem geplant. Neben weiteren Funktionalitäten bringt Multivalent verschiedene Annotationsfunktionen mit.

5.1.1 Architektur

Multivalent stützt sich auf eine Architektur, welche die Funktionalitäten von den Dokumentformaten trennt.

Rückgrat der Datenstruktur von Multivalent bildet dabei ein 'Dokument-Baum'. Der Dokument-Baum wird direkt aus der Struktur der Dokumente gebildet. Dies soll an dem stark vereinfachten Beispiel 'Buch' verdeutlicht werden. Wie im Bild 5.1 gezeigt, ist ein Buch meist mit Hilfe von verschiedenen Unterteilungskonzepten gegliedert. So wird ein Buch meist in Kapitel unterteilt. Diese wiederum in Abschnitte usw.. Diese rekursiv unterteilten Objekte werden in Multivalent direkt in den Dokumentbaum übernommen. Dies geschieht anhand von sogenannten 'internen Knoten'. Dabei wird das in Abbildung 5.1

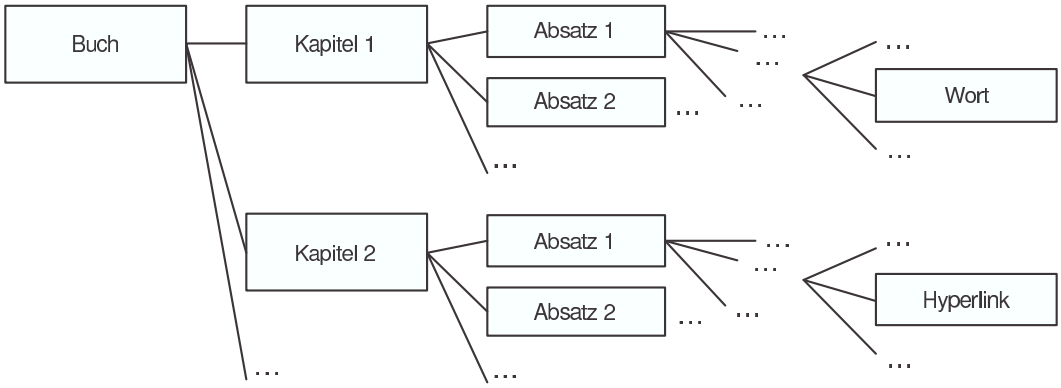

Abbildung 5.1: Beispiel Baum – Buch

angedeutete Konzept des Aufgliederns von Dokumenten in extremer Weise durchgezogen. Das Herunterbrechen wird bis hinunter zu den einzelnen Wörtern durchgeführt.

Die Vorgehensweise, den Dokumentbaum zu bilden bzw. das Dokument zu unterteilen, ist unabhängig vom Dokumenttyp immer dieselbe. Vom Dokumenttyp abhängige Daten werden dabei separat in sogenannten 'Leaves' abgelegt. Die Methode, dokumententypspezifische Daten von unspezifischen zu trennen, ermöglicht es, jegliche Funktionalitäten gegenüber des somit idealisierten Dokumentbaumes zu realisieren.

Dieses Konzept wird durch die Kapselung der Funktionalitäten in sogenannte 'Behaviors' strikt durchgezogen. Die Behaviors kommunizieren dabei nur über zwei Schnittstellen. Diese sind mittels der zwei in Multivalent internen Kommunikationsprotokolle festgelegt. Das eine Protokoll ist dabei für die Kommunikation entlang des Dokumentbaumes zuständig. Das zweite ist für die Kommunikation innerhalb einer Ebene des Baumes verantwortlich. Es organisiert die Kommunikation nach dem 'Round-Robin-Algotithmus'.

Behaviors können sehr mächtige Funktionen wie beispielsweise das Parsen eines Dokumenttypes, aber auch weniger mächtige Funktionen wie die Realisierung von Hyperlinks oder das Darstellen bzw. Verbergen von Menü-Einträgen oder anderen GUI (Graphical User Interface) Elementen bereitstellen.

Jedes Dokument wird jeweils in einem eigenen Multivalent-Browser-Fenster (MBF) geöffnet. Dabei stehen nur die Buttonleisten und Menüs zur Verfügung, welche für den angezeigten Dokumenttyp mit gültigen Behaviors realisiert sind. Welche das sind, ist in sogenannten Hubs definiert.

Hubs sind XML-Dateien mit einer für Multivalent festgelegten Struktur. Sie werden beim Start von Multivalent geladen.

5.1.2 Funktionsumfang

Im Folgenden sollen die wichtigsten Funktionen, die von Multivalent zur Verfügung gestellt werden, kurz aufgelistet werden. Danach wird die Annotationsfunktion 'Kurzer Kommentar' genauer beschrieben, da diese erweitert und angepasst (siehe 7.9) werden soll.

Medien Adapter: diese stellen die Unterstützung folgender Datentypen sicher:

- PDF 1.3
- TeX DVI
- ASCII
- scanned paper (zwei Arten: XDOC und PDA)
- UNIX manual pages (mit Inhaltsverzeichnis)
- HTML 3.2 mit fast allen CSS1
- Zip Dateien
- Verzeichnisse

Annotationsfunktionalitäten: Dokumenttyp unabhängige Annotationsfunktionen.

- Markern
- Hyperlink
- Kurzer Kommentar
- Schriftart
- Schriftgröße
- fett/kursiv/unterstrichen/...
- Schrift- / Hintergrundfarbe
- 'Text verschieben' – Anweisung
- 'Text ersetzen mit ...' – Anweisung
- 'Text löschen' – Anweisung

Werkzeuge: weitere Funktionen

- Favoriten-Verwaltung
- Präsentationsmodus
- Funktion zum Sortieren von Tabellen
- Browserfunktionalitäten - z.B.: Inhalte aus HTML-Formularen senden

Die Annotationsfunktion 'kurzer Kommentar' kann in Multivalent unabhängig vom Dokumentenformat erstellt werden. Es handelt sich um Fenster, die auf den Dokumenten frei positionierbar sind. In diesen kann der Benutzer einen kurzen Text eingeben. Die Annotationsdaten werden separat vom Dokument gespeichert. Dies geschieht anhand von oben schon beschriebenen Hubs (hier annos.hub). Dabei werden die Position, die Größe sowie der Text der Annotation gespeichert. Bei erneutem Anzeigen des Dokumentes wird die Annotation wieder auf derselben Position angezeigt. Die eindeutige Zuordnung der Annotationen ist dabei über die URL des Dokumentes realisiert.

Um einen feingranularen Bezug zum Dokument zu gewährleisten, werden die Annotationen immer mit der entsprechenden Position auf dem Dokument gespeichert, und

bei der nächsten Darstellung wieder an dieser Stelle angezeigt. Hierbei wird zwischen Seiten-orientierten und nicht-Seiten-orientierten Dokumenten unterschieden. Bei Seiten-orientierten Dokumenten, wie beispielsweise PDF-Dokumenten, werden die Seitenzahl und die Position der Annotation auf der Seite gespeichert. Bei nicht-Seiten-orientierten Dokumenten (z.B. HTML) wird nur mittels der Position bezogen auf das gesamte Dokument die Zuordnung gewährleistet.

Außerdem kann die Fenstergröße der Annotation frei festgelegt werden. Sie wird ebenfalls gespeichert und bei erneutem Anzeigen wieder abgebildet.

Eine Weitergabe der Annotationen kann nur durch eine spezielle Speicherfunktion geschehen. Die Annotationsdaten werden dabei in einem speziellen Dateiformat abgelegt. Nach dem Weitergeben können sie dann zusätzlich zu eigenen Annotationen angezeigt werden. Jedoch ist ein Editieren oder Übernehmen der fremden Annotationen nicht möglich.

5.2 UniTEC

Das UniTEC (Universal Trust Architecture for Electronic Commerce) - Projekt ist in Zusammenarbeit der Abteilung Verteilte Systeme des Instituts für Parallele und Verteilte Systeme der Universität Stuttgart mit der Firma Hewlett-Packard entstanden.

Beim UniTEC-System wurde der Ansatz verfolgt, Prozesse, die im realen Leben zu einer Kaufentscheidung eines Produktes führen, abzubilden. Hierfür können Empfehlungen zu Objekten, beispielsweise Produkten oder Firmen erstellt und anderen Personen zur Verfügung gestellt werden. Mit Hilfe von einer Anfrage an das UniTEC-System können Nutzer diese Empfehlungen dann auffinden und lesen. Die Empfehlungen sollen dann bei einer Kaufentscheidung unterstützen. Haben Nutzer mit dem entsprechenden Produkt eigene Erfahrungen gemacht, können sie die Empfehlungen bewerten. Die Bewertung einer Empfehlung beeinflusst das Vertrauen des Anfragenden gegenüber des Erstellers der Empfehlung. Anhand der hierdurch entstehenden Vertrauenswerte wird entschieden, an wen neue Anfragen gesendet werden.

Bei der Realisierung von UniTEC wurde sehr viel Wert auf die Unabhängigkeit des Systems von Anbietern gelegt. Es sollte nicht abhängig von Serverbetreibern sein. Auch die Anonymität der Nutzer sollte gewährleistet werden, um auf diese Weise möglichst objektive Hilfestellungen für Käufer zu erreichen.

UniTEC ist generisch realisiert worden. Deshalb kann es nicht nur für den Bereich des Handels verwendet werden.

Aus funktionaler Sicht besteht das UniTEC-System aus drei Teilen, dem Empfehlungsdienst, dem Vertrauensmodell und dem Identitätsmanager. Diese drei Teile des Systems sollen im Folgenden beschrieben werden. Im Anschluss daran wird dann auf die Architektur des UniTEC-Systems eingegangen.

5.2.1 Empfehlungsdienst

Der Kern des Systems ist der Empfehlungsdienst. Er setzt direkt auf der Peer-to-Peer-Komponente (siehe 5.2.4) des UniTEC-Systems auf. Der Empfehlungsdienst ist einerseits für die Erstellung und Verwaltung von Empfehlungen verantwortlich. Andererseits ist er für das Stellen sowie für das Beantworten von Anfragen nach Empfehlungen zuständig.

Eine Empfehlung ist ein Ratschlag zu einem bestimmten Sachverhalt, der von einer bestimmten Person gegeben wird. Abhängig vom beurteilten Objekt müssen Empfehlungen sehr unterschiedliche Daten aufnehmen können. Deshalb kann der Benutzer eine beliebige Anzahl von Feldern definieren. Jedes Feld wird dabei durch drei Eigenschaften beschrieben. Feldname, Feldtyp und Feldinhalt. Es sind folgende Feldtypen wählbar: Fließtext, Zahlen, Boolsche Elemente und Binäre Objekte.

Bei der Erstellung wird eine Empfehlung einer oder mehreren Kategorien zugeordnet. Diese Kategorien sollen die Suche nach Empfehlungen erleichtern. Durch sie kann das grobe Sachgebiet, in dem eine Empfehlung zu einem bestimmten Produkt oder einer bestimmten Dienstleistung gesucht wird, bestimmt werden.

Möchte man sich zu einem bestimmten Produkt oder einer Produktgruppe Empfehlungen ansehen, kann man an das System eine Anfrage stellen. Dies geschieht mit einer Volltextanfrage, wobei mehrere Suchbegriffe mit boolschen Operatoren verknüpft werden können. Eine Anfrage wird dabei stets auf eine Kategorie bezogen.

Das UniTEC-System liefert daraufhin die entsprechenden, im Peer-to-Peer-Netz auffindbaren Empfehlungen. Dies wurde dadurch realisiert, dass ein Knoten, der eine Anfrage von einem anderen Nutzer erhält, versucht, diese zu beantworten. Dazu werden als Erstes die vom Benutzer dieses Knotens selbst erstellten Empfehlungen nach passenden durchsucht. Werden keine gefunden, wird geprüft, ob mit Hilfe von temporär gespeicherten Empfehlungen anderer die Anfrage befriedigt werden kann. (Also Empfehlungen, die der Nutzer des Knotens selbst als Antwort auf Anfragen erhalten hat.) Kann die vom Anfragenden gewünschte Trefferzahl immer noch nicht erfüllt werden, stellt der Angefragte selbst eine Anfrage und leitet die Antworten an den Anfragenden weiter. Die Antwort enthält also stets die Anzahl an Treffern, die vom Anfragenden gewünscht wurde, sofern so viele Treffer im Peer-to-Peer-Netz überhaupt vorhanden sind.

5.2.2 Vertrauensmodell

Der zweite Teil des UniTEC-Systems ist für das Verwalten von Vertrauenswerten verantwortlich. Auch die Aktualisierung der Vertrauenswerte ist in diesem Teil realisiert. Eine ausführliche Beschreibung hierzu findet sich in [Aic03].

In der realen Welt werden auf der Suche nach Empfehlungen diejenigen Menschen befragt, denen man in dem entsprechenden Sachgebiet hohes Vertrauen entgegenbringt. Dieses Vorgehen wurde in UniTEC mit Hilfe einer qualitativen Abbildung des Vertrauens auf einen Vertrauenswert abgebildet. Dabei hat jeder Benutzer des Empfehlungsdienstes seine eigene Zuordnung eines Vertrauenswertes zu einem Empfehlenden bzw. einem Pseudonym (siehe 5.2.3).

Ein Empfehlender kann Empfehlungen in unterschiedlichen Kategorien erstellen. In den verschiedenen Kategorien können die Empfehlungen je nach Wissensstand des Empfehlenden unterschiedliche Qualität aufweisen. Deshalb wird im UniTEC-System jedem Empfehlenden in jeder Kategorie ein eigener Vertrauenswert zugeordnet.

Der Vertrauenswert ist dabei eine dynamische Komponente, die sich im Laufe der Zeit durch Enttäuschung oder Bestätigung von Vertrauen verändern kann. Eine solche positive oder auch negative Erfahrung kann der Nutzer dem System mitteilen, indem er bezogen auf eine Empfehlung eine Rückmeldung abgibt. Daraufhin wird der Vertrauenswert vom Vertrauensupdate-Algorithmus verändert.

Dabei kann eine Änderung in einer Kategorie Einfluss auf die Vertrauenswerte in anderen Kategorien haben. Beispielsweise wird eine positive Vertrauenswertänderung in der Kategorie 'Auto' eine positive Vertrauenswertänderung in der Kategorie 'Kleinwagen' des Empfehlenden bewirken, da diese beiden Kategorien thematisch zusammenhängen. Die Vertrauenswertänderung in den nicht direkt betroffenen Kategorien wird im Normalfall nicht so hoch ausfallen wie in der Kategorie, in welcher die Vertrauenswertänderung vom Nutzer ausgelöst wurde.

Welche Kategorien miteinander verwandt sind und wie hoch in diesen Kategorien die Veränderung des Vertrauenswertes ausfällt, ist in UniTEC mit Hilfe eines gerichteten Graphen definiert. Auch ist im Vertrauensmodell von UniTEC festgelegt, in welchem Wertebereich die Vertrauenswerte generell liegen können und wie sich positive oder negative Erfahrungen auf den Vertrauenswert der direkt betroffenen Kategorie auswirken. Außerdem ist eine 'Alterung' der Vertrauenswerte geplant. Ein Vertrauenswert soll dann automatisch sinken, wenn längere Zeit keine neuen Erfahrungen mit dem Empfehlenden gemacht werden.

5.2.3 Identitätsmanagement

Der Identitätsmanager bildet den dritten Systemteil. Mit ihm können die Benutzer ihre persönlichen Daten verwalten. Außerdem soll mit seiner Hilfe gewährleistet werden, dass die Benutzer stets die Kontrolle über die Weitergabe ihrer persönlichen Daten haben. Ausführliche Informationen zum Identitätsmanager befinden sich in [Wid03].

Wie schon beschrieben, wurde bei der Entwickling des UniTEC-Systems sehr viel Wert darauf gelegt, dass es anonym verwendet werden kann. Deshalb arbeiten die Nutzer in UniTEC nicht mit ihren eigenen Namen, sondern erhalten durch den Identitätsmanager die Möglichkeit, Empfehlungen unter dem Deckmantel von Pseudonymen zu erstellen. Ein Nutzer kann mehrere Pseudonyme besitzen.

Mit Hilfe des Identitätsmanagers können die Nutzer ihre Pseudonyme verwalten. Außerdem haben sie die Möglichkeit, gezielt einzelnen anderen Nutzern gegenüber ein Pseudonym aufzudecken.

Pseudonyme können außerdem einer oder mehreren Kategorien zugeordnet werden. Hierdurch soll eine Vergleichbarkeit zwischen den realen Interessensgebieten und den verwendeten Kategorien verhindert werden.

5.2.4 Architektur

Das UniTEC-System ist in Java realisiert und basiert auf einer Peer-to-Peer-Architektur. Durch das Verwenden eines verteilten Verzeichnisdienstes sind dabei keinerlei zentrale Komponenten erforderlich.

Die Software, die der UniTEC-Benutzer verwendet, wird als 'Agent' bezeichnet. Dieser bildet jeweils einen Knoten. Beim Start eines Agenten kann die Adresse eines anderen Agenten angegeben werden. Man tritt hierdurch in das bestehende Netz ein, in welchem sich der angegebene Agent befindet. Wird keine Adresse angegeben, 'erstellt' man ein neues Netz. Dieser erste Knoten in einem Netz wird auch mit Bootstrap-Node bezeichnet.

Um eine Kommunikation zwischen den Agenten in einem Netz zu ermöglichen, bei der Sender sowie Empfänger anonym bleiben, wurde im Rahmen des UniTEC-Projektes ein neues skalierbares Verfahren entwickelt. Hierfür wurde das Onion Routing Verfahren (siehe [Ter03]) herangezogen und erweitert.

Jeder Nutzer kann beim Start des Agenten entscheiden, ob er einen sogenannten Mix-Dienst zur Verfügung stellt. Dieser Mix-Dienst wird für das Routing bei der anonymen Kommunikation benötigt. So werden Nachrichten innerhalb des UniTEC-Systems über mehrere Stationen (Mixe) zum Ziel transportiert. Die Mixe eines Netzes bilden dabei einen sogenannten Chord-Ring, aus welchem beim Senden einer Nachricht die unterschiedlichen Mixe ausgewählt werden. Um die Anonymität zu wahren, wird außerdem jede Nachricht im Zwiebelprinzip ein- bzw. ausgepackt und verschlüsselt. Dadurch ist jedem Mix immer nur das nächste Teilziel bekannt.

Die für diese Kommunikation notwendigen Daten, wie beispielsweise der öffentliche Schlüssel der einzelnen Knoten, werden mit Hilfe eines verteilten Verzeichnisdienstes bereitgestellt.

Die lokal bei den einzelnen Knoten benötigten Daten werden auf dem jeweiligen Rechner des Benutzers in einer XML-Datenbank gehalten. Diese ist in die Software integriert.

Kapitel 6

Grobentwurf

Im Kapitel 4 wurde deutlich, dass bisherige Systeme den Anforderungen dieser Arbeit nicht oder nur teilweise gerecht werden. Deshalb ist ein neues System erforderlich. In diesem Kapitel soll anhand der im Kapitel 3 erarbeiteten Anforderungen eine Systemarchitektur entworfen werden. Dabei sollen die im letzten Kapitel vorgestellten Systeme miteingebunden und erweitert bzw. angepasst werden.

Zu Beginn soll auf grundlegende Dinge wie die Entwicklungsumgebung und die zu verwendende Grundstruktur der Architektur eingegangen werden. Danach werden die in 3.8 aufgeführten Funktionen gegliedert.

Daraufhin wird auf den Server und den Client sowie die Kommunikation zwischen diesen eingegangen. Zum Schluss dieses Kapitels wird die sich aus den getroffenen Entwurfsentscheidungen ergebende Architektur nochmals zusammenfassend beschrieben.

6.1 Grundlegendes

Teil der Anforderungen ist, dass das System plattformunabhängig ist und online sowie offline verwendet werden kann. Aus diesem Grund muss eine rein webbasierte Architektur ausgeschlossen werden.

Mit Hilfe von Java kann eine Plattformunabhängigkeit erreicht werden. Java-Applikationen können auf jeder Plattform ausgeführt werden, auf der eine Java-VM installiert ist. Java-VMs sind für Windows und Linux frei verfügbar. Da das System ohnehin nur mit Laptops und PCs sinnvoll verwendet werden kann und nicht etwa mit PDAs oder ähnlichen Geräten, ist bei der Verwendung von Java die geforderte Unabhängigkeit ausreichend gegeben.

Im Weiteren muss zunächst die Frage geklärt werden, was für eine Architektur bezüglich des Datenaustausches verwendet werden soll. Es stehen grundsätzlich eine Client-Server-Architektur und eine Peer-to-Peer-Architektur zur Diskussion. Vorteil einer Peer-to-Peer-Architektur ist, dass kein Server zur Verfügung stehen und gewartet werden muss. Jedoch kann mit einer Peer-to-Peer-Architektur die in den Anforderungen aufgeführte Zugangskontrolle zum Schutz der Informationen im System nur mit sehr großem Aufwand realisiert werden. Dies gilt auch, weil sichergestellt werden muss, dass nachvollziehbar ist, wer für

eine Veröffentlichung im System verantwortlich ist. Außerdem ist es schwierig, mit einer Peer-to-Peer-Architektur zu gewährleisten, dass jederzeit auf alle Dokumente zugegriffen werden kann. Daher fiel die Entscheidung, eine Server-Client-Architektur zu verwenden.

6.2 Funktionale Gliederung

In diesem Abschnitt soll eine erste logische Gliederung der erforderlichen Funktionalitäten vorgenommen werden. Dafür werden die in 3.8 aufgeführten Funktionalitäten in Funktionsgruppen zusammengefasst. Manche funktionalen Anforderungen (FA) müssen dabei allerdings in mehrere Funktionsgruppen aufgenommen werden. Zum Beispiel, wenn eine Gruppe für die Visualisierung einer Funktionalität und eine andere für die Bereitstellung der eigentlichen Funktionalität verantwortlich ist.

Anhand der Gruppe 'Dokumenten-Viewer' sollen die 'Frontend'-Funktionalitäten, wie in der Abbildung 6.1 ganz links dargestellt, gegliedert werden. Diese Gruppe beinhaltet zum einen die Untergruppe 'Dokumentendarstellung', die durch FA70 - 'formatunabhängige Dokumentenanzeige' und FA75 - 'Skriptstellensuche' bedingt ist. Zum anderen gehören FA110 - 'Sichten', FA120 - 'persönliche Sicht', FA125 - 'Fragen' und FA 170 - 'Diskussion' als Untergruppe 'Annotationsdarstellung' mit dazu. Die Untergruppe 'Annotationserstellung' bildet die Zusammenfassung der FA80 - 'Annotationen erstellen', FA90 - 'Annotationsklassifizierung', FA100 - 'Annotationen veröffentlichen' und FA125 - 'Fragen'. Außerdem gehören die Untergruppen 'Bewertungserstellung' (siehe FA130 - 'Bewertungen') sowie 'Verknüpfungserstellung und -anzeige' (siehe FA140 - 'Verknüpfungen erstellen', 150 - 'Verknüpfungen verfolgen' und FA160 - 'Verknüpfungen anzeigen') wie in Abbildung 6.1 dargestellt mit in die Gruppe 'Dokumenten-Viewer'.

Den Forderungen in FA10 - 'Veranstaltungen und Dokumente anzeigen' nachkommend, soll mit der Gruppe 'Dokumenten-Explorer' gewährleistet werden, dass der Nutzer einfach über die vorhandenen Veranstaltungen und Dokumente navigieren kann und verschiedenste Funktionen von ihm angestoßen werden können. Hierzu werden außerdem FA20 - 'Veranstaltungen hinzufügen / veröffentlichen', FA60 - 'Dokumentenabonnement', FA35 - 'Feedback - übernehmen von Teilen einer abgeschlossenen Veranstaltung', FA37 - 'Literaturverzeichnis' und FA135 - 'Annotationsabonnement' mit in diese Gruppe eingegliedert. Dies gilt außerdem für FA215 - 'Daten sichern / auslagern' und FA230 - 'Löschen lokaler Daten'.

Die in Abbildung 6.1 mit 'Datenaustausch' benannte Funktionsgruppe ist bei jedem Nutzer lokal für die Kommunikation mit dem Server erforderlich (siehe FA20 - 'Veranstaltungen hinzufügen / veröffentlichen', FA30 - 'Dokumente hinzufügen / veröffentlichen', FA40 - 'Dokumente herunterladen', FA60 - 'Dokumentenabonnement', FA100 - 'Annotationen veröffentlichen', FA135 - 'Annotationsabonnement'). Für den Austausch werden weiter Mechanismen zur Synchronisation und zum Filtern benötigt, wie in FA50 - 'Veranstaltungs- / Dokumentensynchronisation' und FA190 - 'Annotationssynchronisation' aufgeführt. Aus diesem Grund wurde in die Gruppe 'Datenaustausch' die Untergruppe 'Synchronisation

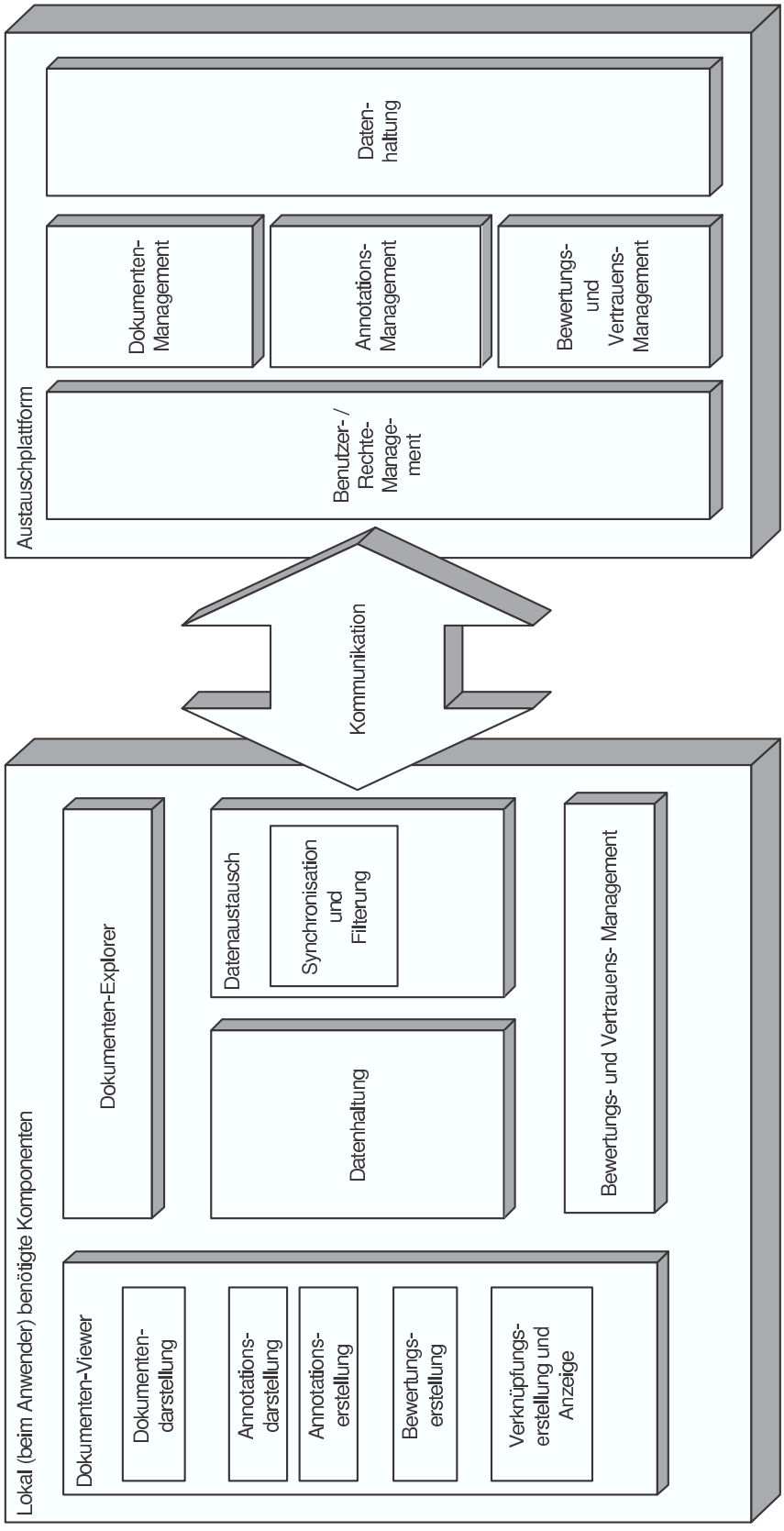

Abbildung 6.1: Funktionale Gliederung

und Filterung' aufgenommen.

Um das Filtern bewerkstelligen zu können, sind die in FA130 - 'Bewertungen' und FA 180 - 'Vertrauensbildung und -management' aufgeführten Funktionalitäten erforderlich. Diese sollen zur Gruppe 'Bewertungs- und Vertrauensmanagement' zusammengefasst werden, wie rechts unten im linken Teil der Abbildung 6.1 geschehen.

FA210 - 'lokale Datenhaltung' soll, wie in Abbildung 6.1 im Zentrum des linken Teiles zu sehen ist, eine eigene Gruppe bilden.

Der Server muss die in den Szenarien und den daraus entwickelten Anforderungen in FA1 - 'Online/Offline', FA220 - 'Austauschplattform', FA240 - 'löschen globaler Daten' und FA200 - 'Rechte-/Benutzerverwaltung und Zugangskontrolle' beschriebenen Funktionalitäten zur Verfügung stellen. Diese erforderlichen Funktionalitäten können, wie in der Abbildung 6.1 im rechten Teil dargestellt, mittels folgender Funktionalitätengruppen gegliedert werden: Benutzer-/Rechtemanagement, Dokumentenmanagement, Annotationsmanagement, Bewertungs- und Vertrauensmanagement sowie Datenhaltung. Die Benutzer- und Rechteverwaltung ist dabei allen anderen Funktionalitäten 'vorangeschaltet', da sie den Zugriff auf diese regelt. Als Rückgrat steht, alle anderen Funktionsgruppen des Servers unterstützend, die Gruppe 'Datenhaltung', wie sie in Abbildung 6.1 ganz rechts zu sehen ist.

6.3 Server

In erster Linie müssen auf dem Server Dateien (Dokumente) und Annotationen gespeichert werden. Für die Realisierung der Speicherung und Verwaltung von Dokumenten sind folgende Alternativen möglich: Verwenden eines BSCW-Systems, die Verwendung eines FTP-Servers oder der Entwurf eines neuen Server-Systems.

Wird ein BSCW-System als Server bezüglich der Dokumentenspeicherung verwendet, bringt dies den Vorteil, dass Dokumente, die sich schon auf verwendeten BSCW-Systemen befinden, mitgenutzt werden können. Allerdings muss zur globalen Speicherung der Annotationen eine separate Methode gefunden werden. Eine Möglichkeit wäre, einen 'Annotea-Server' zu verwenden, wie es schon in der Arbeit [Sau03] gemacht und beschrieben wurde. Vorteilhaft an der Verwendung eines 'Annotea-Server' wäre, dass die Konfiguration eines solchen sehr einfach ist und das Annotea-Protokoll einen W3C-Standard darstellt (siehe auch [*Ana]).

Allerdings gestaltet sich dann die Realisierung der in FA200 geforderten Rechte- und Benutzerverwaltung sowie der Zugangskontrolle zu den Daten des Systems sehr schwierig. Bei BSCW und Annotea ist eine Benutzer- und Rechteverwaltung zwar vorhanden, jedoch müssten die Benutzerinformationen bezüglich dieser beiden Systeme immer angeglichen werden.

Außerdem würden mehrere verschiedene Kommunikationsprotokolle zwischen Client und Server, eines für den 'Annotea-Server' und eines für die Kommunikation mit dem BSCW-System, erforderlich.

Die serverseitige Installation und Konfiguration des Systems würde außerdem unnöti-

gerweise verkompliziert. Eine Trennung der eigentlichen Verwaltungsfunktionalitäten und der Speicherung, also eine Aufteilung in einen Applikationsserver und einen Datenhaltungsserver, wäre ebenfalls nicht möglich, was bedeutet, dass dann eine Erweiterung der Benutzer und Rechteverwaltung, wie in 3.3 gefordert, nahezu unmöglich wird.

Verwendet man zur Speicherung der Dateien einen FTP-Server, bringt dies den Vorteil mit sich, dass der Implementierungsaufwand durch Zurückgreifen auf Open Source Client- und Server-Software sehr gering ausfällt. Jedoch trifft man auf dieselbe Problematik wie schon bei BSCW. Die Realisierung der Zugriffskontrolle ist mit Hilfe der Zugriffsrechte des Server-Dateisystems zwar möglich, führt aber ebenfalls zu einer Zweigleisigkeit. Durch diese bedingt entstehen eine sehr komplexe Architektur und Schwierigkeiten, was die Erweiterbarkeit angeht. Auch würden wieder mehrere verschiedene Kommunikationsprotokolle zwischen Client und Server benötigt.

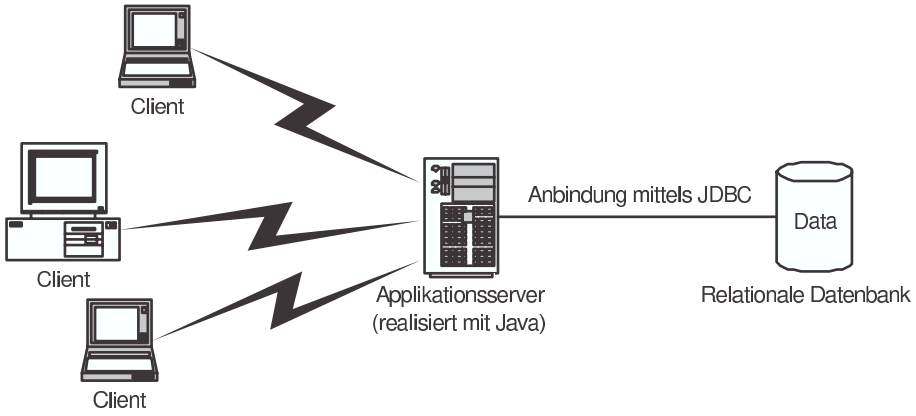

Abbildung 6.2: Trennung von Applikationsserver und Datenbank

Wegen der großen Probleme bei den ersten beiden Realisierungsmöglichkeiten und um allen Anforderungen gerecht zu werden, wurde beschlossen, die dritte Variante zu wählen. Es wurde also ein Applikationsserver neu entworfen. Dieser soll, wie in der Mitte der Abbildung 6.2 dargestellt, die Funktionalitäten, die serverseitig erforderlich sind, kapseln. Für die eigentliche Datenspeicherung wird dann, wie in 6.2 auf der rechten Seite gezeigt, auf ein Datenbank-System zurückgegriffen. Hierdurch ist der Aufbau ausreichend modular, sodass Anpassungen und Erweiterungen, wie sie in 3.3 gefordert wurden, möglich sind.

Zur Realisierung des Datenbanksystems gibt es zum einen die Möglichkeit, eine XML-Datenbank zu verwenden. Zum anderen könnte eine relationale Datenbank verwendet werden.

Für eine XML-Datenbank spricht, dass Multivalent (siehe 6.4.1 bzw. 5.1) die Annotationsdaten im XML-Format in Hub-Dateien vorhält.

Die Wahl fiel jedoch auf eine relationale Datenbank, da diese für hohe Nutzerzahlen und parallel anfallende Anfragen optimiert sind. Außerdem können relationale Datenbanken über JDBC (Java DataBase Connectivity), also eine klar definierte Schnittstelle, in Java Applikationen eingebunden werden. Deshalb wurde an dieser Stelle auch in Kauf genommen, dass die Annotationsdaten aus der 'dynamischen' XML-Struktur in eine starre

Tabellenstruktur umgewandelt werden müssen.

Mit früheren relationalen Datenbanksystemen war es zwar möglich Dateien zu verwalten, jedoch wurde diese Funktionalität in der Praxis nicht verwendet, da die Leistungsfähigkeit der Systeme in diesem Bereich relativ schlecht war. Mitlerweile sind Datenbanksysteme größtenteils in der Lage leistungsfähig Dateien zu verwalten. Da die Dateigröße jedoch meist begrenzt ist und immer noch nicht alle relationalen Datenbanksysteme effizient Dateien verwalten können wurde beschlossen, die Dokumente nicht im Datenbanksystem zu speichern. Sie sollen im Dateisystem abgelegt werden. Nur die Metadaten der Dokumente sind also in die relationale Datenbank aufzunehmen.

Die für die Benutzer und Rechteverwaltung nötigen Daten sollen ebenfalls in der relationalen Datenbank gespeichert werden.

6.4 Client

In diesem Abschnitt soll nun der Client-Teil des Systems, in der Abbildung 6.1 links dargestellt, betrachtet werden. Dabei soll darauf eingegangen werden, wie die Funktionengruppe 'Dokumenten-Viewer' realisiert werden soll. Weiter wird diskutiert, wie die Gruppe 'Bewertungs- und Vertrauensmanagement' mittels UniTEC (siehe 5.2) realisiert werden kann, und welche Folgen die Verwendung von UniTEC für die System-Architektur hat. Darauffolgend wird dann noch angesprochen, wie die Datenhaltung lokal bewerkstelligt werden soll.

6.4.1 Dokumenten-Viewer

Wie schon aus Abbildung 6.1 ersichtlich, muss der Dokumenten-Viewer eine relativ große Zahl an Funktionalitäten bereitstellen. Insbesondere soll es möglich sein, Dokumente unabhängig von ihrem Dateiformat darzustellen. Zu den angezeigten Dokumenten sollen mit feingranularem Bezug Annotationen erstellt und angezeigt werden können.

Diese Funktionen bietet Multivalent (siehe 5.1) bereits teilweise. Jedoch sind in Multivalent die verschiedenen Sichten (persönliche und nicht-persönliche Sicht) noch nicht realisiert. Diese Funktionen müssen also genauso wie das Erstellen von Bewertungen zu Annotationen und Dokumenten noch hinzugefügt werden. Durch die modulare Architektur von Multivalent ist das Einbinden dieser Erweiterungen jedoch relativ einfach möglich. Dies gilt insbesondere hinsichtlich der Anpassung der vorhandenen Annotationsfunktion 'kurzer Kommentar', die in 5.1.2 bereits beschrieben wurde. Deshalb wurde entschieden, Multivalent zur Realisierung der Client-Seite des Systems mitzuverwenden.

6.4.2 Bewertungs- und Vertrauens-Management

Für die Realisierung des Bewertungs- und Vertrauens-Managements sollen die Algorithmen und Funktionalitäten von UniTEC herangezogen werden, wie in der Aufgabenstellung nahegelegt und in 1.4 erläutert. Wie schon in 5.2.4 erläutert, ist UniTEC in Java implementiert und basiert auf anonymer Peer-to-Peer-Kommunikation. Zum Aufbau eines Net-

zes ist ein erster Knoten erforderlich, der weiteren, die in dasselbe Netz eintreten möchten, bekannt sein muss. Dies stellt kein Problem dar, da der Server als 'erster' Knoten - so genannter Bootstrap - fungieren kann, wie in Abbildung 6.3 oben verdeutlicht.

Der Peer-to-Peer-Ansatz von UniTEC birgt zunächst den Nachteil, dass dadurch das System zu einer gemischten Architektur kommt. Das System greift seitens der Annotationen und Dokumente sowie der Benutzer- und Rechteverwaltung auf eine Server-Client-Architektur zurück, wie in Abbildung 6.3 durch die einfachen, dicken Linien verdeutlicht. Weiter weisen die Clients und der Server bezüglich der Bewertungen und der Vertrauenswerte eine Peer-to-Peer-Architektur auf. Dieser Sachverhalt ist in 6.3 unten mit Hilfe der Rauten in den Clients verdeutlicht. Diese wurden, um die Peer-to-Peer-Kommunikation darzustellen, mit dem 'Bootstrap-Node' und untereinander verbunden. Die Clients wurden mit Ausnahme des unten in Abbildung 6.3 auffindbaren stark vereinfacht visualisiert.

Vorteil an dieser Mischarchitektur ist, dass der Applikationsserver dadurch stark entlastet wird. Ein weiterer Vorteil ist auch, dass Bewertungen und somit die Vertrauensbildung völlig anonym vonstatten gehen können, obwohl die Nutzer im System, zumindest gegenüber dem Applikationsserver, nicht anonym sind. Mehr dazu im Anschnitt 7.8.

6.4.3 Lokale Datenhaltung

Lokal eine XML-Datenbank zur Speicherung der benötigten Daten zu verwenden, scheint naheliegend, da Multivalent Annotationen mittels XML-Dateien speichert (siehe 5.1) und UniTEC schon eine XML-Datenbank beinhaltet, mittels der Bewertungen und Vertrauenswerte gespeichert werden. Auch ist lokal das Verwenden einer relationalen Datenbank nicht sinnvoll, da dadurch ein höherer Installations- und Konfigurationsaufwand durch die zusätzliche Installation eines Datenbanksystems entstehen würde, welcher den Nutzern nicht zugemutet werden soll.

Jedoch würde das Verwenden der UniTEC-XML-Datenbank für alle lokal anfallenden Daten den Anforderungen widersprechen, da in 3.3 gefordert ist, dass bezüglich UniTEC eine starke Kapselung vorgenommen werden soll.

Aus diesem Grund wurde entschieden, auf Client-Seite zur Speicherung zweigleisig zu fahren. Die für das Bewertungs- und Vertrauensmanagement lokal anfallenden Daten sollen also, mittels der UniTEC-eigenen XML-Datenbank gespeichert werden, wie in Abbildung 6.3 im ganz unten dargestellten Client zu sehen ist. Alle weiteren lokal zu speichernden Daten sollen hingegen mit Hilfe der bei Multivalent vorhandenen Routinen zur Speicherung mittels XML-Dateien abgelegt werden (siehe Abbildung 6.3). Dies gilt auch für die Metadaten der Dokumente. Die Dokumente selbst sollen in einem Verzeichnis gespeichert werden.

Von der Möglichkeit, eine als Open Source zur Verfügung stehende XML-Datenbank zu verwenden, wurde abgesehen, da dies einen hohen Aufwand zur Anpassung von Multivalent und UniTEC nach sich ziehen würde, der den Zeitrahmen dieser Studienarbeit weit überschritten hätte (siehe 9).

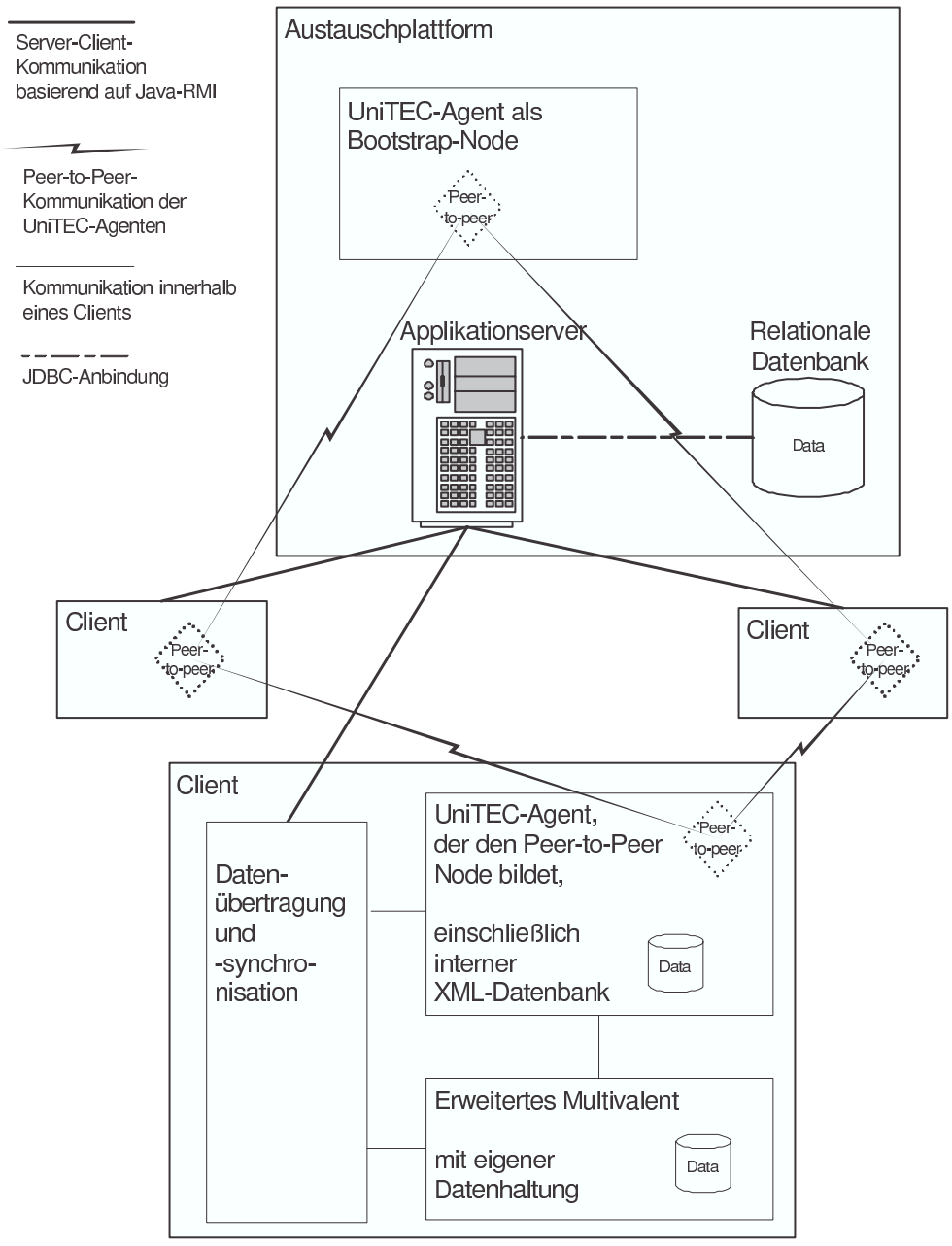

Abbildung 6.3: Server mit Bootstrap und mehreren Clients

6.5 Kommunikation

Wie zu Anfang dieses Kapitels in 6.1 schon geschildert, soll der Client in Java implementiert werden. Außerdem ist wegen der in 6.3 beschriebenen Entscheidungen die Implementierung des Applikationsservers ebenfalls in Java sinnvoll, da dadurch auch auf Serverseite Plattformunabhängigkeit erreicht wird. Aus diesem Grund bietet sich zur Realisierung der Kommunikation zwischen Server und Client in besonderer Weise Java-RMI an. Mittels Java-RMI können die hier benötigten Kommunikationsfunktionalitäten sehr einfach und schnell realisiert werden. Auch ist mit Hilfe von Java-RMI die geforderte 'Firewall-Tauglichkeit' zumindest ausreichend gewährleistet, da an Universitäten 'private Netzwerke' oder 'virtuelle private Netzwerke' zu erwarten sind. Auch können die zur Kommunikation verwendeten Ports mittels Java RMI frei festgelegt werden. Das ermöglicht ein 'Umgehen' von Firewalls, falls eine solche doch vorhanden ist. Das Umgehen kann geschehen, indem von der Firewall nicht gesperrte Ports gewählt werden. Allerdings sollte diese Möglichkeit natürlich nur angewendet werden, wenn keine anderen Möglichkeit gegeben sind.

Auch bieten andere Technologien, wie beispielsweise Java Sockets, gegenüber Java-RMI keine hier relevanten Vorteile. Deshalb wurde entschieden, Java-RMI für die Kommunikation zu verwenden.

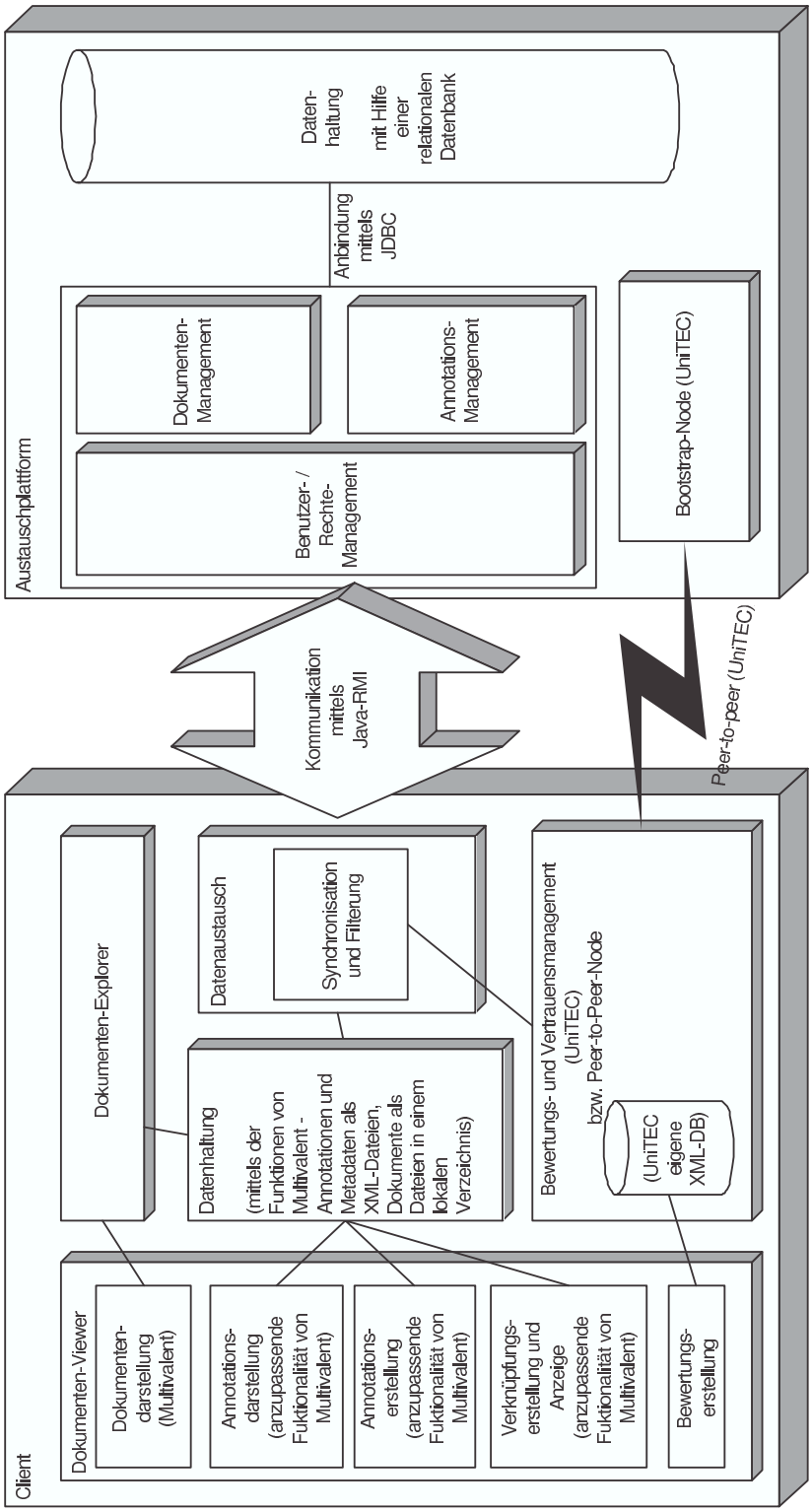

Abbildung 6.4: Funktionale Gliederung und Kommunikation bei Verwendung von Multivalent und UniTEC

6.6 Ergebnisse

In der Abbildung 6.4 sind zusammenfassend nochmals alle Teile des Systems dargestellt.

Auf der linken Seite ist der Client zu sehen. Das Funktionspaket 'Dokumenten-Viewer' wird, wie dargestellt, zu einem großen Teil von den Funktionalitäten von Multivalent abgedeckt. Dabei müssen die 'Annotationsdarstellung', die 'Annotationserstellung' und die 'Verknüpfungserstellung und Anzeige' wie dargestellt noch angepasst und erweitert werden. Die 'Bewertungserstellung' muss komplett neu realisiert und in Multivalent eingebunden werden.

Auf den oben neben dem 'Dokumenten-Viewer' in 6.4 dargestellten 'Dokumenten-Explorer wird in 7.1 näher eingegangen. Die darunter zu sehende 'Datenhaltung' wird, wie in 6.4 beschrieben, mittels der Funktionalitäten von Multivalent realisiert. Hierbei wird die Speicherung der Bewertungen und der Vertrauensdaten nicht miteingeschlossen. Diese wird mittels der im Block 'Bewertungs- und Vertrauensmanagement' dargestellten in UniTEC enthaltenen XML-Datenbank realisiert. Hierdurch wurde, wie durch die Verbindungslinien verdeutlicht, die Kapselung von UniTEC bestmöglich vollzogen.

Die Peer-to-Peer-Kommunikation der Clients wurde mittels des 'Blitz-Pfeiles' angedeutet. Die Server-Client-Kommunikation hingegen ist mit einem Doppelpfeil zwischen dem Funktionsblock 'Datenaustausch' und dem Applikationsserver visualisiert. Auf den 'Datenaustausch' wird in 7.6 sowie in 7.7 näher eingegangen.

Die auf der rechten Seite der Abbildung 6.4 gezeigte Austauschplattform setzt sich, wie schon erläutert, aus dem Applikationsserver, der mittels JDBC angebundenen relationalen Datenbank und dem durch UniTEC realisierten 'Bootstrap-Node' zusammen.

Der Applikationsserver besteht aus einem vorgeschalteten 'Benutzer- / Rechte- Management' und dem in 6.3 geschilderten Dokumentenmanagement sowie dem Annotationsmanagement.

Kapitel 7

Feinentwurf

Der im letzten Kapitel erstellte Grobentwurf soll nun verfeinert werden. Es ist jedoch klar, dass hier nicht auf jedes Entwurfsdetail eingegangen werden kann, da dies den Rahmen dieser Ausarbeitung sprengen würde. Deshalb fiel die Entscheidung, nur die für diese Arbeit besonders wichtigen Entwurfsaspekte zu beschreiben.

Auch soll aus Zeitgründen, und da es sich nicht um Kernfunktionalitäten handelt, die Zugangsbeschränkung sowie die Benutzer- und Rechteverwaltung im Folgenden ausgeklammert werden. Durch die im Grobentwurf getroffenen Entscheidungen ist eine spätere Realisierung dieser Funktionalitäten ohnehin ohne Einschränkungen möglich. Außerdem stellt das Ausklammern dieser Funktionalitäten zum jetzigen Zeitpunkt die in 1.4 geschilderten Entscheidungen nicht in Frage.

Auf den Datenbankentwurf soll ebenfalls nicht tiefgreifender eingegangen werden, da dieser sich verhältnissmäßig einfach gestaltet. So sind nur wenige Tabellen, je eine für Annotationen, die Metadaten der Dokumentdateien sowie die Benutzer und deren Rechte nötig. Hierbei erhalten die Annotationen und die Dateien eindeutige fortlaufende Nummern, um sie global identifizieren zu können. Siehe hierzu 7.2.

7.1 Dokumenten-Explorer

Das Funktionspaket 'Dokumenten-Explorer' soll den Nutzern ein Navigationswerkzeug an die Hand geben. Dabei soll, wie in 3.1 gefordert, eine den Nutzern bekannte Umgebung verwendet werden. Dies soll geschehen, indem zum Auflisten der Veranstaltungen und der Dokumente eine Darstellung verwendet wird, wie sie für Dateien und Verzeichnisse in Datei-Explorern, beispielsweise in MS Windows, üblich sind.

Es werden dabei die lokal sowie die global verfügbaren Veranstaltungen und Dokumente angezeigt und entsprechend kenntlich gemacht. Auch wird als Eigenschaft der Veranstaltungen dargestellt, ob diese mit einem Dokumentenabonnement vom Nutzer versehen wurden. Das gleiche gilt bezüglich der Dokumente im Zusammenhang mit Annotationsabonnements.

Mittels Anklicken eines Dokumentes soll es möglich sein, ein solches zu öffnen, indem ein Multivalent-Browser-Fenster automatisch geöffnet wird. Die Funktionalitäten 'erstel-

len' und 'aufheben' von Annotationsabonnements bzw. Dokumentenabonnements, 'hinzufügen', 'löschen' und 'sichern' von Veranstaltungen und Dokumenten sowie das Veröffentlichen dieser sollen ebenfalls mit Hilfe des Dokumenten-Explorers leicht zugänglich sein. So soll beispielsweise durch Auswählen eines Dokumentes und Anwählen der entsprechenden Funktion 'Annotationen abonnieren' im Menü das Erstellen eines Annotationsabonnements durchführbar sein. Ebenso auch beim Abbestellen dieses Abonnements und auch bei den anderen Funktionen.

Die Dateien der eigentlichen Dokumente liegen alle in einem Verzeichnis. Jedoch soll dieses, wie schon beschrieben, in Baumstruktur, Veranstaltungen zugeordnet, mit weiteren Metadaten aufgelistet werden. Um dies zu ermöglichen, soll ein virtueller Dateibaum erzeugt und gespeichert werden. Hierfür werden die Metadaten der Dokumente bzw. Veranstaltungen zur Laufzeit des Systems in Objektbäumen gehalten. Beim Ende der Arbeit mit dem System werden diese dann in den in 6.4 schon beschriebenen Hub-Dateien, wie bei Multivalent üblich, abgelegt. Auf diese Weise können dann auch die öffentlichen Metadaten der verfügbaren, aber noch nicht heruntergeladenen Dokumente lokal gehalten und visualisiert werden.

7.2 Globale Eindeutigkeit

Annotationen müssen Dokumenten zugeordnet werden, sodass sie nach dem Laden des Dokuments mitangezeigt werden können. In Multivalent wird zu jedem Dokument, zu welchem Annotationen erzeugt werden, eine Hub-Datei erzeugt, in welcher die Annotationen abgelegt werden. Es wird mittels des zugehörigen Dateinamens und Dateipfades (der URL) gewährleistet, dass die Annotationen eindeutig einem Dokument zugeordnet werden können. Es stellt sich also die Frage, ob dieses Konzept, welches lokal die eindeutige Zuordnung gewährleistet, auch zur global eindeutigen Zuordnung herangezogen werden kann.

Dies wäre möglich, wenn jeder Nutzer lokal ab einem bestimmten 'root'-Verzeichnis dieselbe Verzeichnisstruktur wie auf dem Server vorliegen hätte, sodass dadurch ein lokal eindeutiger Dateipfad und Dateiname auch global eindeutig wäre. Allerdings müsste dann gewährleistet werden, dass nicht von Hand, sondern nur mittels des Systems auf dieses Verzeichnis zugegriffen würde, da nur so Veränderungen am Pfad der Dateien verhindert werden könnten.

Dies ist kaum möglich, weshalb diese Methode sehr unsicher ist, da durch Verändern der Pfade die Eindeutigkeit der Zuordnung zerstört würde. Außerdem ist dieses Vorgehen sehr unflexibel. Deshalb wurde entschieden, lokal weiter mit der Zuordnung, die in Multivalent verwendet wird, zu arbeiten, global jedoch eine andere, zweite Methode zu verwenden. Hierzu wird vom Server beim Hochladen eines Dokumentes sowie beim Hochladen einer Annotation diesen jeweils eine eindeutig identifizierende Nummer zugeteilt. Mittels dieser können dann andere Nutzer Dokumente und zugehörige Annotationen zuordnen.

Die globale Eindeutigkeit von Annotationen ist dabei aus zwei Gründen sinnvoll. Zum einen wird dadurch gewährleistet, dass Bewertungen eindeutig zugeordnet werden können und zum anderen ist dies beim Synchronisieren wichtig, damit nur die Annotationen

heruntergeladen bzw. hochgeladen werden, die noch nicht transferiert wurden. Diese hier gewählte Methode macht außerdem möglich, dass die Dokumente in einem Verzeichnis liegen und mittels einer virtuellen Baumstruktur angezeigt werden können.

7.3 Bewertungen

Eine Bewertung muss, wie in 3.6 beschrieben, innerhalb kürzester Zeit durchführbar, aber trotzdem möglichst aussagekräftig sein. Außerdem müssen Bewertungen vergleichbar sein. Wie in 7.5 beschrieben, ist dies erforderlich, damit mit Hilfe der Bewertungen Vertrauenswerte ermittelt werden können. Wegen dieser Gründe ist eine klare Definition der Form von Bewertungen erforderlich.

Um das Verhältnis von Aussagekraft und Erstellungsaufwand einer Bewertung zu optimieren, ist eine gewisse Anzahl von Kategorien in Betracht zu ziehen. Sind es zu wenige, verliert die Bewertung an Aussagekraft. Sind es zu viele, dauert eine Bewertung zu lange. Im Entscheidungsprozess kamen folgende zwei Möglichkeiten in die engere Auswahl.

Möglichkeit 1: 5 Kategorien:

- Relevanz (zum Thema passend)
- Korrektheit
- Aufbereitung (Übersichtlichkeit/Gliederung)
- Verständlichkeit (Formulierung/Erklärung)
- Hilfreich (für das Verständnis des Stoffs)

Möglichkeit 2: 3 Kategorien:

- Relevanz
- Korrektheit
- Hilfreich

Es wurde entschieden, die zweite Möglichkeit zu verwenden, da bei der ersten die Kategorien sich teilweise überschneiden bzw. unterschiedlich interpretiert werden und deshalb Bewertungen dieser Art nicht oder nur schlecht verglichen werden können.

Eine Bewertungskategorie 'Gliederung' ist bei Dokumenten sinnvoll, bei Annotationen jedoch überflüssig. Verschiedene Bewertungsformen für Dokumente und Annotationen würden deshalb Sinn machen. Es wurde jedoch die Entscheidung gefällt, für Annotationen und Dokumente dieselbe Bewertungsform zu verwenden, da dies einfacher zu implementieren ist. Die Nutzer müssen dann auch nicht zwei Arten von Bewertungen kennenlernen und somit kann die Einarbeitungszeit bezüglich des Systems durch die Verwendung einer einheitlichen Bewertungsform geringer gehalten werden.

Wie die Bewertungen innerhalb einer Kategorie aussehen sollen, wird im Folgenden beschrieben. Eine Möglichkeit wäre eine Skala wie die in Deutschland übliche Notengebung - also 1 bis 6 - zu verwenden. Dies hätte den Vorteil, dass es einfach zu implementieren

und darzustellen wäre. Jedoch ist die Verwendung einer solchen Skala eine 'sehr subjektive' Bewertungsmethode. Es besteht die Gefahr, dass jeder Nutzer anders bewerten würde und dadurch die Bewertungen nicht mehr vergleichbar wären. Beispielsweise gibt ein Nutzer eine 3, ein anderer eine 4, obwohl sie dasselbe meinen. Um dieses Problem der Skalenverschiebung zu umgehen, wurde beschlossen, 'beschriebene' Werte in den Kategorien zu verwenden und diese systemintern auf Zahlenwerte abzubilden. Durch diese Methode wird das Skalenverschiebungsproblem zumindest teilweise behoben. Die komplette Bewertungsskala sieht dann wie folgt aus.

Relevanz - mit den fünf Werten:

- sehr relevant (1)
- relevant (2)
- teilweise relevant (3)
- nicht relevant (4)
- völlig fehl am Platz (5)

Korrektheit - mit den fünf Werten:

- korrekt (1)
- wenige nicht bedeutsame Fehler (2)
- einige nicht bedeutsame Fehler / wenige gewichtige Fehler (3)
- einige gewichtige Fehler (4)
- falsch (5)

Hilfreich - mit den fünf Werten:

- sehr hilfreich (1)
- hilfreich (2)
- teilweise hilfreich (3)
- nicht hilfreich (4)
- störend (5)

Mit der Kategorie 'Relevanz' soll dabei bestimmt werden, ob Aussagen in einer Annotation zum annotierten Dokument bzw. der annotierten Dokumentenstelle und dem dort behandelten Thema passen. Ist die Bewertung auf ein Dokument bezogen, soll hier eine Aussage darüber gemacht werden, ob das Dokument das Themengebiet der Veranstaltung behandelt. Ein Beitrag, also ein Dokument oder eine Annotation, welche im inhaltlichen Zusammenhang auf keinen Fall fehlen darf, fällt unter 'sehr relevant'. Ein inhaltlich passender, aber nicht so existenziell wichtiger Beitrag hingegen soll mit 'relevant' bezeichnet werden. Mit 'teilweise relevant' werden Beiträge bezeichnet, die passende Inhalte aufweisen, aber auch andere Themen enthalten. Bezieht sich ein Beitrag zwar auf den inhaltlichen

Zusammenhang, schweift aber komplett vom Thema ab, so wird dieser mit 'nicht relevant' bezeichnet. Beiträge, die fachlich überhaupt nichts mit dem Inhalt der Veranstaltung zu tun haben, werden damit mit 'völlig fehl am Platz' bewertet.

Die Kategorie 'Korrektheit' nimmt Stellung dazu, ob die Aussagen eines Beitrages nach Meinung der bewertenden Person richtig oder falsch sind, bzw. wie viele Fehler vorhanden sind. Eine Annotation oder ein Dokument, in dem keine Fehler gefunden wurden, wird dementsprechend als 'korrekt' bezeichnet. Ist eine geringe Anzahl von Fehlern vorhanden, die aber die inhaltliche Aussage nicht verfälschen, wird der Beitrag mit 'wenige nicht bedeutsame Fehler' eingestuft. Dies könnte beispielsweise der Fall sein, wenn eine Rechnung einen Rechenfehler enthält, aber der Rechenweg trotzdem korrekt ist. Ein Rechenbeispiel, in welchem ein Rechenfehler gemacht wurde, der dazu führte, dass ein falscher Rechenweg eingeschlagen wurde, wird jedoch mit 'einige nicht bedeutsame Fehler / wenige gewichtige Fehler' bewertet werden. So auch, wenn eine Rechnung mit sehr vielen Rechenfehlern gespickt ist und deshalb kaum mehr nachvollzogen werden kann. Wenn zum Beispiel in einer Schlussfolgerung zwar der Ansatz richtig ist, aber mehrfach falsche Schlüsse gezogen wurden, fällt dies in den Bereich 'einige gewichtige Fehler'. Ist außerdem auch noch der Ansatz fehlerhaft, würde der Beitrag unter 'falsch' fallen.

Wie hilfreich für das Verständnis des Stoffes ein Beitrag ist, soll in der dritten Kategorie festgehalten werden. Eine Annotation, ohne die der Inhalt eines Dokumentes nicht verständlich wäre, soll dabei mit 'sehr hilfreich' bezeichnet werden. Ein Beitrag hingegen, der beim Verstehen hilft, aber ohne welchen ebenfalls der Stoff verstanden würde, soll als 'hilfreich' eingestuft werden. Beispielsweise ein zweites Rechenbeispiel zu einem mathematischen Zusammenhang, das hilft, das Verständnis zu festigen. 'Teilweise hilfreich' ist ein Beitrag, aus welchem nur Auszüge hilfreich sind. Ist in einem Beitrag beispielsweise ein Sachverhalt, der ohnehin verstanden wurde, mit anderen Worten nochmals erklärt, so ist dieser 'nicht hilfreich'. 'Störend' sind Annotationen oder Dokumente, die überhaupt nicht in den Zusammenhang passen oder sogar destruktiv auf das Verständnis Einfluss nehmen. So können Ausführungen beispielsweise verwirrend wirken.

Die Bewertungen sollen als Empfehlungen (siehe 5.2) gespeichert werden. Zur Verwaltung und Verteilung der Empfehlungen werden die in 5.2 beschriebenen Funktionen von UniTEC verwendet. Intern in UniTEC sollen die Bewertungen die im Folgenden erläuterte Form aufweisen.

Sie enthalten die global eindeutige Identifizierungsnummer der Annotation bzw. des Dokumentes, auf welches sie sich beziehen. Eine Bewertung enthält weiter die Schlüssel-Wert-Paare 'Relevanz', 'Korrektheit' und 'Hilfreich'. Die Werte zu diesen Schlüsseln liegen zwischen eins und fünf.

Vor einer Veröffentlichung einer Annotation bzw. eines Dokumentes macht eine Bewertung dieser keinen Sinn. Daher wurde beschlossen, dass nur bei Dokumenten bzw. Annotationen, welche veröffentlicht wurden, ein Bewerten möglich sein soll. Aus diesem Grund ist es auch nicht problematisch, dass Dokumente bzw. Annotation erst bei ihrer Veröffentlichung eine Identifizierungsnummer (siehe 7.2) erhalten.

7.4 Themenkategorien

Wie schon in 5.2 geschildert, wird jeder Vertrauenswert in UniTEC auf ein Pseudonym in Verbindung mit einer Themenkategorie bezogen (siehe auch 7.5). Deshalb werden Bewertungen je einer Themenkategorie zugeordnet. Wird eine Vertrauensänderung innerhalb einer Kategorie für ein Pseudonym vorgenommen, ändern sich in verwandten Kategorien die Vertrauenswerte bezüglich dieses Pseudonyms ebenfalls. Welche Kategorien verwandt sind und wie hoch der Verwandtschaftsgrad ist, also wie stark die Vertrauensänderung sich auf den Vertrauenswert in der verwandten Kategorie auswirkt, wird in UniTEC mittels eines Graphen definiert.

Dieser wird bei UniTEC für jeden Nutzer ganz persönlich festgelegt, wobei nicht per se gewährleistet wird, dass Themenkategorien, auf welche sich eine Bewertung bezieht, beim Nutzer vorhanden und zugehörige Abhängigkeiten definiert sind.

Um das zu gewährleisten, soll in diesem System ein globaler Themenkategoriengraph definiert werden(siehe FA25 - 'Themenkategorien'). Das soll geschehen, indem bei der Erstellung bzw. Veröffentlichung einer Veranstaltung diese einer bestehenden Themenkategorie zugeordnet wird. Ist keine passende vorhanden, wird eine neue Themenkategorie vom Veröffentlicher der Veranstaltung erzeugt, und die Abhängigkeiten zu bestehenden Kategorien definiert (dies geschieht also von einer begrenzten Nutzerzahl - den Dozenten). Jedes Dokument und jede Annotation, welche sich auf die Veranstaltung bezieht, gehört daraufhin automatisch dieser Kategorie an. Das Zuordnen von Veranstaltungen zu einer Themenkategorie ist für die Dozenten nur mit sehr wenig Aufwand verbunden. Das Erstellen einer Themenkategorie ist schon aufwändiger, aber immer noch in einem Bereich der Zumutbarkeit, wie durch das folgende Beispiel ersichtlich wird.

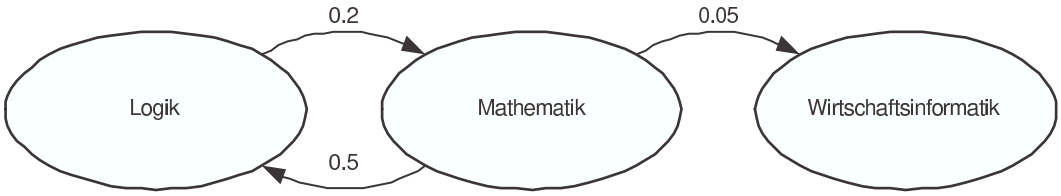

Abbildung 7.1: Beispiel Graph

Angenommen, es bestünde der in Abbildung 7.1 dargestellte Graph bereits im System. Auf der linken Seite in der Abbildung ist die Kategorie 'Logik' zu sehen. Eine Vertrauensänderung in dieser Kategorie nimmt mit 20 % Einfluss auf die rechts dargestellte Kategorie 'Mathematik'. In anderer Richtung ist eine 50-prozentige Beeinflussung definiert, wie aus der Abbildung ersichtlich ist. Die ganz rechts visualisierte Kategorie 'Wirtschaftsinformatik' wird von der Kategorie 'Mathematik' nur sehr gering beeinflusst. Eine ungekehrte Beeinflussung ist nicht gegeben.

Nun möchte ein Dozent eine Veranstaltung zum Thema 'Theoretische Informatik' abhalten. Deshalb fügt er, wie in Abbildung 7.2 zu sehen, die Kategorie 'Theoretische Informatik' ein und definiert die neuen Abhängigkeiten. Wie zu sehen, geht der Dozent davon aus, dass jemand, der in theoretischer Informatik über Wissen verfügt, sich ebenfalls in Logik

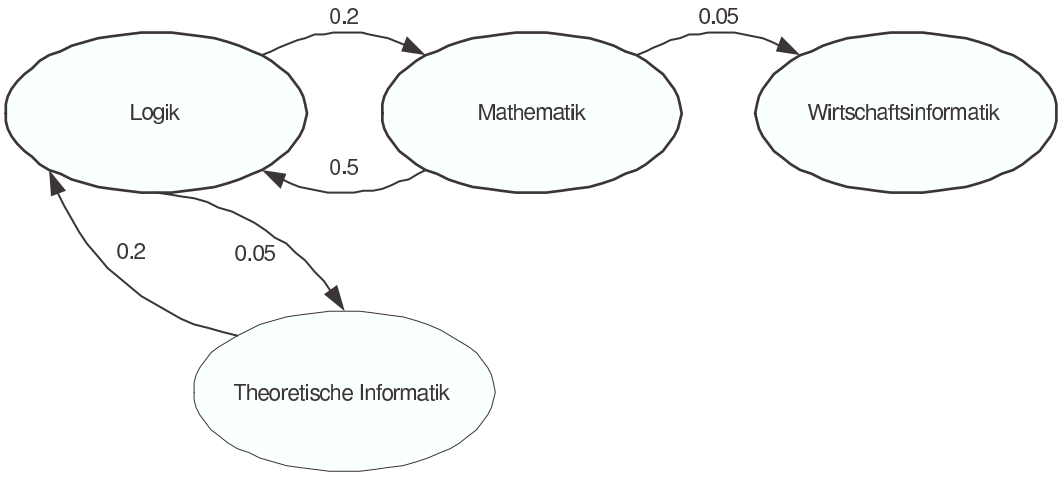

Abbildung 7.2: Beispiel Graph mit hinzugefügter Kategorie

auskennt. Dies gilt im gleichen Maße für Mathematik. Andersherum ist die Beeinflussung jedoch wesentlich geringer, da in der theoretischen Informatik einige Themen hinzukommen. Von einem direkten Bezug der Kategorien 'Wirtschaftsinformatik' und 'Theoretische Informatik' geht der Dozent nicht aus. Deshalb definiert er, wie in Abbildung 7.2 zu sehen ist, hierfür keine Abhängigkeiten.

Der globale Themenkategoriengraph wird lokal benötigt, damit Bewertungen auch offline vorgenommen werden können. Deshalb wird dieser vom Server geladen und lokal gehalten. Danach wird, wenn man mit dem Server Verbindung aufnimmt, geprüft, ob der globale Graph neue Kategorien aufweist. Ist dies der Fall, werden diese mit den dazugehörigen Abhängigkeiten in die lokale Version übernommen. Dies ist nötig, weil der Nutzer die Möglichkeit haben soll, den globalen Graphen an seine persönlichen Vorstellungen hinsichtlich der Verknüpfungsstärken der Themenkategorien anzupassen. Werden Änderungen an den Verknüpfungsstärken vorgenommen, so gelten diese nur lokal, persönlich für den Nutzer, der diese vornimmt.

7.5 Vertrauen

Zur Speicherung, Verwaltung und Abbildung von Vertrauenswerten stellt UniTEC, wie in 5.2 beschrieben, Algorithmen und Funktionalitäten zur Verfügung. Diese sollen herangezogen werden.

Die Funktion von UniTEC zum Bewerten von Empfehlungen ist eigentlich gedacht, um vom Benutzer selbst 'per Hand' ausgeführt zu werden. In diesem System soll dies jedoch automatisiert ablaufen. Deshalb muss die Funktionalität von UniTEC auf die im Folgenden beschriebene Weise erweitert werden.

Bei der gefilterten Synchronisation (siehe 7.7) wird je Annotation eine Anfrage an UniTEC gemacht. UniTEC liefert daraufhin alle innerhalb einer vom Nutzer konfigurierbaren Zeit auffindbaren Empfehlungen zurück.

Mit Hilfe der zu erstellenden Erweiterung soll dann eine Teilmenge aus den von UniTEC

gelieferten Empfehlungen gewonnen werden. Vom Nutzer soll hierfür eine Anfangsgröße des Vertrauenswertes definierbar sein. Aus dieser Teilmenge wird durch die Erweiterung eine vom Benutzer einstellbare Anzahl Empfehlungen herausgenommen, welche im Vergleich der Vertrauenswerte untereinander am besten abgeschnitten haben. Beim Filtern werden also nur die Urteile derer betrachtet, denen man am meisten vertraut.

Von diesen Empfehlungen wird in jeder Bewertungskategorie und über die Vertrauenswerte je ein Kumulationswert gebildet. Anhand dieses Wertes wird entschieden, ob die Annotation heruntergeladen werden soll (siehe 7.7). Ist dies der Fall, werden die kumulierten Werte gespeichert, damit mittels dieser beim Anzeigen die Filterung verschärft werden kann (ebenfalls in 7.7 näher beschrieben).

Wird vom Nutzer nun eine heruntergeladene Annotation bewertet, wird die Bewertung mit Hilfe der Routinen von UniTEC in der UniTEC-XML-Datenbank gespeichert. Außerdem muss zusätzlich durch eine Erweiterung gespeichert werden, dass diese Bewertung neu ist. Dies ist nötig, damit Bewertungen auch offline gemacht werden können.

Ist der Nutzer online, wird die Liste der neuen Empfehlungen abgearbeitet. Dies geschieht, indem ein im Hintergrund ablaufender Prozess zu der Annotation bzw. zu dem Dokument, auf welche sich die jeweilige Empfehlung bezieht, die Anfrageroutine von UniTEC aufgerufen wird. Dadurch erhält der Prozess die Empfehlungen anderer, welche sich auf die entsprechende Annotation bzw. das entsprechende Dokument beziehen.

Diese nach der Anfrage erhaltenen Empfehlungen werden dann einzeln mit der eigenen Bewertung verglichen. Wie schon in 7.3 angesprochen, sollen hierfür den einzelnen 'beschriebenen' Bewertungsskalen intern Skalenwerte von eins bis fünf zugeordnet werden. Stimmt eine Bewertung mit der eigenen – abgesehen einer 'Gesamtabweichung' von drei – überein, wird durch den Aufruf der Vertrauensupdate-Funktion von UniTEC (siehe 5.2) eine positive Vertrauensänderung vorgenommen. Stimmen die Bewertungen nicht auf die geschilderte Weise überein, wird eine negative Vertrauensänderung veranlasst.

Zur Bestimmung der 'Gesamtabweichung' werden in den drei Kategorien die Skalenwerte-Differenzen der zu vergleichenden Bewertungen gebildet. Diese werden addiert und ergeben die Gesamtabweichung. Dieser Mechanismus soll im Folgenden noch anhand eines Beispieles gezeigt werden.

Eigene Bewertung:
relevant (2)
wenige nicht bedeutsame Fehler (2)
teilweise hilfreich (3)

1. fremde Bewertung:
sehr relevant (1); Differenz = 1
wenige nicht bedeutsame Fehler (2); Differenz = 0
sehr hilfreich (1); Differenz = 2
Gesamtabweichung = 3; positive Vertrauensänderung

2. fremde Bewertung:
teilweise relevant (3); Differenz = 1
einige gewichtige Fehler (4); Differenz = 2
hilfreich (2); Differenz = 1
Gesamtabweichung = 4; negative Vertrauensänderung

Durch die erneute Anfrage an UniTEC zum Vertrauensupdate müssen nicht alle Empfehlungen bzw. Bewertungen lokal gespeichert werden. Außerdem wird das Vertrauensupdate dadurch immer mit aktuellen Empfehlungen durchgeführt. Durch die Neuanfrage ist gewährleistet, dass auch 'unbekannte' Nutzer zu einem Vertrauenswert gelangen. Das liegt daran, dass bei der beschriebenen Methode unabhängig vom bisherigen Vertrauenswert die Vertrauensänderung angestoßen wird.

Die für die 'Feinfilterung' beim Anzeigen gespeicherten kumulierten Werte können und sollten nach einer einstellbaren Zeit aktualisiert werden, wenn in der Zwischenzeit keine Abfrage erfolgt ist.

7.6 Synchronisation

Wie schon mehrfach geschildert, soll ein Offline-Arbeiten mit dem System möglich sein. Deshalb müssen die dafür nötigen Daten lokal gespeichert werden. Dies gilt auch für die Annotationen anderer Nutzer des Systems, die zugänglich sein sollen, wenn die eigenen Annotationen nicht ausreichen. Aus diesem Grund wird eine Synchronisation mit dem Server benötigt, wie in 3.8 gefordert.

Wann die Synchronisation mit dem Server angestoßen und durchgeführt wird, soll im Folgenden geklärt werden.

Wenn das System gestartet wird und eine Verbindung zum Server hergestellt werden kann, erwartet der Nutzer, dass dann eine Synchronisation mit diesem stattfindet und er erfährt, ob es neue Dokumente oder Veranstaltungen gibt. Im Zuge dessen ist es sicherlich auch sinnvoll, Dokumente, die wegen eines Dokumentenabonnements heruntergeladen werden sollen, herunterzuladen. Dokumente, die veröffentlicht wurden, sollten dann ebenfalls auf den Server geladen werden. Wird ein Dokument veröffentlicht während der Nutzer online ist, soll dieses natürlich sofort transferiert werden.

Nur wenn ein Dokument mit den Funktionalitäten von Multivalent im System angezeigt wurde, kann sich etwas an den Annotationen zu diesem Dokument geändert haben. Deshalb ist es sinnvoll, immer, wenn ein Dokument in einem Multivalent-Browser-Fenster geöffnet war, zu kontrollieren, ob etwas hochgeladen werden muss und wenn nötig, dies durchzuführen. Ist der Nutzer offline, wird gespeichert, dass das betreffende Dokument geöffnet war und sobald wieder eine Verbindung zum Server besteht, wird der Transferprozess angestoßen.

Der Benutzer wird nicht von allen Dateien, die auf dem Server zur Verfügung stehen, Annotationen herunterladen wollen. Daher werden, wie in den Anforderungen aufgeführt, nur Annotationen zu Dateien heruntergeladen, die vom Benutzer abonniert sind. Direkt

nach dem Einrichten eines Annotationsabonnements muss das Herunterladen der Annotationen für dieses soeben abonnierte Dokument in Gang gesetzt werden. Aber auch, wenn das gesamte System gestartet wird, muss, da sich seit dem letzten Arbeiten der Datenbestand auf dem Server stark geändert haben kann, das Herunterladen – und zwar bezogen auf alle abonnierten Dateien – durchgeführt werden. Außerdem ist es sicher sinnvoll, auf expliziten Wunsch des Benutzers den Annotationsdatenbestand zu aktualisieren.

Wie verdeutlicht, müssen beim Start des Systems sehr viele Daten transferiert werden. Aus diesem Grund und weil der Nutzer zumindest kein direktes Interesse an den Transferprozessen hat und lieber weiterarbeiten möchte, sollten diese Prozesse im Hintergrund ablaufen (siehe FA190 - 'Annotationssynchronisation' und FA50 - 'Veranstaltungs- / Dokumentensynchronisation'). Deshalb fiel hier die Wahl auf einen Deamon-Thread. Dieser wird beim Starten des Programms initialisiert und gestartet.

Wird eine Datei in einem Multivalent-Browser-Fenster (MBF) betrachtet, werden, sofern vorhanden, die zugehörigen Annotationen von Multivalent geladen. Dies geschieht, wie schon in 5.1 beschrieben, indem die entsprechende Hub-Datei geladen und intern als Baum gehalten wird. Schließt man das MBF oder lässt eine andere Datei in diesem MBF anzeigen, wird der Baum wieder gespeichert. Dies geschieht, indem die zu Anfang geladene Hub-Datei überschrieben wird, egal, ob Änderungen vorgenommen wurden oder nicht.

Um für die Synchronisation an die Annotationsinformationen heranzukommen bzw. um neue vom Server heruntergeladene Annotationen hinzufügen zu können, gibt es die im Folgenden aufgeführten Möglichkeiten.

1. während die 'annos.hub'-Datei als Baum gehalten wird, darauf zugreifen
2. solange nicht auf die Informationen in der entsprechenden Hub-Datei zugegriffen wird, sie also in keinem MBF geöffnet ist, diese bearbeiten/lesen
3. eine Kombination aus 1. und 2.

Für die erste Möglichkeit spricht, dass es möglich wird, an der Datei zu arbeiten während neue Annotationen dazugeladen werden. Auch können Annotationen dann direkt nach Veröffentlichung auf den Server geladen werden.

Jedoch muss für diese Realisierung dieser Variante sehr tief in Multivalent eingegriffen werden. Außerdem ergibt sich dann das Problem, dass davon ausgegangen werden muss, dass der Benutzer das MBF schließt oder sich ein anderes Dokument im MBF anzeigen lässt, wodurch die Hoch- bzw. Herunterladeaktionen abgebrochen oder der Baum bis zu ihrem Beenden weiter vorgehalten werden muss. Dies stellt sich innerhalb der Architektur von Multivalent als sehr schwierig und aufwändig dar.

Aus diesen Gründen ist die dritte Methode ebenfalls nicht sinnvoll. Deshalb wurde entschieden, die zweite Methode zu verwenden, obwohl dies natürlich auch Nachteile hat.

So muss die 'annos.hub'-Datei zum Hoch- bzw. Herunterladen von Annotationen zusätzlich geladen und geparst werden. Ein Nachteil ist auch, dass bei dieser Methode der Nutzer einen MBF schließen muss, wenn er überprüfen will, ob es neue Annotationen zu dem im MBF angezeigten Dokument gibt. Dies gilt auch, wenn der Nutzer neue oder

weitere Annotationen laden möchte. Eine parallele Visualisierung eines Dokumentes und die Synchronisation der Annotationen zu dieser Datei ist nicht möglich, da es sonst zu Lost-Update-Problemen führen würde.

Deshalb ist für die Realisierung der gewählten Methode ein Lock-Mechanismus für die Annotationen haltenden Hub-Dateien nötig. Der Lock-Mechanismus muss in einem extra Thread ablaufen, damit es nicht zu Deadlocks kommt. Da dieser Thread während der gesamten Betriebszeit benötigt wird, bietet sich wie schon für die Realisierung des Synchronisationsmechanismus ein Deamon-Thread an, der beim Start des Programmes initialisiert und gestartet wird.

7.7 Filterung

Warum die Filterung notwendig ist, soll im Folgenden anhand eines Beispiels erläutert werden.

Zu einer Veranstaltung, beispielsweise einer Vorlesung, gebe es ein begleitendes Skript mit 300 Seiten. Die Vorlesung sei von 200 Studierenden besucht. Weiter soll von dem relativ extremen Fall ausgegangen werden, dass im Schnitt drei Annotationen pro Seite des Skriptes von jedem Besucher veröffentlicht werden. So kommt man mit der Annahme, dass eine Annotation mit den zugehörigen Metadaten einen kByte benötigt, auf ca. 170 MByte nur für die Annotationen zu diesem einen Dokument der Veranstaltung.

Dies ist an sich schon eine zu große Datenmenge. Unter der Annahme, dass ein Nutzer nicht nur eine Veranstaltung besucht und im System 'bearbeitet', sondern mehrere, wäre dann allerdings mit astronomisch großen Datenmengen zu rechnen. Deshalb soll es möglich sein, mit Hilfe der schon beschriebenen Bewertungen (siehe 7.3) und der Vertrauensbildung (siehe 7.5) durch Filtern nur einen Teil der Annotationen herunterzuladen. Bei kleinen Veranstaltungen allerdings kann es Sinn machen, alle Annotationen zu laden und lokal zu speichern, deshalb soll dies Möglichkeit den Nutzern ebenfalls offen stehen. Eine dritte Möglichkeit, das temporäre Laden von Annotationen zur Laufzeit, welches für Nutzer gedacht war, welche 'immer' online sind, soll an dieser Stelle noch erwähnt werden. Es wurde jedoch verworfen, da es wegen der sehr hohen Anfragezeiten seitens UniTEC nicht sinnvoll durchführbar wäre.

Der Normalfall ist also das Laden mittels Filterung. Diese Methode soll deshalb im System die gewählte Voreinstellung sein. Wie die Filterung realisiert werden soll, wird im Folgenden erläutert.

Während des Synchronisationsprozesses (siehe 7.6) wird ermittelt, welche auf dem Server zur Verfügung stehenden Annotationen lokal noch nicht vorhanden sind. Bei der gefilterten Synchronisation soll nun vor dem eigentlichen Download der Annotationen ein weiterer Schritt durchgeführt werden.

Von den wie oben beschrieben ermittelten Annotationen wird dann, wie in 7.5 geschildert, mit Hilfe der erweiterten Funktionen von UniTEC eine Teilmenge bestimmt. Daraufhin werden nur diese zur Teilmenge gehörenden Annotationen heruntergeladen und nicht, wie ohne die Filterung, alle vorhandenen Annotationen.

Wie ebenfalls in 7.5 beschrieben, wird beim Herunterladen der Annotationen in jeder Bewertungskategorie und über die Vertrauenswerte jeweils ein Kumulationswert gebildet und mit der Annotation gespeichert. Dies ist notwendig, da neben der schon beschriebenen Filterung vor dem bzw. beim Herunterladen der Annotationen eine verschärfende, weiter eingrenzende Filterung beim Anzeigen möglich sein soll. Hierfür werden die kumulierten und gespeicherten Werte herangezogen und verglichen. Deshalb ist das Ermitteln von Kumulationswerten beim Herunterladen der Annotationen auch erforderlich, wenn alle Annotationen zu einem Dokument geladen werden.

7.8 Anonymität der Nutzer

Warum Nutzer gegenüber dem Betreiber bzw. dem System nicht anonym sein dürfen, ist im Abschnitt 'Haftung für Informationen im System' bereits erläutert worden. In diesem Abschnitt soll es nun um die Frage gehen, ob es möglich sein soll, dass Nutzer gegenüber anderen Nutzern im System anonym bleiben.

Unter 'anonym' wird verstanden, dass ein Nutzer unter einem oder auch mehreren Benutzernamen mit dem System arbeiten kann. Hierbei soll ein direkter Schluss mittels der Nutzernamen auf die tatsächliche Person nicht möglich sein.

Sicherlich kann man sehr viel über eine Person herausfinden, wenn man sich die Dokumente und Beiträge sowie die Bewertungen bzw. die Vertrauenswerte einer Person in den einzelnen Themenkategorien ansieht. Doch ist die Frage, ob dies im universitären Bereich eine so gewichtige Rolle spielt. Betrachtet man beispielsweise einen Informatikstudenten, so wird dieser im Bereich Betriebssysteme sicherlich höhere Bewertungen bekommen als ein Sprachenstudent - doch erfährt man dadurch nicht viel über die Person. Dies gilt zumindest solange, wie das System nur für die direkt mit den universitären Bildungsveranstaltungen und Aufgaben in deren Zusammenhang genutzt wird, und nicht etwa auch für die Bereiche aus der Freizeitgestaltung. Allerdings steht jedem Nutzer ja neben der generellen Nutzung auch frei, wieviel er und was er veröffentlicht. So symbolisiert das Nicht-Vorhandensein von Beiträgen oder Dokumenten zu einem Bereich von einem Nutzer nicht, dass er Defizite in diesem Bereich hat. Er kann in diesem Bereich das System ja auch nur nicht genutzt haben.

Gegen die 'anonyme' Nutzungsmöglichkeit des Systems spricht, dass die Hemmschwelle, das System zu missbrauchen, unsinnige oder unpassende Beiträge oder Dokumente zu veröffentlichen, wesentlich geringer ausfällt. Unterbindet man eine 'anonyme' Nutzung des Systems, würde sich das auf die Qualität der Beiträge positiv auswirken. Mit UniTEC steht in diesem System jedoch ein Mechanismus zur Verfügung, der unqualifizierte Veröffentlichungen relativ schnell aussortieren kann. Auch wird durch das Vertrauen den Nutzern eine Motivation an die Hand gegeben, nicht unüberlegt Veröffentlichungen zu tätigen, da dies eine Vertrauenswertsenkung nach sich ziehen würde. Deshalb ist die Möglichkeit, das System anonym verwenden zu können, sicherlich sinnvoll. Auch wird von manchen Studierenden die Erkennbarkeit als kritisch eingestuft, da so auch die Dozenten die Möglichkeit haben, beispielsweise einzelne als die ständig Fragenden oder die zu viel Kritik Übenden

zu identifizieren, und womöglich solche beispielsweise bei Prüfungen nachteilig behandeln könnten. Auch fürchtet manch einer wegen seiner Fragen belächelt zu werden oder auch, dass er vielleicht verleumdet werden könnte.

Deshalb wurde beschlossen, dass die gegenüber anderen Nutzern anonyme Verwendung des System möglich sein soll. Bezüglich dem Veröffentlichen von Dokumenten und Annotationen sollen die Nutzer einen Benutzernamen frei wählen können, wodurch zumindest gewährleistet ist, dass nicht eindeutig ist, wer hinter einem Benutzernamen steckt. Auf diese Weise kann allerdings nicht verhindert werden, dass verschiedene Veröffentlichungen den Benutzernamen zugeordnet und hierdurch Rückschlüsse auf den Benutzer gezogen werden können. Beim Erstellen von Bewertungen bleiben die Nutzer gegenüber den anderen Nutzern mittels der Funktionalitäten von UniTEC anonym. Deshalb soll wie ohnehin bei UniTEC vorgesehen, jeder Nutzer auch nicht nur ein Pseudonym in UniTEC verwenden können, sondern mehrere.

7.9 Annotationen

Für die Realisierung der Annotationen selbst sollen die in Multivalent vorhandene Annotationsfunktion 'kurzer Kommentar' (siehe 5.1.2] herangezogen und angepasst werden.

Die Annotationseigenschaften 'Position x', 'Position y', 'Seite' (bei nicht-Seiten-orientierten Dokumenten auf 0 gesetzt), sowie 'Höhe' und 'Breite' sollen wie in Multivalent realisiert übernommen werden. Um die Zugehörigkeit zur 'persönlichen Sicht' bzw. der 'nicht persönlichen Sicht', wie in FA110 - 'Sichten' und FA120 - 'persönliche Sicht' gefordert, bewerkstelligen zu können, ist eine weitere Annotationseigenschaft nötig: die 'Sicht'. Außerdem soll es möglich sein, Annotationen zu veröffentlichen oder auch als private Annotation zu behalten. Dies wird mittels der Eigenschaft 'veröffentlicht' erreicht, welche ebenfalls nicht in Multivalent realisiert ist und hinzugefügt werden muss.

Nach dem Veröffentlichen wird, wie schon beschrieben, eine Annotation auf den Server geladen. Hierbei erhält sie eine globale Identifizierungsnummer - 'ID' (siehe 7.2). Diese muss dann ebenfalls als Eigenschaft der Annotation gehalten werden. Die in Multivalent realisierte Zuordnungsmethode soll wie in 7.2 beschrieben für die lokale Zuordnung übernommen werden. Deshalb muss natürlich auch die Eigenschaft 'URL' der Annotationen, die hierfür benötigt wird, beibehalten werden.

Ein Ändern einer veröffentlichten Annotation muss verhindert werden, da ansonsten die eindeutige Zuordnung der Bewertungen zur Annotation wieder aufgehoben würde (Weiteres zum Ändern und Löschen von Annotationen in 9.2). Das soll mittels der Eigenschaft 'editierbar' geschehen, welche hinzugefügt wird. Eine eigene Annotation ist nicht mehr editierbar, wenn sie auf den Server geladen wurde. Fremde Annotationen sind nie editierbar.

Als weitere Annotationseigenschaften sind noch die 'Klassifizierung' und der 'Autor' zu nennen. Die 'Klassifizierung' enthält die Information, ob es sich bei der Annotation um eine Frage handelt. 'Autor' bezeichnet den Benutzername des Erstellers der Annotation.

Wie schon in 7.7 beschrieben, müssen außerdem die für die 'Feinfilterung' benötigten kumulierten Bewertungswerte 'Relevanz', 'Korrektheit' und 'Hilfreich' aufgenommen

werden. Auch ein Wert, der eine Kumulation der Vertrauenswerte enthält, ist erforderlich.

Selbstverständlich gehört zu einer Annotation auch der eigentliche Annotations-Text. Dieser soll mittels eines einfachen Fließtextes freier Länge, wie in FA80 - 'Annotationen erstellen' gefordert, realisiert werden. Hierfür können die Routinen von Multivalent herangezogen werden.

Zur lokalen Speicherung der Annotationen sollen die in Multivalent schon zur Verfügung stehenden Mechanismen zum Erzeugen, Speichern und Laden von 'annos.hub'-Dateien verwendet werden. Weitere Informationen zu 'anno.hub'-Dateien sind in 8.3.3 zu finden.

Kapitel 8

Realisierung

In diesem Kapitel soll darauf eingegangen werden, welche Teile des Entwurfes in der prototypischen Implementierung im Rahmen dieser Studienarbeit realisiert wurden.

Als erstes sollen einige allgemeine Realisierungsaspekte geschildert werden. Danach wird auf die Realisierung der Serverkomponenten eingegangen. Im Anschluss daran wird Stellung zu den realisierten Funktionalitäten der Clientseite genommen.

8.1 Allgemeine Realisierungsaspekte

Um das prototypische Realisieren möglichst vieler Komponenten zu ermöglichen, wurden einige Kompromisse eingegangen. So wurden an verschiedenen Stellen benötigte Konfigurationen 'hart' kodiert. Auch wurden die erstellten Frontends nicht optimiert. Die Bedienung ist an manchen Stellen ebenfalls auf einfachste Weise implementiert worden.

Multivalent verwendet eine Erweiterung von Java, die abstrakte Datentypen zur Verfügung stellt. Bedingt durch die Verwendung der Erweiterung musste zum Kompilieren Ant mit einem entsprechenden Build-Skript verwendet werden. Wo die 'Generics'-Erweiterung zu Java heruntergeladen werden kann, und wie diese installiert wird, kann [*SO] entnommen werden. Zum Kompilieren wurde das Apache Ant in der Version 1.5.4 verwendet. Apache Ant sowie weitere Informationen dazu sind unter [*ANb] erhältlich. Das verwendete Build-Skript wurde von [*SO] heruntergeladen und an die Anforderungen in dieser Arbeit angepasst.

So erzeugt das Build-Skript ein JAR-Archiv für den Server und ein JAR-Archiv für den Client. Die jeweils benötigten Komponenten von UniTEC werden dabei als kompilierte Bibliothek miteingebunden. Dadurch ist es möglich, das Gesamtprojekt als Free Ware zu publizieren. Der selbst erstellte Teil des Systems sowie die Klassen von Multivalent können dabei als Open Source mitgegeben werden.

Die Klassen von Multivalent sind in den Paketen 'multivalent', 'phelps' und 'tool' sowie 'tex' zu finden. An einzelnen Klassen im Paket Multivalent wurden Anpassungen vorgenommen. Auch wurde die Klasse 'MyNote' (siehe 8.3) in das Paket 'multivalent.std' eingefügt, da die Architektur und Organisation von Multivalent Behavior-Klassen dieser Art in diesem Paket vorsieht. Alle anderen selbst erstellten Klassen befinden sich im Paket 'rk'.

8.2 Realisierung des Servers

Zur Realisierung des Servers wurde, wie in Kapitel 6 beschrieben, ein Applikationsserver erstellt. Dieser ist im Paket 'rk.applicationServer' zu finden.

Als Datenbankserver wurde der MySQL Server Version 3.23.51-nt verwendet. Er ist neben näheren Informationen unter [*My] erhältlich. Für die Anbindung mit Hilfe von JDBC wurde der 'MySQL AB's JDBC Driver for MySQL – Connector/J 3.0.6 Stable' verwendet. Er ist unter [*Co] zu finden, wo auch weitere Informationen dazu erhältlich sind.

Für die prototypische Realisierung wurden auf dem MySQL-Server die Tabellen 'annos' und 'files' erzeugt. Die Tabelle 'files' enthält dabei die Spalten 'id', 'path' und 'publisher'. Die 'id' ist der global eindeutige Schlüssel. 'path' enthält den Pfad und den Dateinamen der eigentlichen Datei auf dem Server. 'publisher' hält den Benutzernamen des Nutzers, der dieses Dokument veröffentlicht hat. Die Tabelle 'annos' besteht aus den Spalten 'id', 'fileID', 'author', 'x', 'y', 'width', 'height', 'content' und 'page'. Dabei stellt 'id' wieder den Primärschlüssel dar. 'fileID' enthält den Fremdschlüssel der Datei, auf die sich die Annotation bezieht. 'author' ist der Benutzername des Nutzers, der die Annotation erstellt und veröffentlicht hat. Die Spalten 'x' und 'y' beinhalten die Position der Annotation auf dem Dokument, 'width' und 'height' die Größe des Annotationsfensters. In 'content' wird der eigentliche Annotationstext abgelegt. In der letzten Spalte 'page' wird die Seite, auf welche sich die Annotation bezieht, gespeichert. Bezieht sich die Annotation auf ein nicht seitenorientiertes Dokument, beinhaltet die Spalte eine 0.

```
Paket rk.applicationServer:

    Klasse        Server
    Klasse        DBConnection
    Interface     ApplicationServerInt
    Klasse        ApplicationServerImpl
    Klasse        ApplicationServerImpl_Skel
    Klasse        ApplikationServerImpl_Stub
```

Abbildung 8.1: Paket rk.applicationServer

Die Klasse 'Server' aus dem Paket 'rk.applicationServer' (siehe Abbildung 8.1) wird beim Aufruf der Server JAR-Datei instanziert und ausgeführt. Dabei wird eine Verbindung zur Datenbank aufgebaut. Hierfür werden verschiedene Kommunikationsroutinen benötigt. Diese wurden mit Hilfe der Klasse 'DBConnection' gekapselt. Steht die Verbindung zur Datenbank, wird der RMI-Server gestartet. Hierbei werden die eigentlichen Funktionalitäten des Applikationsservers zur Verfügung gestellt. Diese sind mit dem Interface ApplicationServerInt (siehe Anhang D) definiert und mit der Klasse ApplicationServerImpl realisiert. Die Skel- und die Stub-Klassen werden ebenfalls für die RMI-Kommunikation benötigt. Sie wurden mit RMI aus der ApplicationServerImpl Klasse generiert.

8.3 Realisierung des Clients

Zur Realisierung des Client wurden Multivalent und UniTEC herangezogen. Beim Aufruf der Client JAR-Datei wird die Hauptklasse von Multivalent 'multivalent.Multivalent' gestartet. Diese wurde um die Instanzierung und das Starten des Dokumenten-Explorers sowie der Deamons erweitert. Wie der Dokumenten-Explorer und die Deamons im Einzelnen realisiert wurden, soll im Folgenden geschildert werden.

8.3.1 Dokumenten-Explorer

Um den Dokumenten-Explorer zu realisieren, wurde das in Abbildung 8.2 dargestellte Paket 'rk.fileBrowser' erstellt.

```
Paket rk.fileBrowser:

        Klasse      FileBrowser
        Klasse      ESISTree
        Klasse      ESISNodeTreeModel
        Klasse      PairMap
```

Abbildung 8.2: Paket rk.fileBrowser

Die Klasse 'FileBrowser' realisiert die eigentlichen Funktionen des Dokumenten-Explorers. Sie bildet einen eigenständigen Thread und beinhaltet die Visualisierung des Dokumenten-Explorer-Fensters mit den Menüs und dem Veranstaltungs- bzw. Dokumenten-Baum.

Zur Realisierung des Baumes wurden die Klassen 'ESISTree' und 'ESISNodeTreeModel' erstellt. Sie basieren auf den Java eigenen Klassen 'JTree' und 'TreeModel'. Um das Speichern des Baumes einfach realisieren zu können, wurden als Knoten des Baumes Instanzen der Klasse 'multivalent.ESISNode' herangezogen. Diese Klasse stellt bereits die XML-Parser Funktionen zur Verfügung.

Die Veranstaltungen an sich wurden nicht realisiert. Anstelle dessen wurde eine Art 'Verzeichnis' erstellt. Dieses kann jedoch leicht um die für die Realisierung von Veranstaltungen notwendigen Eigenschaften und Funktionen erweitert werden.

Beim Start des Browsers werden die 'Baumdaten' aus einer XML-Datei geladen und als Baum angezeigt. Außerdem wurde das Überprüfen nach neuen Dokumenten realisiert, welches beim Start des Dokumenten-Explorers durchgeführt wird, wenn eine Verbindung zum Server aufgebaut werden kann. Da die Veranstaltungen nicht realisiert wurden, war die Realisierung des Veranstaltungsabonnements ebenfalls nicht möglich.

Das Dokumentenabonnement, das Hinzufügen von Dokumenten, das Veröffentlichen von Dokumenten, das Herunterladen von Dokumenten sowie eine Routine zum Erstellen von Verzeichnissen wurden realisiert. Auch können Dokumente ausgewählt und durch automatisches Öffnen eines MBF angezeigt werden.

Hierfür wurden die entsprechenden Menüeinträge und ein Action Listener in der Klasse 'FileBrowser' implementiert. Die eigentlichen Transfermechanismen, welche für das Veröffentlichen von Dokumenten sowie für das Herunterladen davon benötigt werden, sind mit dem FileTransferDeamon realisiert worden (siehe 8.3.2).

Wie in Abbildung 8.2 zu sehen, beinhaltet das Paket 'rk.fileBrowser' außerdem noch die Klasse 'PairMap'. Mit ihrer Hilfe wurde eine Liste realisiert, welche die Datenpaare (ID, Pfad) aller lokal verfügbaren Dokumente hält, die sich auch auf dem Server befinden. Diese wird benötigt, um zu überprüfen, ob Annotationen zu einem Dokument auf den Server übertragen werden dürfen. Dies soll nur dann geschehen, wenn das Dokument, auf das sich die Annotationen beziehen, ebenfalls auf dem Server verfügbar ist.

8.3.2 Dokument Transfer

Für den Transfer von Dokumenten, also das Herunterladen und Veröffentlichen von solchen, wurde ein Deamon implementiert. Dieser ist, wie in Abbildung 8.3 zu sehen, im Paket 'rk.deamon' zu finden.

```
Paket rk.deamon:

        Klasse      FileTransferDeamon
        Klasse      AnnoTransferDeamon
        Klasse      FileLockDeamon
        Klasse      TrustUpdateDeamon
```

Abbildung 8.3: Paket rk.deamon

Der Deamon enthält zwei Warteschlangen, eine für das Hochladen, eine für das Herunterladen von Dokumenten. Ist der Nutzer online, wird in regelmäßigen Abständen überprüft, ob sich in diesen Warteschlangen Einträge befinden. Ist dies der Fall, wird mit Hilfe der vom Server zur Verfügung gestellten RMI-Funktionen der Transfer durchgeführt.

Das Hochladen wird mit der Funktion 'putFile' (siehe Anhang D) durchgeführt.Nach dem Hochladen eines Dokumentes erhält der Deamon dabei die global eindeutige ID des Dokumentes zurück und trägt diese in den entsprechenden Knoten im Dokumenten-Explorer sowie in die 'PairList' (siehe 8.3.1) ein.

Mit der Funktion 'downloadFile' (siehe Anhang D) wird das Herunterladen bewerkstelligt.

8.3.3 Annotationen

Die Annotationen wurden mit Hilfe der Klasse 'multivalent.std.MyNote' realisiert. Hierfür wurde die in Multivalent enthaltene Klasse 'multivalent.std.Note' herangezogen und erweitert bzw. angepasst.

Wie vorgesehen wurden die 'persönliche Sicht' sowie die 'nicht-persönliche Sicht' implementiert. Das Unterscheiden von privaten und öffentlichen Annotationen ist ebenfalls

realisiert worden. Die Annotationen können also veröffentlicht werden. Hierfür wird eine Eintragung in die Warteschlange im AnnoTransferDeamon (siehe 8.3.4) vorgenommen. Auch können die eigenen Annotationen, solange sie nicht veröffentlicht wurden, gelöscht und editiert werden.

Zur lokalen Speicherung der Annotationen wurden wie geplant die Mechanismen von Multivalent verwenden. Wie der Aufbau der annos.hub-Dateien nun aussieht, ist im Folgenden anhand der Abbildungen 8.4 und 8.5 erläutert.

Schon bei Multivalent wurde zwischen 'seitenorientierten' und 'nicht-seitenorientierten' Dokumenten unterschieden. Dies wurde beibehalten. In Abbildung 8.4 ist eine annos.hub-Datei zu einem seitenorienten Dokument zu sehen. Die Abbildung 8.5 zeigt hingegen eine nicht-seitenorientierte annos.hub-Datei.

Das <personal>-tag ist das root tag. Es kommt immer nur einmal vor. Innerhalb des <personal>-tag können nun entweder 0 bis n <page> tags (also Seiten) mit wiederum 0 bis n <mynote> tags (also Annotationen) enthalten sein (siehe Abbildung 8.4), oder nur <mynote> tags, ebenfalls wieder 0 bis n.

Da Multivalent auch noch andere Annotationsarten wie zum Beispiel Links oder Markierungen unterstützt, können die entsprechenden korrespondierenden tags ebenfalls vorkommen.

Näheres zu den einzelnen Attributen des des <mynote>-tags können dem Programmcode der 'MyNote' Klasse entnommen werden.

```
<personal Behavior='Layer'
URI='file:/C:/Dokumente%20und%20Einstellungen/kuntzrd/
.Multivalent/workspace/8%201_Micro_einf_WS03_04_v2.pdf'>
  <page  page='1'>
    <mynote Behavior='MyNote' X='35' Y='23' Width='300'
    Height='100' menueoderanz='false' onserver='false'
    autor='ReinhardKuntz' perssicht='true' notshareable='false'>
      <content>
        hier steht der inhalt also der text einer Annotation
      </content>
    </mynote>
  </page>
  <page page='5'>
    <mynote Behavior='MyNote' X='38' Y='53' Width='500'
    Height='100' menueoderanz='false' onserver='false'
    autor='ReinhardKuntz' perssicht='true' notshareable='false'>
      <content>
        hier steht der inhalt also der text einer Annotation
      </content>
    </mynote>
    <mynote Behavior='MyNote' X='20' Y='35' Width='280'
    Height='120' menueoderanz='false' onserver='false'
    autor='ReinhardKuntz' perssicht='true' notshareable='false'>
      <content>
        hier steht der inhalt also der text einer Annotation
      </content>
    </mynote>
  </page>
</personal>
```

Abbildung 8.4: 'annos.hub'- Datei zu einem seitenorientierten Dokument

```
<personal Behavior='Layer'
URI='file:/C:/Dokumente%20und%20Einstellungen/kuntzrd/
.Multivalent/workspace/7%20Lüchle_Alm__2003_002%5B1%5D.JPG'>
  <mynote Behavior='MyNote' X='100' Y='100' Width='300'
  Height='100' menueoderanz='false' onserver='false'
  autor='ReinhardKuntz' perssicht='true' notshareable='false'>
    <content>
      Dies ist der inhalt der Annotation .. .. tolles Bild
    </content>
  </mynote>
  <mynote Behavior='MyNote' X='100' Y='100' Width='300'
  Height='100' menueoderanz='false' onserver='false'
  autor='ReinhardKuntz' perssicht='true' notshareable='false'>
    <content>
      Dies ist der Inhalt einer weiteren annotation .. nur
      leider ein wenig unscharf !
    </content>
  </mynote>
</personal>
```

Abbildung 8.5: 'annos.hub'-Datei zu einem nicht-seitenorientierten Dokument

8.3.4 Annotationstransfer

Wie im Entwurf geschildert, ist für den Annotationstransfer ein Deamon implementiert worden (siehe Abbildung 8.3). Dieser arbeitet wie der in 8.3.2 beschriebene 'FileTransfer-Deamon'. Jedoch werden vom 'AnnoTransferDeamon' vor dem Transfer der Annotationen noch die im Folgenden beschriebenen Operationen durchgeführt.

Bevor der Transfer der Annotationen zu einem Dokument durchgeführt wird, wird mit Hilfe des Lock-Deamon, 'FileLockDeamon' (siehe Abbildung 8.3) überprüft, ob die annos.hub-Datei von einem MBF verwendet wird. Ist das nicht der Fall, wird sie gesperrt. Daraufhin wird eine ID-Liste der zum entsprechenden Dokument auf dem Server verfügbaren Annotationen geladen. Aus dieser Liste werden alle ID's aussortiert, die zu Annotationen gehören, welche schon lokal vorhanden sind. Sollen alle Annotationen geladen werden, wird der Transferprozess in Gang gesetzt. Soll gefiltert geladen werden, wird zu den noch nicht vorhandenen Annotationen eine Anfrage an UniTEC gemacht. Entsprechend der Auswahlkriterien wird daraufhin die ID-Liste entsprechend gekürzt und die restlichen in der Liste befindlichen Annotationen werden geladen. Nach Ende des Transfers wird die neu erstellte Hub-Datei gespeichert und daraufhin die Sperre wieder freigegeben.

Annotationen, die neu veröffentlicht sind, werden vom Deamon daran erkannt, dass das 'privat' Attribut nicht gesetzt ist und noch keine ID vorhanden ist. In diesem Fall wird die Annotation auf den Server geladen.

Bei der Realisierung des Lock-Deamons wurde festgestellt, dass Multivalent, unabhän-

gig von der Anzahl der geöffneten MBFs, nur einen Thread verwendet. Damit kommt es in der ursprünglichen Multivalent-Version zu einem Lost-Update-Fehler, wenn in zwei verschiedenen MBFs ein und dasselbe Dokument geöffnet ist. Das Lost-Update-Problem ist durch den erstellten Sperr-Mechanismus behoben. Jedoch kommt es nun zu einer Verklemmung, wenn man dasselbe Dokument in zwei verschiedenen MBFs öffnet. Das liegt daran, dass in ein und demselben Thread (dem Multivalent-Thread) die Sperre auf die Hub-Datei gehalten und auf die Freigabe derer gewartet wird. Dieser Fehler wurde nicht behoben, da hierfür die Threadverwendung für MBFs hätte eingeführt werden müssen, wofür der Aufwand nicht absehbar war. Außerdem gehört dieses Problem nicht zu den Kernaufgaben dieser Arbeit.

8.3.5 Bewertungen und Vertrauensupdate

Um Bewertungen erstellen zu können, wurde die Klasse 'rk.Voting' implementiert. Sie enthält die Visualisierung des Bewertungs-Fensters. Außerdem wird mit ihrer Hilfe eine Bewertung, wie in [Aic03] beschrieben, erstellt und an den UniTEC-Agenten übergeben.

Allerdings wurde das Bewerten bisher nur für Annotationen, beim Hinzufügen oder Herausnehmen aus der 'persönlichen Sicht' implementiert. Das Bewerten von Dokumenten und ein freiwilliges Bewerten von Annotationen können aber mit wenig Aufwand unter Verwendung der 'rk.Voting' Klasse realisiert werden.

Die Anbindung von UniTEC ist neben der Schnittstelle in 'rk.Voting' komplett über die Klasse 'rk.UniTecUsage' realisiert. Sie wird im 'TrustUpdateDeamon' (siehe Abbildung 8.3) instanziert und gestartet. Daraufhin sorgen die Funktionen der 'UniTecUsage'-Klasse dafür, dass der UniTEC-Agent gestartet wird. Auch das persönliche Vertrauensmodell und der lokal gehaltene Themenkategoriengraph werden hierbei geladen. Die Synchronisationsmechanismen für den Graphen sind allerdings nicht realisiert worden. Stattdessen wurde eine 'standard'-Kategorie definiert, die somit bei jedem vorhanden ist.

Der Vertrauensvergleich und das Vertrauensupdate konnte nicht realisiert werden. Wie sich herausstellte, sind die hierfür benötigten Funktionen von UniTEC in der prototypischen Realisierung nicht implementiert. Die Anbindung und Erweiterung, wie im Entwurf dieser Arbeit vorgesehen, wurde aber dennoch implementiert. Hierfür wurde nur der Aufruf der UniTEC-Funktionen offen gelassen, so dass die endgültige Anbindung an UniTEC und die Realisierung des gefilterten Herunterladens einfach vervollständigt werden können.

Kapitel 9

Zusammenfassung und Ausblick

Abschließend soll in diesem Kapitel zusammengefasst werden, welche Ergebnisse in der vorliegenden Studienarbeit erzielt wurden. Anschließend werden einige Vorschläge zur Erweiterung der entwickelten Lösung gemacht.

9.1 Zusammenfassung

Ziel dieser Arbeit war es, Konzepte für den Weiterentwicklungsprozess von Lernmaterialien zu untersuchen. Dabei lag der Fokus auf dem Modellieren eines Konzeptes, das unter Verwendung des UniTEC-Systems die Problematik der Vertrauenswürdigkeit bzw. der Qualität der Informationen regelt. Insbesondere sollte dafür eine geeignete Methode zur eindeutigen Identifikation des Lernmaterials entworfen werden. Auch sollte ein Konzept erarbeitet werden, das es erlaubt, aus verschiedenen persönlichen Informationsmengen eine kollaborative Informationsmenge zu bilden.

Um das zu erreichen, wurde ein umfassendes Szenario erstellt. Aus weiteren Aspekten und dem Szenario wurden die Anforderungen an ein entsprechendes System abgeleitet. Bei der Untersuchung von in diesem Bereich bereits bestehenden Systemen wurde festgestellt, das diese den gestellten Anforderungen nicht oder nur in geringem Maße entsprechen. Deshalb wurde unter Verwendung der Softwaresysteme Multivalent und UniTEC ein neues System entworfen.

Dabei wurde ein Konzept entwickelt, das es ermöglicht, ohne Medienbruch Lernmaterial zusammenzustellen und an persönliche Anforderungen anzupassen. Außerdem wurde unter Zuhilfenahme von UniTEC und von Veröffentlichungsmechanismen eine Möglichkeit aufgetan, aus den entstehenden persönlichen Lernmaterialzusammenstellungen eine kollaborative Informationsmenge zu bilden, die zur Weiterentwicklung der Lernmaterialien verwendet werden kann.

Die hierbei entstehende Problematik, globale sowie lokale Annotationen den entsprechenden Dokumenten zuzuordnen, wurde gelöst. Auch ist es gelungen, das System so zu entwerfen, dass es sowohl online als auch offline zum Lernen verwendet werden kann.

Leider stellte sich beim Einarbeiten in das Multivalent-System heraus, dass es zwar

eine sehr gute allgemeine Dokumentation aufweist, der Programmcode selbst jedoch nicht oder nur sehr mangelhaft dokumentiert ist. Auch musste leider festgestellt werden, dass Multivalent noch relativ viele Fehler aufweist.

Mangels geeigneter Alternativen wurde Multivalent zur prototypischen Implementierung des Systems trotzdem herangezogen. Dies bedeutete jedoch, dass für das Einarbeiten, Anbinden, Erweitern und Anpassen von Multivalent sehr viel Zeit benötigt wurde (siehe dazu Anhang B). Das UniTEC-System bereitete ebenfalls Schwierigkeiten, da manche Schnittstellen in der prototypischen Realisierung nicht ganz sauber implementiert sind und es sich außerdem noch in der Weiterentwicklung befindet.

Obwohl frühzeitig auf diese Problematik reagiert wurde, konnten wegen des hohen Zeitbedarfs leider nicht so viele Funktionalitäten realisiert werden wie ursprünglich geplant. Die wichtigsten Kernfunktionalitäten des entworfenen Systems sind jedoch prototypisch implementiert worden, sodass deutlich wurde, dass das entworfene System realisierbar ist und die gestellten Anforderungen nicht utopisch sind.

9.2 Ausblick

Um das entworfene und prototypisch implementierte System in einen produktreifen Zustand zu bringen, müssen noch einige Arbeiten getätigt werden. So müssen verschiedene Funktionalitäten, die im Prototypen nicht realisiert wurden, noch implementiert werden. Beispielsweise das Benutzer- und Rechtemanagement und das Verknüpfen von Dokumenten untereinander. Auch muss an der grafischen Oberfläche der Clients noch gefeilt werden, um die intuitive Bedienbarkeit vollständig zu erreichen.

Außerdem kann weitergehend untersucht werden, ob es sinnvoll ist, eine so hohe Sicherheit bezüglich der Anonymität beim Bewerten bzw. beim Vertrauen bereit zu stellen wie im Prototypen geschehen. Eine Skalierung der Sicherheitsmechanismen in UniTEC ist möglich und verhält sich linear zur Arbeitsgeschwindigkeit des eingebundenen UniTEC-Systems. Ein für die Praxis geeigneter Kompromiss zwischen Anonymität und Geschwindigkeit sollte also noch erarbeitet werden.

Ein Test des Systems im realen Einsatz sowie mit größeren Nutzerzahlen konnte im Rahmen dieser Studienarbeit nicht gemacht werden. So bleibt also zu überprüfen, ob die Leistungsfähigkeit des Systems in der Praxis ausreicht.

Wie schon erwähnt, weist das integrierte Multivalent-System noch diverse Fehler auf und läuft teilweise instabil. Diese sollten natürlich ebenfalls angegangen werden.

Zur Erweiterung des entworfenen Systems gibt es diverse Möglichkeiten. So wäre es sinnvoll, nicht nur eine, sondern mehrere Austauschplattformen mit dem gleichen Client verwenden und verwalten zu können. Beispielsweise ist dies erforderlich, wenn ein Studierender Veranstaltungen an verschiedenen Fakultäten besucht, die jeweils einen eigenen Server betreiben.

Auch könnte zusätzlich für Annotationen eine Peer-to-Peer-Variante entwickelt werden. Dies hätte den Vorteil, dass Annotationen, die sich auf Internetseiten beziehen, dann unabhängig von Austauschplattformen annotiert werden könnten.

Die im eingebundenen Multivalent-System unterstützten Dokumenttypen können ebenfalls erweitert werden. Auch kann überlegt werden, ob nicht-Text-orientierte Annotationen wie das von Multivalent unterstützte 'markieren' sinnvoll kollaborativ genutzt werden können. Entsprechende Erweiterungen der Transfer-, sowie Synchronisationsmechanismen müssten dann entwickelt werden.

Im System ist das Löschen veröffentlichter Annotationen nicht möglich. Das Entwickeln von Konzepten und Methoden hierfür sollte in Betracht gezogen werden. Untersuchungen zu Versioning-Mechanismen für die Lernmaterialien im System sind ebenfalls ein wichtiger Erweiterungsbereich des Systems.

9.3 Fazit

Im Rahmen dieser Studienarbeit wurde ein System entwickelt, das es ermöglicht, Lernmaterialien kollaborativ weiterzuentwickeln. Dabei wurde für viele Probleme bisher bestehender Systeme Lösungen gefunden. Beispielsweise kann bei diesem System ohne Medienbruch gearbeitet werden. Dies gilt insbesondere bei der Erstellung von Mitschrieben auf Dokumenten. Lernmaterialien, die schon als digitale Dokumente bestehen, können in das System übernommen werden. Es sind keine aufwändigen, zusätzlichen Mechanismen im Vergleich zu traditionellen Vorgehensweisen sowohl bei der Weiterentwicklung von Lernmaterialien im Allgemeinen, wie auch beim Zusammenstellen einer persönlichen Informationsmenge notwendig.

Die prototypische Implementierung ist weit von einem produktreifen Zustand entfernt. Jedoch konnte mit ihrer Hilfe der Grundstein zur Implementierung des Systems gelegt werden.

Anhang A

Projektplan

Kollaborative Weiterentwicklung von Lernmaterial unter Verwendung eines Trust-Management-Systems

Autor: Reinhard Kuntz

A.1 Einleitung

Der Projektplan dient dem Bearbeiter und den Betreuern als Grundlage für die Projektüberwachung und -steuerung. Er legt den Projektablauf jedoch nicht ein für alle Mal fest, sondern soll im Laufe der Arbeit ergänzt und angepasst werden. Der Projektplan kann in Absprache mit den Betreuern in beidseitigem Einverständnis in den folgenden Fällen angepasst werden:

- bei Verzug
- bei vorzeitigem Abschluss eines Arbeitspaketes
- bei Gewinn neuer Erkenntnisse

Der aktualisierte Projektplan soll in den Anhang der Ausarbeitung einfließen.

Projektbeschreibung:

Bei diesem Projekt handelt es sich um eine Studienarbeit in der Abteilung Verteilte Systeme des Instituts für Parallele und Verteilte Systeme der Universität Stuttgart.

A.2 Beschreibung der Arbeitspakete

Dieser Abschnitt umfasst die Definition der wesentlichen Arbeitspakete, die Abhängigkeiten zwischen einzelnen Paketen und die Identifikation von kritischen Punkten, die die Durchführung des Projektes gefährden könnten.

A.3 Definition der Arbeitspakete

Die im Rahmen der Studienarbeit durchzuführenden Tätigkeiten lassen sich in folgende, zusammenhängende Arbeitspakete aufgliedern:

AP1 - Projektplan

Beschreibung: Erstmalige Erstellung des Projektplans

Status: begonnen am 15.10.03, abgeschlossen am 17.10.03

AP2 - Aufgabe konkretisieren

Beschreibung: Die Einleitung mit Motivation und Aufgabenstellung konkretisieren und erstellen

Status: begonnen am 17.10.03, abgeschlossen am 20.10.03

AP3 - Anforderungen u. Szenarien

Beschreibung: Erarbeiten und Erstellen von Szenarien anhand derer die Anforderungen herauskristallisiert und festgeschrieben werden. Unter anderem sollen dabei folgende Punkte erörtert werden:

- zu unterstützende Arbeitsprozesse
- Plattform
- online/offline Nutzbarkeit
- Anonymität der Nutzer
- Bewertungskriterien und Methoden
- Was sind Lernmaterialien
- Gruppen-/Personenrechte und Unterstützung von abgrenzbarer Gruppenarbeit
- Urheberrechte

Status: begonnen am 20.10.03, abgeschlossen am 26.10.03

AP4 - Verwandte Arbeiten

Beschreibung: Suchen und Bewerten von Arbeiten mit ähnlichen Ansätzen. Dabei soll insbesondere darauf eingegangen werden, ob diese Arbeiten den Anforderungen genügen, ob sie als Quellsystem (für Lernmaterial und Bewertungen) brauchbar sind und ob in den Arbeiten verwendbare Teile bzw. Ansätze vorhanden sind.

Status: begonnen am 26.10.03, abgeschlossen am 27.11.03

AP5 - Projektplan überarbeiten und verfeinern

Beschreibung: Projektplan überarbeiten und verfeinern, insbesondere der Arbeitspakete 6 und 7

Status:

AP6 - Entwurf

Beschreibung: Entwurf des Systems und der Schnittstelle zu UniTEC

Status:

AP7 - Implementierung

Beschreibung: Erstellen der Prototypischen Implementierung.

Status:

AP8 - Ausarbeitung

Beschreibung: Schriftliche Ausarbeitung der wesentlichen Ergebnisse.

Status:

AP9 - Vortrag

Beschreibung: Präsentation der Ergebnisse im Rahmen des VS-Kolloquium

Status:

A.3.1 Abhängigkeiten zwischen den Arbeitspaketen

Zwischen den Arbeitspaketen der Studienarbeit bestehen folgende Abhängigkeiten:

Das Arbeitspaket 1 ist gesondert, vor Beginn der anderen APs zu bearbeiten.
Die AP2 und 3 können parallel bearbeitet werden.
Das Arbeitspaket 4 sollte nach dem Beenden von AP2 und AP3 bearbeitet werden, da dann erst die Anforderungen vollständig bekannt sind.
Die Arbeitspakete 4 bis 7 müssen voraussichtlich sequentiell bearbeitet werden, wobei es zu Überlappungen der numerisch aufeinander folgenden APs kommen kann.
Das AP8 - 'Ausarbeitung' kann zum größten Teil parallel zu allen anderen Arbeitspaketen stattfinden.
Gesondert zu betrachten ist das Arbeitspaket 9, welches im Anschluss an die Arbeit stattfindet.

A.3.2 Kritische Punkte/Risiken

Ein kritischer Punkt ist mit Sicherheit bei AP6 - 'Entwurf' erreicht. Hier werden die Entscheidungen für den Rest des Projektes getroffen. Also, welche Komponenten und Konzepte für das System übernommen werden und welche neuen Konzepte und Komponenten entworfen und entwickelt werden.
Des weiteren ist als Risikobereich zu betrachten, dass nicht bekannt ist, ob es möglich

ist, innerhalb einer Java-Applikation pdf-Dateien darzustellen und zu bearbeiten, was zum 'markierenden Verlinken' von großem Vorteil wäre.

A.4 Zeitplan

In diesem Abschnitt erfolgt die zeitliche Planung der Arbeitspakete und Meilensteine. Der Bearbeitungszeitraum ist 6 Monate oder 26 Wochen - es bleiben 24 Wochen nach Abzug der Zeit um Weihnachten und Neujahr. Wenn sich der Zeitraum mehrerer Arbeitspakete überlappt, dann werden diese parallel bearbeitet. Am Ende des Bearbeitungszeitraumes ist ein Puffer eingeplant worden, der für Verzögerungen oder andere unvorhergesehene Tätigkeiten genutzt werden soll.

Aw	**Arbeitspakete und Meilensteine**	**Gepl. Zeitr.**	**Tats. Zeitr.**	**Verz.**
	AP1 - Projektplan	15.10.03	15.10.03	1 Tag
	Meilenstein 0	16.10.03	17.10.03	
1PW	AP2 - Aufgabe konkretisieren	17.10.03	17.10.03	3 Tage
	AP3 - Anforderugen u. Szenarien	23.10.03	26.10.03	
	Meilenstein 1			
4PW	AP4 - Verwandte Arbeiten	24.10.03	26.10.03	7 Tage
	Meilenstein 2	20.11.03	27.11.03	
1PW	AP5 - Projektplan überarbeiten und verfeinern	21.11.03		
	Meilenstein 3	27.11.03		
6PW	AP6 - Entwurf	28.11.03		
	Meilenstein 4	22.01.04		
5PW	AP7 - Implementierung	23.01.04		
	Meilenstein 5	26.02.04		
4PW	AP8 - Ausarbeitung	26.02.04		
	Meilenstein 6	24.03.04		
2PW	Korrekturlesen und Verbessern	25.03.04		
	Meilenstein 7	07.04.04		
1PW	Puffer	08.04.04		
		14.04.04		

A.5 Dokumente und Meilensteine

Als Ergebnis der Meilensteine sollen die im Folgenden aufgelisteten Dokumente entstehen. Die Dokumente unterliegen nach dem Erreichen des jeweiligen Meilensteins einer Versionskontrolle und dürfen dann nur noch unter Absprache mit den Betreuern geändert werden.

Meilenstein 0: Projektplan

Meilenstein 1: erste Einleitung mit Motivation und konkretisierter Aufgabenstellung sowie Szenarien und eine Liste der Anforderungen

Meilenstein 2: Verwandte Arbeiten

Meilenstein 3: überarbeiteter Projektplan

Meilenstein 4: Entwurf, Architekturplan/-beschreibung

Meilenstein 5: lauffähige Realisierung mit Dokumentation

Meilenstein 6: initiale schriftliche Ausarbeitung

Meilenstein 7: finale schriftliche Ausarbeitung

A.6 Weitere Anmerkungen

Es ist klar, dass die Arbeitspakete AP6 und AP7 zu groß sind und feiner untergliedert werden müssen. Jedoch ist zu diesem Zeitpunkt dies noch nicht möglich, da die nötigen Informationen fehlen. Die feinere Gliederung sollte aber spätestens nach Abschluss von AP4, also nach Abschluss des Meilensteins 2 möglich sein und dann auch gemacht werden. Deshalb wurde Meilenstein 3 bzw. AP5 - 'Projektplan überarbeiten und verfeinern' eingeplant.

Anhang B

Anmerkungen zum Projektplan

Leider stellte sich beim Einarbeiten in das Multivalent-System heraus, dass es zwar eine sehr gute allgemeine Dokumentation aufweist, der Programmcode selbst jedoch nicht, oder nur sehr mangelhaft dokumentiert ist. Da keine Alternative zur Verwendung von Multivalent vorhanden war, musste in Kauf genommen werden,dass zum Einarbeiten in das Multivalent-System sowie zur Einbindung und Erweiterung dessen sehr viel Zeit benötigt wurde.

Am Ende des Meilensteines 2 - 'Verwandte Arbeiten' war dies bereits deutlich erkennbar. Außerdem war klar, dass eine erfolgreiche Implementierung zumindest der wichtigsten Kernfunktionalitäten des Systems davon abhing, Multivalent entsprechend einzubinden und zu erweitern. Deshalb wurde entschieden, den in Anhang A abgebildeten Projektplan nur noch als zeitlichen Anhaltspunkt zu nehmen und auf inkrementelle Weise weiter vorzugehen.

Dies wurde umgesetzt, indem in kurzen Zyklen, zusammen mit den Betreuern erörtert wurde, wie weiter vorgegangen werden sollte.

Anhang C

Rechtliches

C.1 Markenzeichen

In dieser Arbeit wurden eingetragene Warenzeichen, Handelsnamen und Gebrauchsnamen verwendet. Auch wenn diese nicht als solche gekennzeichnet, sind gelten die entsprechenden Schutzbestimmungen.

C.2 Software Lizenzen

Die in der Prototypischen Realisierung des Systems verwendeten Software-Systeme unterliegen verschiedensten Lizenzbestimmungen. Diese können den angegebenen Dokumentationen der einzelnen Systeme entnommen werden. Außerdem sind die Lizenzen auch im Quellcode enthalten, sofern dies in den Lizenzbedingungen der Systeme gefordert wurde.

Anhang D

Interface - ApplicationServerInt

```
package rk.applicationServer;

import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.*;

public interface ApplicationServerInt extends Remote {

/**
 * Überträgt eine Annotation auf den Server (für
 * nicht-seiten-orientierte Dokumente). Liefert
 * die Annotations-ID zurück, wenn die Operation
 * auf dem Server erfolgreich ausgeführt wurde.
 */
public long putAnnotation(String uri,
    String author, String x, String y, String width,
    String height, String content)
    throws RemoteException;


/**
 * Überträgt eine Annotation auf den Server (für
 * seiten-orientierte Dokumente). Liefert
 * die Annotations-ID zurück, wenn die Operation
 * auf dem Server erfolgreich ausgeführt wurde.
 */
public long putAnnotation(String uri, String author,
    String x, String y, String width, String height,
    String content, String page)
```

```
    throws RemoteException;

/**
 * Überträgt eine Datei auf den Server und liefert
 * die global eindeutige Kennung der Datei zurück
 * (file-id), wenn die Operation Erfolg hatte.
 */
public long putFile(String fileName, byte[]
    fileContent, String Veroeffentlicher, String
    Beschreibung)
    throws RemoteException;

public ArrayList getFileList(ArrayList local)
    throws RemoteException;
public ArrayList getFileList()
    throws RemoteException;

/**
 * Gibt eine ArrayList of Strings mit den ID der
 * Annotationen zurück, welche zu der Datei mit
 * der ID - fileID - gehören.
 */
public ArrayList getAnnoIDListToFile
    (String fileID)
    throws RemoteException;

/**
 * Gibt eine ArrayList der Annotationen, die zum
 * Dokument fileID vorhanden sind und deren ID
 * nicht in annosLocal entalten sind zurück.
 */
public ArrayList getAnnos(ArrayList annosLocal,
    String fileID)
    throws RemoteException;

public byte[] downloadFile(String fileName)
    throws RemoteException;
}
```

Abbildungsverzeichnis

4.1 Tabellarische Zusammenfassung 'Verwandte Arbeiten - Lernplattformen' . . 37
4.2 Tabellarische Zusammenfassung 'Verwandte Arbeiten - Annotationssysteme' 38
4.3 Tabellarische Zusammenfassung 'Verwandte Arbeiten - Textverarbeitung / -visualisierung mit Annotationssystem' 38
4.4 Tabellarische Zusammenfassung 'Verwandte Arbeiten - Informationsbereitstellungsdienste' 38

5.1 Beispiel Baum – Buch 41

6.1 Funktionale Gliederung 49
6.2 Trennung von Applikationsserver und Datenbank 51
6.3 Server mit Bootstrap und mehreren Clients 54
6.4 Funktionale Gliederung und Kommunikation bei Verwendung von Multivalent und UniTEC 56

7.1 Beispiel Graph 63
7.2 Beispiel Graph mit hinzugefügter Kategorie 64

8.1 Paket rk.applicationServer 73
8.2 Paket rk.fileBrowser 74
8.3 Paket rk.deamon 75
8.4 'annos.hub'- Datei zu einem seitenorientierten Dokument 77
8.5 'annos.hub'-Datei zu einem nicht-seitenorientierten Dokument 78

Literaturverzeichnis

[*Ad] Adobe Acrobat 6.0 Professional. Website, Adobe Systems Incorporate. unter: `http://www.adobe.com/products/acrobatpro/` (11. März 2004).

[Aic03] Fabian Aichele. Realisierung und Evaluierung von Trust-Update-Algorithmen im Rahmen des UniTEC-Reputationssystems. Studienarbeit 1906, Universität Stuttgart, Fakultät Informatik, 2003.

[*Ana] Annotea Projekt. Website, Annotea Project, W3C. unter: `http://www.w3.org/2001/Annotea/`.

[*ANb] The APACHE ANT projekt. Website, Apache Software Foundation. unter: `http://ant.apache.org/` (08. April 2004).

[App03] W. Appelt. BSCL - Basic Support for Collaborative Learning. In *Tagungsband der 1. Deutschen E-Learning Fachtagung Informatik (DeLFI)*, München, Sep 16-18 2003. Gesellschaft für Informatik, Bonn.

[BBPW02] D. Bosau, C. Burger, M. Pfletschinger, and A. Wacker. *Context-Sensitive Interaction Support during Augmented Lectures.* Universität Stuttgart, Fakultät Informatik, 2002.

[BBW+03] D. Bosau, C. Burger, A. Wacker, S. Behling, and M. Schneider. Gemeinsame Anwendungsnutzung in der notebook-unterstützten Lehre - Technische und Didaktische Basis. Evaluation - Technischer Bericht 03/2003, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, 2003.

[BEH+02] F. Behringer, D. Engeldinger, F. Höschle, P. Jehlicka, F. Kaiser, T. König, F. Lösch, and M. Wetzel. Studienprojekt ECÜ. Abschlussbericht, Universität Stuttgart, Institut für Formale Methoden der Informatik, Abteilung Formale Konzepte, 2002.

[*BS] BSCW. Website, BSCW, Frauenhofer FIT, OrbiTeam. unter: `http://bscw.gmd.de/` (17. März 2004).

[Cer03] Holger Cermann. Fernsteuerung von Anwendungen. Studienarbeit 1872, Universität Stuttgart, Fakultät Informatik, 2003.

[*Co] Connector/J. Website, MySQL.com. unter: http://www.mysql.com/products/connector/j/ (08. April 2004).

[*ec] eclaus home. Website. unter: https://eclaus.informatik.uni-stuttgart.de:8543/eclaus/eclaus (19. April 2004).

[GJ03] Robert Godwin-Jones. Blogs and Wikis: Environments for online Collaboration. (Volume 7, Number 2):12–16, Mai 2003. unter: http://llt.msu.edu./vol7num2/emerging/default.html bzw. unter: http://llt.msu.edu./vol7num2/pdf/emerging.pdf sowie unter: http://llt.msu.edu/vol7num2/emerging/ (16. März 2004).

[HTZS03] M. Hupfer, E. Thor, A. Zobel, and U. Schubert. Kooperatives Authoring in einer virtuellen Umgebung. In *Tagungsband der 1. Deutschen E-Learning Fachtagung Informatik (DeLFI)*, München, Sep 16-18 2003. Gesellschaft für Informatik, Bonn.

[Huh02] Jochen Huhmann. Content Management für Eilige - Schnell, schnell. (Ausgabe 10/2002):S. 84 : Webwerkzeuge, 2002. unter: http://www.heise.de/ix/artikel/2002/10/084/ (11. März 2004).

[*IL] ILIAS OPEN SOURCE. Website, ILIAS Projekt, Universität zu Köln. unter: http://www.ilias.uni-koeln.de/ios/
(17. März 2004).

[*IT] Innovative Technology for Collaborative Learning and Knowledge Building (ITACOLE) project. Website, ITACOLE project, University of Art and Design Helsinki, Media Lab. unter: http://www.euro-cscl.org/site/itacole/public_deliverables_html
(11. März 2004).

[KP04] Michael Kinateder and Siani Pearson. A Privacy-Enhanced Peer-to-Peer Reputation System. Paper, University of Stuttgart (IPVS) and Hewlett-Packard Research Laboratories Bristol, 2004.

[*MU] Multivalent Projekt. Website, Multivalent Projekt, UC Berkeley's Digital Library Projekt, Digital Library Initiative. unter: http://www.cs.berkeley.edu/~phelps/Multivalent (16. März 2004).

[*My] Database Server. Website, MySQL.com. unter: http://www.mysql.com/products/mysql/ (08. April 2004).

[Nag03] Michael Nagy. Zwischenfragen in Augmented Lectures. Diplomarbeit 2039, Universität Stuttgart, Fakultät Informatik, 2003.

[PW] Thomas Phelps and Robert Wilensky. The Multivalent Browser - A Plattform for New Ideas. Website, University of California, Berke-

ley. unter: `http://www.cs.berkeley.edu/~phelps/Multivalent/Reserch/PlattformForNewI%deas.pdf` (11. März 2004).

[RK03] Kurt Rothermel and Michael Kinateder. Architecture and Algorithms for a Distributed Reputation System. In *Trust Management: First International Conference, iTrust*, Heraklion, Crete, Greece, May 28-30 2003. Nixon and Paddy.

[Sau03] Robert Sauter. Zustandsabhängige Verankerung von Zusatzinformation in Protokollsimulationen. Studienarbeit 1891, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, 2003.

[Sch02] Thilo Schnabel. Dynamisches Einbinden des Notizeditors in eine aus PowerPoint generierte Lernapplikation. Hiwi-arbeit, Universität Stuttgart, Institut für Automatisierungs- und Softwaretechnik, 2002.

[*SO] Projektseite zu Multivalent. Website, SOURCEFORGE.NET. unter: `http://sourceforge.net/projects/multivalent`
(17. März 2004).

[Sta] Gerry Stahl. Groupware Goes to School. Paper, Drexel University (Philadelphia, USA) und Frauenhofer-FIT (Bonn, Germany). unter: `http://bscl.fit.fraunhofer.de/download/criwg.pdf`
(16. März 2004).

[*SY] About Synergeia. Website, ITACOLE-Projekt, Frauenhofer FIT, OrbiTeam. unter `http://bscl.fit.fraunhofer.de/`
(17. März 2004).

[*Te] Testdaten für ToGather. Website, ToGather, Technische Universität Darmstadt. unter: `http://www.tk.informatik.tu-darmstadt.de/Lehre/ws0304/inf1/vo_recording%s` (10. März 2004).

[Ter03] Ralf Terdic. Konzeption und Realisierung eines Peer-to-Peer- Empfehlungsdienstes. Diplomarbeit 2097, Universität Stuttgart, Fakultät für Informatik, Elektrotechnik und Informationstechnik, 2003.

[*To] ToGather. Website, ToGather, Technische Universität Darmstadt. unter: `http://www.nu.tu-darmstadt.de/ToGather`
(10.März 2004).

[TRB+03] Christoph Trompler, Guido Rößling, Henning Cornelius Bär, Chin Choi, and Magnus Man. Kooperative Digitale Mitschriften auf mobilen Computern. In *Tagungsband der 1. Deutschen E-Learning Fachtagung Informatik (DeLFI)*, München, Sep 16-18 2003. Gesellschaft für Informatik, Bonn. unter: `http://www.tk.informatik.tu-darmstadt.de/Forschung/publikationen/pubs.p%hp3?lang=de&author=trompler`
(16. März 2004).

[*Wi] Wiki Wiki Web Internetseite. Website. unter: `http://www.c2.com/cgi/wiki?WikiWikiWeb` (17. März 2004).

[Wid03] Jochen Widmaier. Identitäts-Management in Peer-2-Peer-Umgebungen. Diplomarbeit 2094, Universität Stuttgart, Fakultät Informatik, 2003.

[Zha03] Yehua Zhang. Generierung von Lehrinhalten für die Online Nutzung mit erweiterter Funktionalität der Suche und Annotation. Bachelorarbeit, Technische Universität München, Institut für Informatik, 2003.

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfaßt und nur die
angegebenen Quellen benutzt zu haben.

_____

(Reinhard Kuntz)