

Master Thesis

Analyse, Design und Implementierung einer webbasierten Kollaborations- anwendung für die Entwicklung von semantischen Bezugssystemen

An der Fachhochschule Dortmund

im Fachbereich Informatik

Studiengang Medizinische Informatik

zur Erlangung des akademischen Grades

Master of Science in Medical Informatics

M. Sc. in Medical Informatics

vorgelegt durch

Robert Mützner, geboren am 14.06.1982

Dortmund, 17. Februar 2010

Betreuung:

Prof. Dr. Peter Haas

In Kooperation mit:

DIMDI

Dr. Sylvia Thun

Verfasser:

Robert Mützner, 7063037

Student im Master-Studiengang Medizinische Informatik
4. Fachsemester



Bearbeitungs-

zeitraum:

01.10.2009 – 28.02.2010

In Kooperation mit:

DIMDI

Deutsches Institut für

Medizinische Dokumentation und Information

Waisenhausgasse 36-38a

50676 Köln

Betreuung:

Prof. Dr. Peter Haas
FH Dortmund

(Erstbetreuer)

Dr. Sylvia Thun
DIMDI

(Zweitbetreuer)

Abstract

Diese Masterthesis behandelt Analyse, Design und Implementierung einer webbasierten Kollaborationsanwendung für die Entwicklung von Terminologien. Basierend auf einem vorhandenen Terminologieserver der Fachhochschule Dortmund wird der gesamte Entwicklungsprozess einer Plattform beschrieben, auf der Terminologien von Fachkräften entwickelt bzw. weiterentwickelt werden können. Im Rahmen der Arbeit wird anhand der Analyse und des Designs ein Konzept zur Erstellung der Software entwickelt. In der Analyse werden die Akteure, die Anwendungsfälle und das Verhalten erarbeitet. Anschließend wird daraus die Kollaborationsanwendung entworfen und implementiert.

Die abschließend dargestellten Testszenarien und Tests zeigen die sinnvolle Zusammenarbeit an einer Terminologie von mehreren Teilnehmern. Die daraus entstandenen Eindrücke und Kritikpunkte werden in der abschließenden Diskussion und im Ausblick erörtert.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Motivation.....	1
1.2	Ziele der Arbeit.....	2
2	Material, Methoden und Vorgehen.....	3
2.1	Ausgangssituation.....	3
2.1.1	Recherche	3
2.1.2	Terminologieserver der FH-Dortmund.....	5
2.1.3	Webbasierte Kollaborationsumgebung	6
2.2	Verwendete Methoden und Werkzeuge	7
2.2.1	Enterprise Architect	7
2.2.2	NetBeans IDE.....	8
2.2.3	Microsoft Excel	9
2.2.4	Java.....	10
2.2.5	Java-Applet.....	11
2.2.6	Log4j.....	11
2.2.7	MySQL.....	12
2.2.8	Web-Services (Web-Dienste)	13
2.2.9	Tomcat.....	14
2.3	Projektplanung und -verlauf.....	15
2.3.1	Zeitplan.....	16
3	Analyse	17
3.1	Akteure	17
3.1.1	Benutzer	18
3.1.2	Administrator	18
3.2	Anwendungsfälle (Use Cases)	20
3.2.1	Vorschlag	21
3.2.2	Diskussion	23
3.2.3	Benutzerverwaltung	24
3.2.4	Daten	25
3.3	Anwendungsfunktionen und Navigation	27
3.4	Verhalten/Aktivitäten	28
3.4.1	Status	29
3.4.2	Aktionen	30
3.4.3	Statuswechsel.....	31
3.5	Benutzerfunktionen / Pflichtenheft	32
3.5.1	Zielbestimmung.....	32

3.5.2	Einsatz	33
3.5.3	Umgebung.....	33
3.5.4	Funktionalität	34
3.5.5	Daten	35
3.5.6	Leistungen	35
3.5.7	Benutzeroberfläche	35
3.5.8	Qualitätsziele	35
4	Design.....	36
4.1	Klassenmodell Fachkonzept.....	36
4.1.1	Rollen- und Benutzerkonzept	37
4.1.2	Vorschlag-Konzept.....	39
4.1.3	Diskussion	43
4.1.4	Status-Verwaltung	45
4.1.5	Sicherheit.....	47
4.2	Datenbankmodell	48
4.3	Webservice	50
4.3.1	Dienste	51
4.3.2	Dienstmatrix.....	52
5	Implementierung	53
5.1	Datenhaltung	53
5.1.1	Ableitung des Datenbankentwurfs	53
5.1.2	Performance-Überlegungen	54
5.2	Webservice Kollaborationsumgebung.....	55
5.2.1	Projekt und Webservice erstellen.....	55
5.2.2	Typen erstellen	55
5.2.3	Dienste erstellen	56
5.2.4	Datenbankverbindung	56
5.2.5	Programmlogik	56
5.2.6	Sicherheit.....	57
5.3	Java-Applet.....	58
6	Test und Testergebnisse	59
6.1	Testumgebung.....	59
6.2	Probelauf	60
6.3	Testergebnisse	68
7	Diskussion und Ausblick.....	69
7.1	Diskussion	69
7.2	Ausblick.....	71
8	Anlagen.....	72

8.1	Installationshinweise	72
8.2	Datenbankmodell	72
8.3	Dienste-Matrix.....	72
9	Anhang	73
9.1	Quellenverzeichnis	73
9.2	Verwendete Hilfsmittel und Materialien	75
9.3	Verzeichnisse.....	76
9.3.1	Abbildungen.....	76
9.3.2	Tabellen	76
9.3.3	Modelle.....	76
9.3.4	Bildschirmabzüge.....	77
9.4	Abkürzungsverzeichnis	78
9.5	Glossar	79
9.6	Index	81

Danksagung

An erster Stelle möchte ich mich bei Herrn Prof. Dr. Peter Haas bedanken, für die Betreuung während des gesamten Studiums, für das Vertrauen, dass er in mich als wissenschaftlicher Mitarbeiter und zuvor als Hilfskraft gesetzt hat und schließlich für die Betreuung während der Masterarbeit. Für die Möglichkeit der technischen Umsetzung danke ich Herrn Witold Schiprowski, welcher mich bei der Publikation der Arbeit auf dem Server der Fachhochschule Dortmund unterstützt hat.

Ein ganz besonderer Dank gilt dem DIMDI und Frau Dr. Thun, welche sich als Zweitprüferin zur Verfügung gestellt hat und für die gute Kooperation zwischen der FH Dortmund und dem DIMDI gesorgt hat.

Zuletzt möchte ich mich bei meiner Familie für die Unterstützung während des gesamten Studiums und während der Anfertigung dieser Arbeit bedanken. Besonderer Dank gilt dabei meiner Mutter, Ursula Mützner, für ein ausgiebiges Korrekturlesen.

Dortmund, im Februar 2010

Robert Mützner

1 Einleitung

Die folgende Arbeit beschäftigt sich mit Implementierung einer Kollaborationsumgebung zur Entwicklung bzw. Weiterentwicklung von Terminologien in der Medizin durch eine webbasierte Zusammenarbeit von mehreren Personen und/oder Organisationen.

1.1 Motivation

Zum Austausch von Daten und Dokumenten zwischen Informationssystemen ist die Festlegung der Syntax und Semantik notwendig. Während die Syntax für Empfängersysteme die Möglichkeit schafft, die empfangenen Datensätze bzw. Informationsobjekte in ihre einzelnen Anteile zu zerlegen und so diese Teile in die Datenhaltung des Systems syntaktisch korrekt einzufügen, ermöglicht eine vereinbarte Semantik, diese Daten auch inhaltlich zu interpretieren und zu verarbeiten. Durch vereinbarte Semantik wird die „richtige“ Interpretation, Weiterverarbeitung und Speicherung der strukturell zerlegten Daten erst möglich.

Ein großes Problem besteht dabei perspektivisch in der Versionspflege und der Verteilung der vielen Vokabulare innerhalb der Telematikplattform, die manuell bei der o.a. Anzahl von Teilnehmersystemen kaum mehr zeitnah und adäquat ohne entsprechende Technologieunterstützung geleistet werden kann. Bei einer solchen Pflege sind auch die politischen und sachlichen Grenzen zu berücksichtigen, um eine hinreichende Stabilität der Vokabulare zu erreichen.

Erfahrungsgemäß ist vor allem die fachlich-inhaltliche Konsentierung unter den Beteiligten sehr arbeits- und zeitintensiv. Von größter Bedeutung ist daher der Einsatz einer mit oder in dem Terminologieserver integrierten Kollaborations- und Workflow-Software für die IT-gestützte Organisation der Terminologearbeit, sodass ein Terminologieserver nicht nur zur gezielten Verteilung bzw. Bereitstellung von innerhalb der Gesundheitstelematikplattform gültigen Semantikdefinitionen beiträgt, sondern auch zu deren Erarbeitung und nicht-redundanten Nutzung.

1.2 Ziele der Arbeit

Gegenstand der Masterthesis sind die Analyse, der Entwurf und die Implementierung der wesentlichen Funktionen einer Kollaborationsplattform für die nationale Terminologiearbeit im Gesundheitswesen mittels folgender Arbeiten:

- Analyse
 - ... der vorhandenen Workflows bzw. Organisationsprinzipien in den einschlägig national tätigen Organisationen bezüglich derer Terminologieentwicklung
 - ... der Anforderungen und Wünsche der potentiellen Endbenutzer (was, wie, wann) → Beschreibungen der Anwendungsfälle
 - ... Analyse ggf. vorhandener Lösungsansätze (Literatur, DIMDI, usw.)
 - ... Identifikation ggf. zu berücksichtigender Schnittstellen
- Design
 - Gesamtkonzept, Architektur
 - Anwendungsfälle
 - Klassenmodell und Objektlebenszyklen
 - Oberflächendesign
- Implementierung
 - ... einer Webanwendung in Form eines Java-Applets
 - ... einem Webservice für die Datenbankbindung
 - ... der notwendigen Datenbankstrukturen
 - ... der notwendigen Anwendungsfunktionen und Algorithmen
- Test und Testergebnisse
 - Test der Anwendung im Labor und mit ausgewählten Anwendern
- Inbetriebnahme für Evaluation
 - Inbetriebnahme des Lösungsansatzes mit einem kleinen Kreis von Anwendern über 2 Wochen sowie
 - Evaluation des Einsatzes (Fehler, Akzeptanz, Nutzeneffekte etc.)

2 Material, Methoden und Vorgehen

2.1 Ausgangssituation

Im Folgenden wird die Ausgangssituation beziehungsweise das vorliegende Material zu Beginn des Projekts beschrieben. Die Kollaborationsumgebung nutzt u. A. auch bereits verfügbare Dienste des Terminologieservers der FH Dortmund, welcher im Kapitel 2.1.1.2 näher beschrieben ist.

2.1.1 Recherche

2.1.1.1 Terminologien

Als Terminologie wird die Gesamtheit aller Begriffe und Benennungen einer Fachsprache bezeichnet. Es gibt mehrere Darstellungsformen von Terminologien:

- Wörterbuch
Wörterbücher können aus einer oder mehreren Spalten bestehen. Diese enthalten eine Auflistung von Begriffen.
- Wörterbuch mit enthaltenen Taxonomien
Diese Form besteht aus Einträgen, welche auf unterschiedlichen Ebenen liegen können. Im Allgemeinen ist ein solcher Aufbau mittels einer Baumstruktur gut darstellbar. Ein Begriff Computertomographie kann z.B. auf Ebene 1 sein, „Schädel-CT“ oder „Thorax -CT“ liegen entsprechend auf Ebene 2.
- Mehrachsige Begriffssysteme
Bei einer solchen Form setzen sich Begriffe aus mehreren Achsen zusammen. Wenn z.B. eine Achse Körperteile und eine andere Achse Krankheitstypen enthält, wird ein Beinbruch aus dem Begriff „Bein“ der Achse Körperteile und dem Begriff „Bruch“ der Achse Krankheitstypen zusammengesetzt.

Eine spezielle Form ist ein „kontrolliertes Vokabular“. Man spricht von dieser Form, wenn Einträge nur von einer zentralen Stelle festgelegt werden dürfen.

Terminologien sind in der Medizin sinnvoll, sie helfen bei der Abrechnung und können bspw. vor unerwünschten Arzneimittelwirkungen warnen oder der Qualitätssicherung dienen.

2.1.1.2 Terminologieserver

Ein Terminologieserver ist ein Anwendungssystem, welches Terminologien - meist in Form von kontrollierten Vokabularen, aber auch in Form von Ordnungssystemen wie Klassifikationen - für Benutzer bzw. Anwendungssysteme maschinenlesbar zur Verfügung stellt. Dadurch wird die semantische Interoperabilität in einem verteilten System unterstützt. Folgende Abbildung zeigt den Zusammenhang zwischen Systemen und dem Terminologieserver (Haas, 2006):

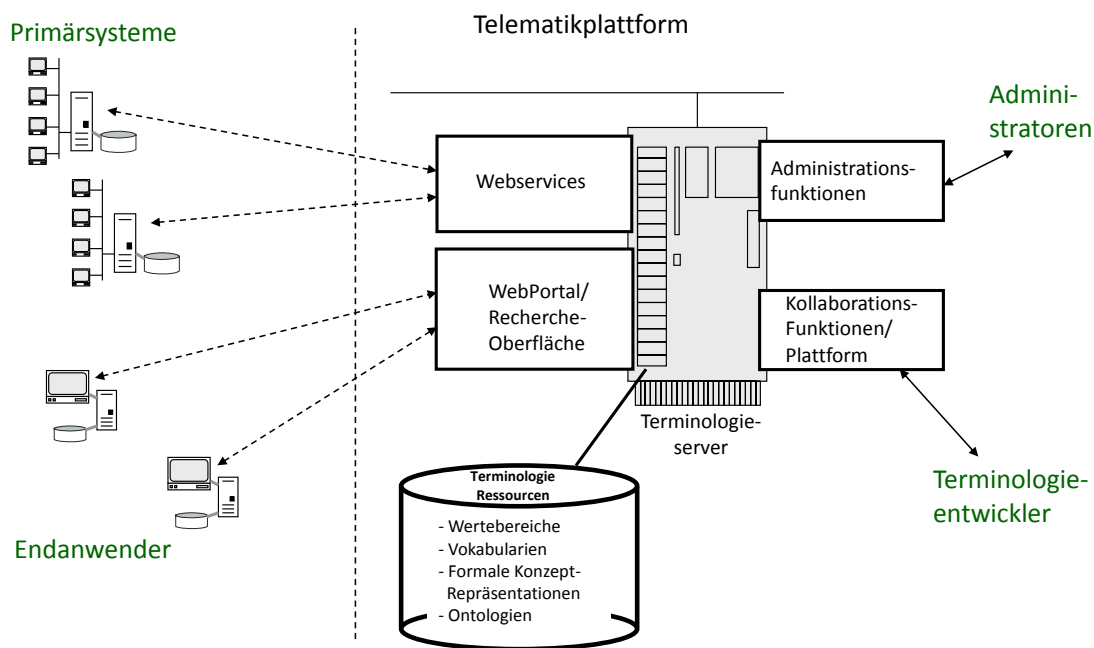


Abbildung 2-1 Terminologieserver

Sowohl Primärsysteme als auch Endanwender sollen mit verschiedenen Mitteln auf den Terminologieserver zugreifen können. Während Primärsysteme ihre lokalen Terminologien per Webservice synchronisieren, pflegen oder recherchieren Endanwender Terminologien per dialogbasierter Weboberfläche.

2.1.2 Terminologieserver der FH-Dortmund

Das Projekt „Terminologieserver“ der FH Dortmund bietet die Möglichkeit, Terminologien auf einem Server zu speichern und diese per Webservices abzurufen oder zu pflegen. Besondere Merkmale sind die Versionsverwaltung sowie die Möglichkeit des Cross-Mappings¹.

Der Terminologieserver hat vier Dienstklassen mit den jeweiligen Funktionen, welche an den Standard CTS II angelehnt sind (HL7 Common Terminology Services 2 Service Functional Model):

- Administration (Administration)
 - Vokabular importieren (Import Code System)
 - Vokabular exportieren (Export Code System)
- Suche/Zugriff (Search)
 - Vokabulare auflisten (List Code Systems)
 - Details zu Vokabularen erhalten (Return Code System Details)
 - Begriffe auflisten (List Code System Concepts)
 - Details zu Begriffen erhalten (Return Concept Details)
 - Verbindungstypen auflisten (List Concept Association Types)
 - Details zu Verbindungstypen erhalten (Return Concept Ass. Type Details)
 - ValueSets² auflisten (List ValueSets)
 - Details zu ValueSets erhalten (Return ValueSets Details)
 - Inhalte zu ValueSets auflisten (List ValueSet Contents)
- Pflege (Authoring)
 - Vokabular erstellen (Create Code System)
 - Vokabular bearbeiten (Maintain Code System Version)
 - Status eines Vokabulars ändern (Update Code System Version Status)
 - Begriff erstellen (Create Concept)
 - Begriff bearbeiten (Maintain Concept)
 - Status eines Begriffs ändern (Update Concept Status)
 - ValueSet erstellen (Create ValueSet)
 - ValueSet bearbeiten (Maintain ValueSet)
 - Status eines ValueSets ändern (Update ValueSet Status)

¹ Cross Mapping – Begriffe verschiedener Terminologien miteinander verbinden

² Value Set – manuelle Zusammenfassung von Begriffen

- Konzept-Verbindungen (Concept Association)
 - Verbindungen zwischen Begriffen auflisten (List Concept Associations)
 - Details zu Verbindungen erhalten (Return Concept Association Details)
 - Verbindungen erstellen (Create Concept Association)
 - Verbindungen bearbeiten (Maintain Concept Association)

Die oben genannten Dienste basieren auf dem HL7 CTS 2-Standard „Common Terminology Services 2“

([http://216.47.173.10/apelcore/index.php/HL7_Common_Terminology_Services_2_Service_Functional_Model_\(SFM\)](http://216.47.173.10/apelcore/index.php/HL7_Common_Terminology_Services_2_Service_Functional_Model_(SFM)) , letzter Zugriff am 13.01.2010).

Der Terminologieserver bietet seine Dienste über Webservices an, welche öffentlich erreichbar sind.

2.1.3 Webbasierte Kollaborationsumgebung

Die Entwicklung einer webbasierten Kollaborationsumgebung für den Terminologieserver ist keine normale Anwendung, welche vom Anwender-PC ausgeführt wird, sondern ist wie der Titel bereits verrät eine webbasierte Anwendung.

Was bedeutet webbasiert?

Es bedeutet, dass der Benutzer die Anwendung aus dem Web aufrufen kann, ohne dass dieser ein spezielles Programm auf seinem PC installieren muss. Dazu kommen mehrere Technologien, wie z.B. PHP, in Frage. Aufgrund der Komplexität vieler Bildschirmmasken wäre eine Umsetzung in PHP jedoch sehr aufwändig. Deshalb wird eine Umsetzung der Anwendung mittels eines Java-Applets realisiert. Damit ist es möglich, komplexe Bildschirmmasken zu erstellen. Ein Applet lädt sich beim Starten automatisch auf den PC und wird von dort aus ausgeführt. Der Zugriff auf die Daten, welche auf einem Server liegen, erfolgt ausschließlich per Webservices.

2.2 Verwendete Methoden und Werkzeuge

Verwendete Methoden sind Standards, Programmiersprachen oder Framework, verwendete Werkzeuge sind Entwicklungstools oder Datenbanken. Alle Methoden und Werkzeuge werden im Folgenden erläutert.

2.2.1 Enterprise Architect

Softwarearchitektur

Als Entwicklungswerkzeug wird der Enterprise Architect der Firma „Sparx Systems“ (<http://www.sparxsystems.com.au/>, letzter Zugriff am 09.02.2010) in der Version 7.0 eingesetzt. Damit ist es möglich, die grundlegenden Elemente sowie die Struktur des Softwaresystems zu beschreiben, also kurz: Softwarearchitektur. Dazu gehören die Akteure, die Anwendungsfälle, das Softwaremodell, das Datenbankmodell und die Beschreibung des Workflows.

Eigenschaften wie Modifizierbarkeit, Wartbarkeit oder Sicherheit spielen bei dem Entwurf eine große Rolle. Spätere Änderungen dieser Eigenschaften sind nur schwer möglich, deswegen muss gerade diese Phase gut durchdacht werden.

UML

Zur Verdeutlichung von Modellen, Prozessen oder Use-Cases kommt die UML zum Einsatz. Verwendet wird die aktuelle Notation UML 2. UML ist eine der führenden Modellierungssprachen. Mit ihrer Hilfe ist es möglich, Begriffe zu definieren sowie Beziehungen zwischen diesen festzulegen. Die Darstellung erfolgt dabei größtenteils grafisch. Die erzeugten Diagramme sind weitestgehend plattformunabhängig und auch anfangs noch nicht an eine Programmiersprache gebunden.

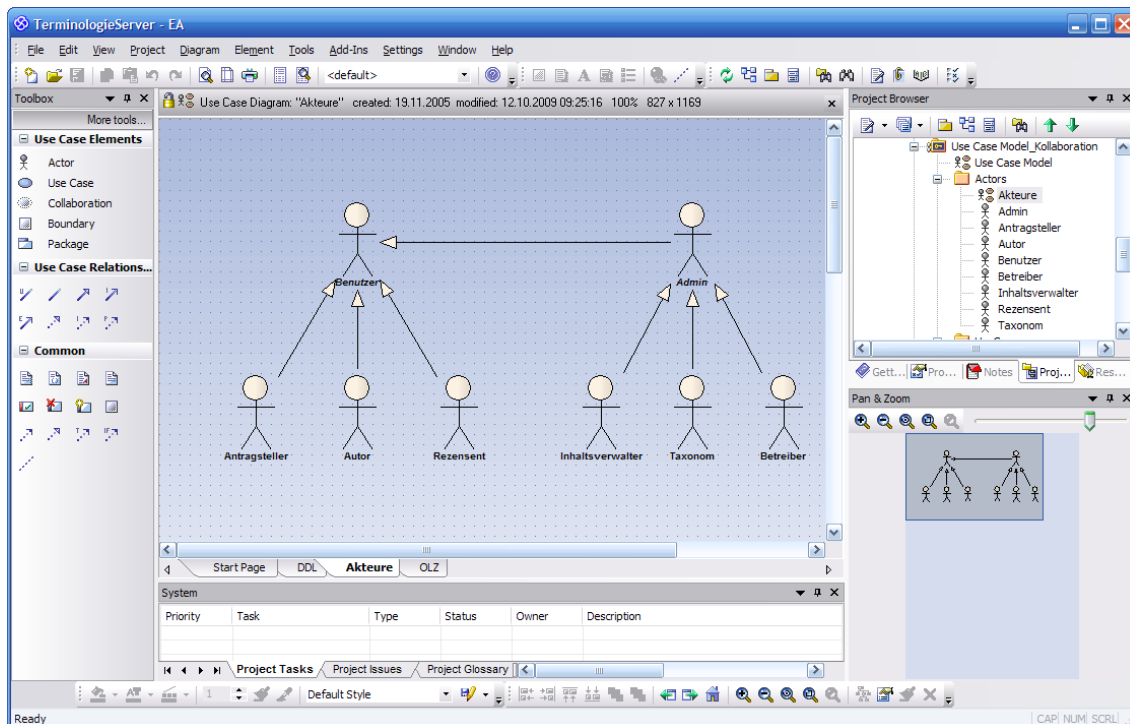


Abbildung 2-2 Enterprise Architect (Akteure)

Codegenerierung

Mit Hilfe des Enterprise Architects ist es möglich, aus den erstellten Modellen Programmcode in verschiedenen Programmiersprachen zu generieren. Aus dem OOD lassen sich die entsprechenden Java-Klassen mit den angegebenen Attributen und Methodenrumpfen erstellen.

Aus dem Datenmodell lässt sich ein SQL-Script erstellen, welches die Tabellen der Datenbank und ihre Beziehungen anlegt. Dieses vereinfacht die Erstellung der Datenbank sehr.

2.2.2 NetBeans IDE

NetBeans IDE ist eine mächtige Programmierumgebung, welche hauptsächlich für die Java-Entwicklung geschrieben wurde. NetBeans selbst wurde komplett in Java geschrieben und ist ein Open-Source-Projekt.

Der Vorteil von NetBeans ist, dass bereits die wichtigsten Plugins vorinstalliert sind. Auch die Übersichtlichkeit und Einfachheit ist besser als bei anderen Entwicklungsumgebungen für Java. Besonders hervorzuheben ist der Editor für graphische Oberflächen, welcher einer der besten für Java ist.

Für diese Masterarbeit kam NetBeans IDE in der Version 6.7.1 zum Einsatz.

2.2.3 Microsoft Excel

Beim Einsatz von Webservices ist es wichtig zu wissen, welche Attribute die einzelnen Dienste verlangen und welche Attribute diese liefern. Ein bewährtes Mittel dafür ist die Erstellung einer Dienste-Matrix. Auf der horizontalen Ebene werden die verfügbaren Dienste, auf der vertikalen Ebene werden die verwaltbaren Objekttypen mit ihren Attributen in hierarchischer Form angegeben. Im Inhalt der Matrix werden nun Pflichtfelder, Kannfelder sowie deren Kardinalitäten gekennzeichnet. Für die Erstellung einer solchen Matrix ist Excel gut geeignet.

		Format	Element / Attribut	Login (Request)	(Response)	Logout (Request)	(Response)	ListProposals (Request)	(Response)	GetProposalDetails (Request)	(Response)	CreateProposal (Request)	(Response)
2	XML-Node		Type										
3	ReturnInfos												
4	ReturnTypeInfo		ReturnTypeInfo	E	1	1	1	1	1	1	1	1	1
5	overallErrorCategory		Enum	A	1	1	1	1	1	1	1	1	1
6	status		Enum	A	1	1	1	1	1	1	1	1	1
7	message		String	A	1	1	1	1	1	1	1	1	1
8	Login												
9	LoginTypeInfo		LoginTypeInfo	E	1	1	1	1	1	1	1	1	1
10	sessionID		String	A	0..1	1	1	1	1	1	1	1	1
11	username		String	A	1								
12	password		String (MDS)	A	1								
13	Role												
14	RoleTypeInfo		List<RoleType>	E	0..n								
15	id		long	A	1								
16	name		String	A	1								
17	adminFlag		boolean	A	1								
18	Action												
19	ActionTypeInfo		List<ActionType>	E									

Abbildung 2-3 Microsoft Excel - Kollaborationsmatrix

Die Kardinalität wird bei einem Pflichtfeld mit einer „1“ oder einer „1..n“, bei einem Kannfeld mit einer „0..1“ oder „0..n“ gekennzeichnet. Anfrage (Request) und Antwort (Response) werden von einem Dienst nebeneinander dargestellt.

Bei diesem Projekt wurde Microsoft Excel in der neuesten Version 2007, SP2 eingesetzt.

2.2.4 Java

Java ist eine objektorientierte Programmiersprache und ein eingetragenes Zeichen der Firma SUN. Java zeichnet sich als Sprache besonders durch ihre Plattformunabhängigkeit aus. Java-Programme werden in Byte-Code übersetzt und danach in einer speziellen Laufzeitumgebung ausgeführt. Die Java Virtual Machine interpretiert diesen Byte-Code und führt ihn aus. Die Übersetzung des Bytecodes erfolgt mittels eines JIT-Compilers.

Die Sprache ist logisch aufgebaut und das SDK beinhaltet bereits viele Pakete für die Anwendung in unterschiedlichen Bereichen (Netzwerk, Grafik, verschiedene Collectionklassen³ etc.).

Weiterhin unterstützt die Sprache die Behandlung von Ausnahmen, Threads für nebenläufige Programmierung und eine automatische Speicherbereinigung (Garbage Collection).

Java lebt als objektorientierte Sprache von ihrer Einfachheit, da sie im Vergleich zu anderen Sprachen keine Zeiger oder Mehrfachvererbung unterstützt. Java besitzt eine enorme Robustheit, vor allem durch die starke Typisierung (verschiedene Datentypen können nicht ohne weiteres untereinander zugewiesen werden), durch Garbage Collection (der Speicher wird automatisch bereinigt) oder durch die gegebene Sicherheit (Java Security-Manager). Ein weiterer Vorteil ist die Plattformunabhängigkeit, welche auch in diesem Projekt deutlich wird. Die Kollaborationsanwendung wird unter Microsoft Windows entwickelt, die Services werden später jedoch auf einem Linux-Server laufen.

Dieses Projekt verwendet die zurzeit aktuellste Version von Java: Java 1.6.0_17

³ Collectionklasse - Vorgefertigte Containerklassen für transiente Vorhaltung von Objekten

2.2.5 Java-Applet

Ein Java-Applet ist eine Software, welche mit Java entwickelt wurde und im Webbrowser ausgeführt wird. Der Code wird dabei nicht auf dem Server, sondern auf dem Client-PC selbst ausgeführt. Im Gegensatz zu Servlets, welche auf dem Server laufen und scheinbar besser geeignet sind, bieten Applets jedoch bessere Möglichkeiten der Implementierung ergonomischer grafischer Oberflächen an.

Damit ein Applet auf dem Client lauffähig ist, muss ein Browser installiert sein, welcher die Java-VM unterstützt und installiert hat.

Die Sicherheit eines Applets ist im Vergleich zu „normalen“ Anwendungen höher, da ein Applet in einer abgeschotteten Laufzeitumgebung (Sandbox) läuft. Ein Zugriff auf das Dateisystem des Computers bleibt dem Applet verwehrt.

2.2.6 Log4j

Logging ist bei größeren Software-Projekten ein wichtiger Bestandteil, da es bei der Behebung von Fehlern viele Vorteile bietet. Log4j (<http://logging.apache.org/log4j/docs/index.html>, letzter Zugriff am 09.02.2010) ist ein Framework zum Loggen von Anwendungsmeldungen in Java. Meldungen können mit unterschiedlichen Prioritäten geloggt werden (Stufen): Trace, Debug, Info, Warn, Error, Fatal. Es kann beim Programmstart bestimmt werden, dass nur ab einer bestimmten Stufe geloggt wird. So ist es möglich, sich im Testbetrieb alle nötigen Info- und Debug-Warnungen anzeigen zu lassen. Im späteren Betrieb werden diese Meldungen dann aus Performance-Gründen nicht gespeichert oder ausgegeben.

Für das Format der Log-Dateien sind verschiedene Appender zuständig. So ist es z.B. mit Hilfe des FileAppenders möglich, Meldungen in eine Datei zu speichern. Es gibt andere Appender, welche unter anderem Emails verschicken können bei schwerwiegenden Fehlern. Es ist möglich, mehrere Appender parallel zu schalten, so dass Meldungen auf unterschiedliche Weise gleichzeitig geloggt werden können. Dieses ist gerade interessant, wenn Meldungen auch auf der Konsole ausgegeben werden sollen.

2.2.7 MySQL

Das Produkt MySQL (<http://www.mysql.de/>, letzter Zugriff am 09.02.2010) der Firma „Sun Microsystems“ ist ein relationales Datenbanksystem, welches frei unter einer Open-Source-Lizenz vertrieben wird. Die Software ist für viele verschiedene Betriebssysteme verfügbar und ist die Basis vieler dynamischer Webauftritte. MySQL darf nur im nicht-kommerziellen Bereich frei benutzt werden.

Dem Datenbankmanagementmodul von MySQL können mehrere Datenbanken und darin Tabellen zugeordnet werden. Die Größe dieser Datenbanken wird lediglich durch das verwendete Betriebssystem beschränkt.

Die Datenbanken lassen sich über die mitgelieferten Kommandozeilentools „mysql“ und „mysqladmin“ verwalten. Inzwischen werden jedoch auch grafische Tools („MySQL Administrator“, „MySQL Query Browser“) angeboten, um die Datenbanken und deren Tabellen zu verwalten und um Sicherungskopien der bestehenden Datenbankstruktur anlegen zu können.

Eine Datenbank ist die Grundlage für einen Terminologieserver sowie der Kollaborationsumgebung. Der Zugriff auf die Daten erfolgt in beiden Projekten per Webservice.

Das eigentliche Datenbankskript wurde aus dem Enterprise Architect generiert und in das Datenbanksystem eingespielt.

2.2.8 Web-Services (Web-Dienste)

Das WWW, besser bekannt als das Internet oder weltweites Netz, wird immer mehr für die Kommunikation unter Anwendungen verwendet. Ein Webservice ist eine Anwendungsschnittstelle, welche weltweit über eine URI abrufbar ist. Die Schnittstelle des Programms ist im XML-Format beschrieben. Dies soll eine einfache Bereitstellung und hohe Verfügbarkeit von Web Services gewährleisten. Das erklärte Ziel ist die Interoperabilität von Softwaresystemen, um unabhängig von Plattform und Programmiersprachen miteinander kommunizieren und arbeiten zu können.

Das Ansprechen eines Webservices arbeitet nach folgendem Prinzip:

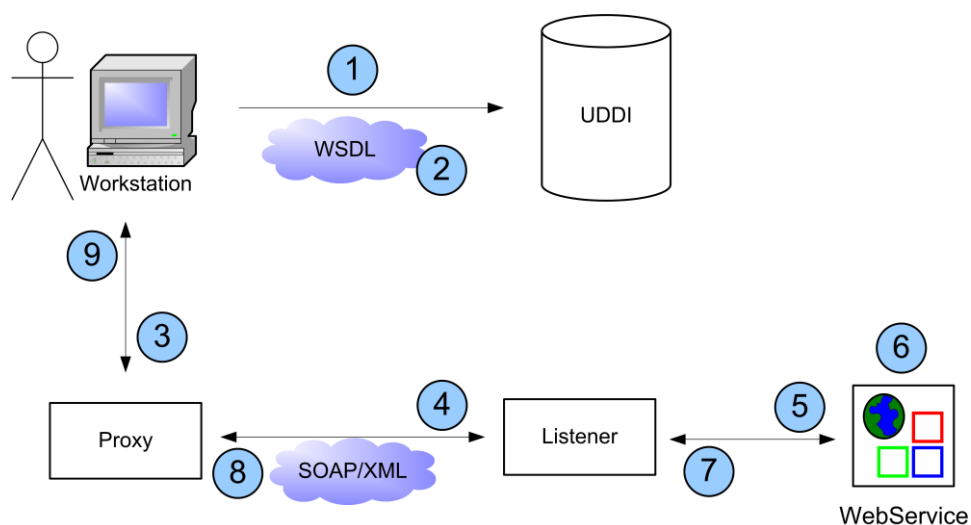


Abbildung 2-4 Funktionsweise eines Webservices

1. Discovery

Die Discovery-Ebene stellt Funktionalitäten zur Verfügung, um Web Services sowie die dazugehörigen Beschreibungen zu finden. Prinzipiell steht mit UDDI das meist genutzte Verfahren zur Verfügung. Die aktuellsten Standards sind auf der Webseite <http://uddi.xml.org/> verfügbar.

2. Description

Um für einen Web Services zu beschreiben, welche Protokolle für die einzelnen Schichten unterstützen, werden die Informationen mittels WSDL dargestellt. Die aktuellen Standards werden auf der Seite <http://www.w3.org/2002/ws/desc/> veröffentlicht.

3. Proxy Creation

Ein lokales Proxy wird zu der entfernten Seite erstellt. Proxys besitzen keinen eigenen Standard sondern werden anhand von SOAP Nachrichten und XML verschickt.

4. Soap Message Creation

Bei diesem Schritt wird eine SOAP/XML-Nachricht erstellt und an die URL geschickt, welche in der WSDL-Datei spezifiziert wurde. Letzte Standards sind auf der Seite <http://www.w3.org> zu finden.

5. Listener

Ein SOAP Listener auf der Host-Seite empfängt alle Aufrufe und interpretiert diese für den Web Service.

6. Web Service

Der Web Service ist das Programm an sich und stellt die Funktionalität zur Verfügung.

7. Rückweg

Der Web Service gibt seine Antwort – falls vorhanden – an den Listener zurück.

8. Rückweg (Proxy)

Der Listener leitet die Antwort in Form einer SOAP-Nachricht an den Proxy weiter. Auch, wenn der Web Service keine Antwort zurückgegeben hat, wird eine SOAP-Nachricht zurückgegeben mit weiteren Informationen, dass der Web Service erfolgreich aufgerufen worden ist.

9. Rückweg (Anwendung)

Die Anwendung erhält vom Proxy die Antwort vom Webservice.

2.2.9 Tomcat

Tomcat (<http://tomcat.apache.org/>, letzter Zugriff am 15.02.2010) der „Apache Software Foundation“ ist ein Webserver, welcher eine Umgebung zur Ausführung von Java-Code bereitstellt. Mit Tomcat ist es möglich, Webservices ausführen zu können. Die Anfragen werden an den Java-Code weitergeleitet und die Antworten wieder zurück an den Aufrufer weitergegeben. Tomcat ist für mehrere Plattformen wie Windows, Linux, Unix oder MacOS erhältlich. Auch in diesem Projekt wird zunächst unter Windows entwickelt, die Umgebung soll später jedoch auf einem Linux-System laufen.

Das Produkt steht unter der Apache-Lizenz (freie Software).

2.3 Projektplanung und -verlauf

Bei der Entwicklung der webbasierten Kollaborationsumgebung für den Terminologieserver stellt sich wie bei vielen Projekten die Frage der zu verwendenden Technologien und Werkzeuge. Im Folgenden werden diese mit Begründungen genannt.

Die Datenbank

Zunächst wird ein Datenbankmodell mithilfe des Tools Enterprise Architect (2.2.1) modelliert, welches anschließend zu einem Skript umgewandelt wird. Das Skript wird im Tool MySQL (2.2.7) eingefügt und die Datenbank wird daraus erstellt.

Analyse

Was soll das Programm leisten? Wie sehen Lösungsvorschläge aus? Diese Antworten gibt es im Kapitel 3, Analyse. Es werden Akteure, Anwendungsfälle sowie das Verhalten der Anwendung analysiert und erarbeitet. Die nötigen Dienste werden hier erörtert und die Anforderungen in Dienstematrizen dargestellt.

Design

Das Design beschäftigt sich mit der Entwicklung von Modellen wie dem Klassen- und Datenbankmodell. Der Grundstein für die Anwendung wird in diesem Kapitel beschrieben.

Implementierung

In der Implementierungsphase müssen die Webservices, welche die Datenbankanbindung realisieren, und das Java-Applet realisiert werden. Eine Anbindung an den Terminologieserver muss zusätzlich per Webservices vorhanden sein, da die Kollaborationsumgebung auf diesen zugreifen muss. Schließlich muss die Anwendung mithilfe verschiedener Tests gründlich geprüft werden.

Test

In diesem Kapitel wird die Anwendung anhand eines kleinen Beispiel-Szenarios getestet. Die Ergebnisse werden zeigen, ob die Anwendung sinnvoll einsetzbar ist.

2.3.1 Zeitplan

Der folgende Zeitplan zeigt den Verlauf dieser Thesis, also alle Aufgaben und Termine. Der Zeitplan hilft bei der Abarbeitung der Aufgaben in der richtigen Reihenfolge und der angegebenen Zeit.

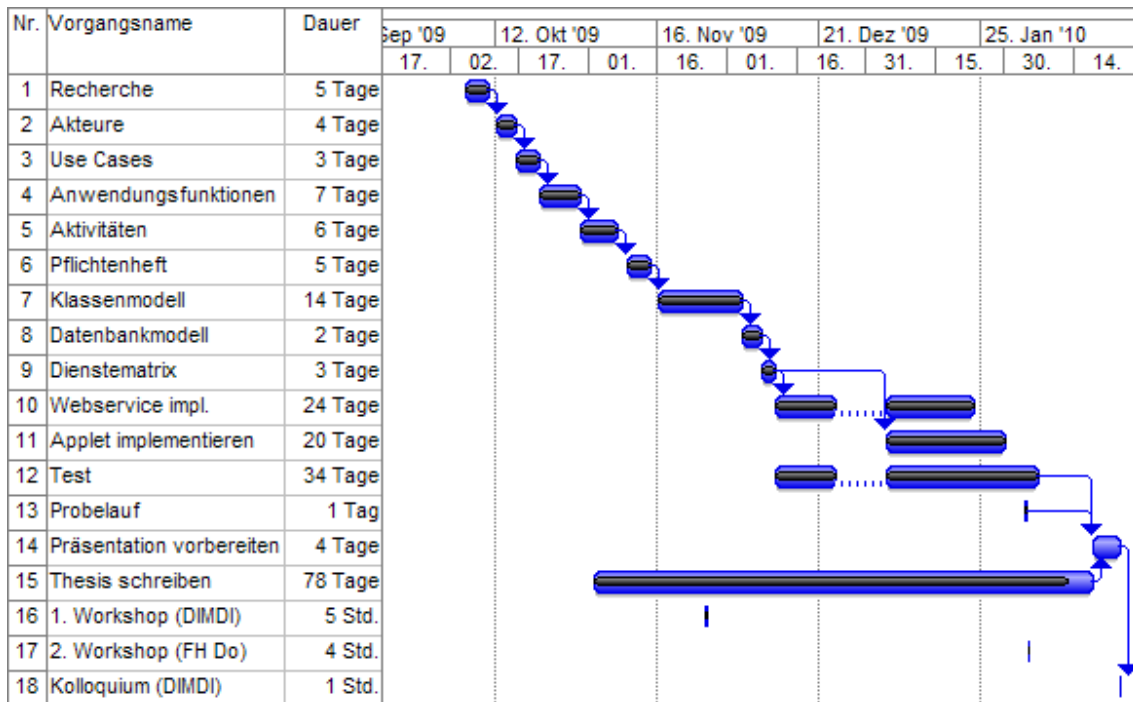


Abbildung 2-5 Projekt-Zeitplan

Kommt es bei einer Aufgabe zu Verschiebungen, ist schnell zu erkennen, an welcher Stelle die verlorene Zeit wieder eingeholt werden kann.

Erledigte Aufgaben sind in der Abbildung durchgestrichen. Zu dem Zeitpunkt der Erstellung dieses Dokuments sind die letzten Aufgaben noch nicht abgeschlossen.

3 Analyse

Die Analyse umfasst von der Recherche bis hin zur Festlegung von Anforderungen für die Kollaborationsumgebung des Terminologieservers alle Aspekte der Herausarbeitung von Einsatzbedingungen und Anforderungen. Neben der Beschreibung der Akteure, den Anwendungsfällen und dem Verhalten der Anwendung werden Funktionen der Anwendung in einem Pflichtenheft festgehalten.

3.1 Akteure

Die Einteilung der Akteure der Kollaborationsumgebung ist am bestehenden Modell einer Wissensdatenbank [Quelle: <http://de.wikipedia.org/wiki/Wissensdatenbank>, letzter Zugriff 12.02.2010] orientiert. Eine Wissensdatenbank beinhaltet vordefinierte Benutzerrollen. Die Hauptaufgabe einer Wissensdatenbank ist die Unterstützung der Zusammenarbeit von mehreren Personen.

Die folgende Abbildung zeigt alle Akteure, die für eine Wissensbank angegeben werden:

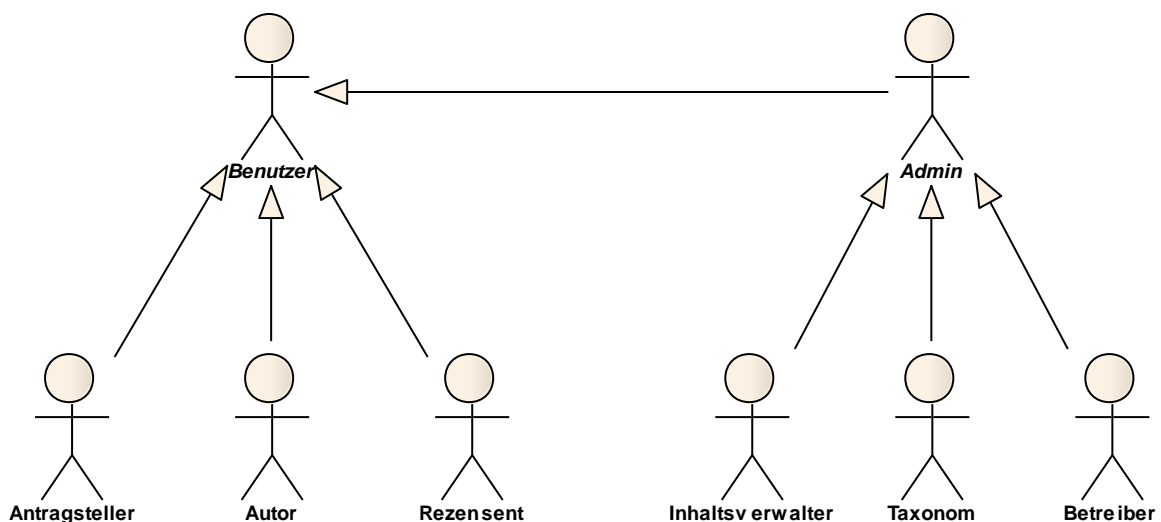


Abbildung 3-1 Übersicht der Akteure

In den nachfolgenden Unterkapiteln werden alle Akteure näher erläutert.

3.1.1 Benutzer

Der Benutzer ist ein abstrakter Akteur. Das bedeutet, es gibt in konkret nicht. Implementierungstechnisch stellt dieser Akteur jedoch Eigenschaften bereit, welche an andere Akteure vererbt werden. Der Benutzer dient also als „Basis“ für alle anderen Benutzer.

3.1.1.1 Antragsteller

Der Antragsteller beantragt neue Inhalte (Vorschläge) oder Änderungen für ein vorhandenes Vokabular oder aber beantragt ganz neue Vokabulare. Vorschläge können also aus neuen Begriffen, Vokabularen oder Beziehungen sowie weiteren Anwendungsfällen bestehen.

3.1.1.2 Autor

Der Autor erstellt den Inhalt von genehmigten Anträgen. Seine Rolle ist es, Konzepte/Begriffe in ein Vokabular einzutragen bzw. dort freizugeben, verschiedene Beziehungen der Terminologien auszuarbeiten und dergleichen. Der Autor kann die gleiche Person wie der Antragsteller sein.

3.1.1.3 Rezensent

Der Rezensent übernimmt die Aufgabe eines Korrekturlesers. Er überprüft nach dem Vier-Augen-Prinzip die Neueinträge der Autoren auf Fehler jeglicher Art, seien es semantische-, grammatikalische- oder Rechtschreibfehler. Anschließend erfolgt eine Freischaltung bzw. Veröffentlichung durch den Inhaltsverwalter (3.1.2.1).

3.1.2 Administrator

Wie der Akteur „Benutzer“ ist der „Administrator“ eine abstrakte Rolle und gibt seine Eigenschaften an alle Akteure bzw. konkrete Benutzer weiter, die Administrationsrechte haben sollen. Darunter fallen der Inhaltsverwalter, der „Taxonom“ und der „Betreiber“.

3.1.2.1 Inhaltsverwalter

Die Aufgabe des Inhaltsverwalters ist die Freischaltung von beantragten Einträgen und die Veröffentlichung von bearbeiteten Einträgen sowie ggf. die eigenständige Pflege von Vokabularen. Er verwaltet den Inhalt der Antragsteller. Dabei muss er auf folgende Aspekte der Einträge achten:

- Überprüfung der Eignung der Einträge

- Duplikate bei gleichartigen Einträgen in der selben Terminologie extrahieren
- Freischaltung und Veröffentlichung der Einträge

3.1.2.2 Taxonom

Die Rolle des Taxonom ist nicht zwingend erforderlich, dient jedoch der Qualitätssicherung. In manchen Fällen ist sie identisch mit der des Inhaltsverwalters.

Der Taxonom legt Normen fest, welche die Konsistenz der Inhalte zum Zweck haben. Er setzt z.B. lokale Standards für die Begriffskontrolle und die Zusammensetzung in einer Terminologie oder er definiert Wörterbücher, welche als Vokabular verwendet werden.

3.1.2.3 Betreiber

Die Rolle des Betreibers ist zwingend erforderlich. Er übernimmt keine inhaltlichen Aufgaben der Kollaborationsumgebung. Sein Auftrag ist es, die Plattform zu betreiben, dazu zählen Aufgaben wie

- Bereitstellung der Infrastruktur, die
- Erstellung und Wartung des Informationssystems (Funktionen, Oberfläche usw.) oder die
- Organisation von Weiterbildungen, unter anderem zur Informationsbeschaffung

3.2 Anwendungsfälle (Use Cases)

In der folgenden Abbildung sind alle Anwendungsfälle der Kollaborationsplattform dargestellt. Es werden die beteiligten Akteure angezeigt. Es ist zu erkennen, welcher Akteur Zugriff auf welche Anwendungsfälle besitzt.

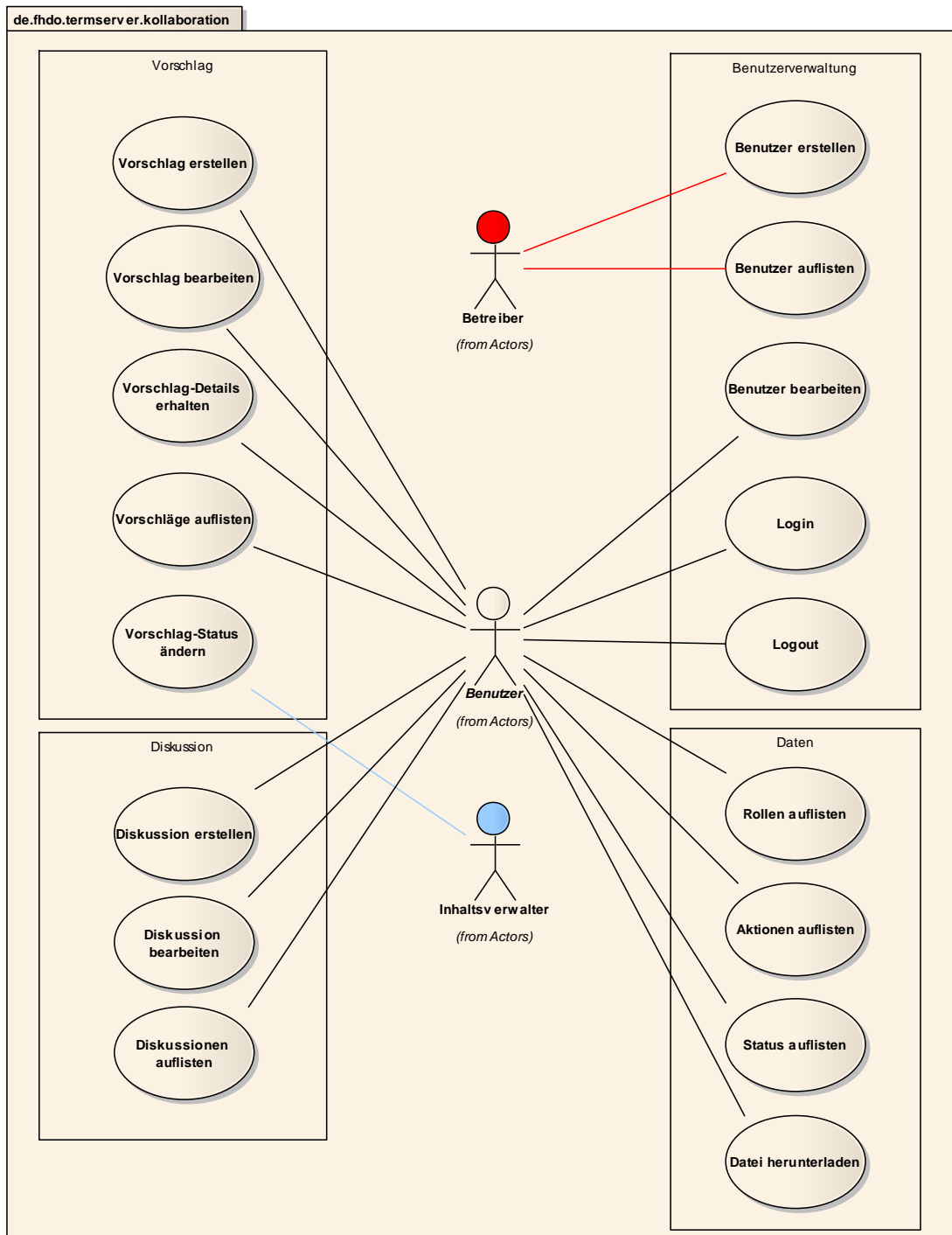


Abbildung 3-2 Übersicht der Anwendungsfälle (Use Cases)

Die Anwendungsfälle werden aufgeteilt in die Bereiche Vorschlag, Diskussion, Benutzerverwaltung und Daten. Diese werden im Folgenden näher erläutert.

3.2.1 Vorschlag

Name Vorschlag erstellen (UC 1.01)

Beschreibung	Ein neuer Vorschlag wird erstellt.
Service	CreateProposal
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Der Vorschlag befindet sich im Status „Vorgeschlagen“.
Standardablauf	Ein beliebiger Benutzer teilt einen Änderungswunsch an einem Vokabular mit. Dazu gehören Details zum Vorschlag selbst, Links, Terminologie, Objekte und Privilegien.
Alternativen	-
Hinweise	-

Name Vorschlag bearbeiten (UC 1.02)

Beschreibung	Ein vorhandener Vorschlag wird bearbeitet.
Service	MaintainProposal
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Vorschlag erstellen (UC 1.01), Vorschläge auflisten (UC 1.04)
Ergebnis	-
Standardablauf	Der Besitzer des Vorschlags möchte Änderungen an diesem vornehmen. Er kann Links hinzufügen, bearbeiten oder löschen. Er kann anderen Benutzern Rechte für diesen Vorschlag erteilen.
Alternativen	-
Hinweise	-

Name	Vorschlag-Details erhalten (UC 1.03)
Beschreibung	Es werden alle Details zu einem Vorschlag abgerufen.
Service	GetProposalDetails
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Vorschläge auflisten (UC 1.04)
Ergebnis	Ein Vorschlag inkl. aller Details
Standardablauf	Ein Benutzer möchte einen für ihn bestimmten Vorschlag ansehen. Nach dem Listen der Vorschläge können mithilfe der ID Details zu einem Vorschlag abgerufen werden.

Name	Vorschläge auflisten (UC 1.04)
Beschreibung	Es werden Vorschläge inkl. Filterkriterien aufgelistet.
Service	ListProposal
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), [Vorschlag erstellen (UC 1.01)]
Ergebnis	Liste von Vorschlägen
Standardablauf	Der Dienst wird aufgerufen, wenn die Schreibtischansicht eines Benutzers erstellt wird.

Name	Vorschlag-Status ändern (UC 1.05)
Beschreibung	Der Status eines Vorschlags wird geändert.
Service	UpdateProposalStatus
Akteure	Inhaltsverwalter
Vorbedingung	Einloggen (UC 3.01), Vorschläge auflisten (UC 1.04)
Ergebnis	-
Standardablauf	Der Inhaltsverwalter möchte den Status des Vorschlags ändern. Die Statusänderung wird dabei zusätzlich gespeichert, um Änderungen verfolgen zu können.

3.2.2 Diskussion

Name Diskussion erstellen (UC 2.01)

Beschreibung	Eine neue Diskussion wird erstellt.
Service	CreateDiscussion
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Vorschlag-Details erhalten (UC 1.03)
Ergebnis	-
Standardablauf	Ein Vorschlag wird geöffnet. Wenn sich dieser im Status „In Diskussion“ befindet, können Diskussionseinträge (Foreneinträge) vorgenommen werden.

Name Diskussion bearbeiten (UC 2.02)

Beschreibung	Eine Diskussion wird bearbeitet.
Service	MaintainDiscussion
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Diskussionen auflisten (UC 2.03)
Ergebnis	-
Standardablauf	Eine Diskussion (Forenbeitrag) kann vom Besitzer geändert werden. Es können zusätzliche Dokumente oder Links zugefügt werden.

Name Diskussionen auflisten (UC 2.03)

Beschreibung	Eine neue Diskussion wird erstellt.
Service	ListDiscussions
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Vorschlag-Details erhalten (UC 1.03)
Ergebnis	Eine Liste von Diskussionen (Foreneinträge).
Standardablauf	Die Forenansicht wird geladen, dabei werden alle Diskussionen aufgelistet.

3.2.3 Benutzerverwaltung**Name** Einloggen (UC 3.01)

Beschreibung	Ein Benutzer meldet sich am System an.
Service	Login
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	keine
Ergebnis	Der Benutzer erhält eine Session-ID.
Standardablauf	Ein Benutzer möchte sich am System mit Benutzernamen und Passwort anmelden. Bei Erfolg erhält er eine Session-ID, welche er bei anderen Use-Cases nutzen muss.

Name Ausloggen (UC 3.02)

Beschreibung	Ein Benutzer meldet sich vom System ab.
Service	Logout
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Der Benutzer ist vom System abgemeldet, seine Session-ID wird gelöscht.
Standardablauf	Der Benutzer meldet sich vom System anhand seiner Session-ID ab.

Name Benutzer erstellen (UC 3.03)

Beschreibung	Ein neuer Benutzer wird in der Kollaborationsumgebung erstellt.
Service	CreateUser
Akteure	Betreiber
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Ein neuer Benutzer wurde erstellt, welcher auf Wunsch per Email benachrichtigt wurde.
Standardablauf	Der Betreiber erstellt einen neuen Benutzer mit den wichtigsten Attributen.

Name	Benutzer auflisten (UC 3.04)
Beschreibung	Vorhandene Benutzer werden aufgelistet.
Service	ListUser
Akteure	Betreiber
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Eine Liste von Benutzern.
Standardablauf	Der Betreiber ruft die Administrationsoberfläche auf und eine Liste von Benutzern wird angezeigt.

Name	Benutzer bearbeiten (UC 3.05)
Beschreibung	Ein vorhandener Benutzer wird bearbeitet.
Service	MaintainUser
Akteure	Betreiber, Benutzer
Vorbedingung	Einloggen (UC 3.01), Benutzer auflisten (UC 3.04)
Ergebnis	-
Standardablauf	Ein Benutzer ändert seine Stammdaten oder ein Betreiber ändert Rollenzugehörigkeiten des Benutzers.

3.2.4 Daten

Name	Rollen auflisten (UC 4.01)
Beschreibung	Rollen der Kollaborationsanwendung werden aufgelistet.
Service	ListRoles
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Eine Liste von Rollen.
Standardablauf	Für verschiedene Auswahllisten werden die vorhandenen Rollen benötigt.

Name	Aktionen auflisten (UC 4.02)
Beschreibung	Es werden alle Aktionen der Kollaborationsanwendung aufgelistet.
Service	ListActions
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Eine Liste von Aktionen.
Standardablauf	Für verschiedene Auswahllisten werden die vorhandenen Aktionen benötigt.

Name	Status auflisten (UC 4.03)
Beschreibung	Es werden alle Status der Kollaborationsanwendung aufgelistet.
Service	ListStatus
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01)
Ergebnis	Eine Liste mit allen Status.
Standardablauf	Für verschiedene Auswahllisten werden die vorhandenen Status benötigt.

Name	Datei herunterladen (UC 4.04)
Beschreibung	Eine Datei wird herunter geladen.
Service	GetLinkData
Akteure	Benutzer (inkl. alle abgeleiteten Akteure)
Vorbedingung	Einloggen (UC 3.01), Vorschlag-Details erhalten (UC 1.03)
Ergebnis	Eine Datei
Standardablauf	Ein Benutzer hat Links zu einem Vorschlag oder einer Diskussion gelistet. Er möchte eine vorhandene Datei aus der Datenbank herunterladen.

3.3 Anwendungsfunktionen und Navigation

Der Programmablauf zeigt mögliche Programmfenster mit ihren Zusammenhängen bzw. der möglichen Navigation. Die Anwendung ist neben dem Login grob in 2 Teile aufgeteilt: Desktop und Vorschlag. Zwischen diesen beiden Bereichen kann hin- und her gewechselt werden.

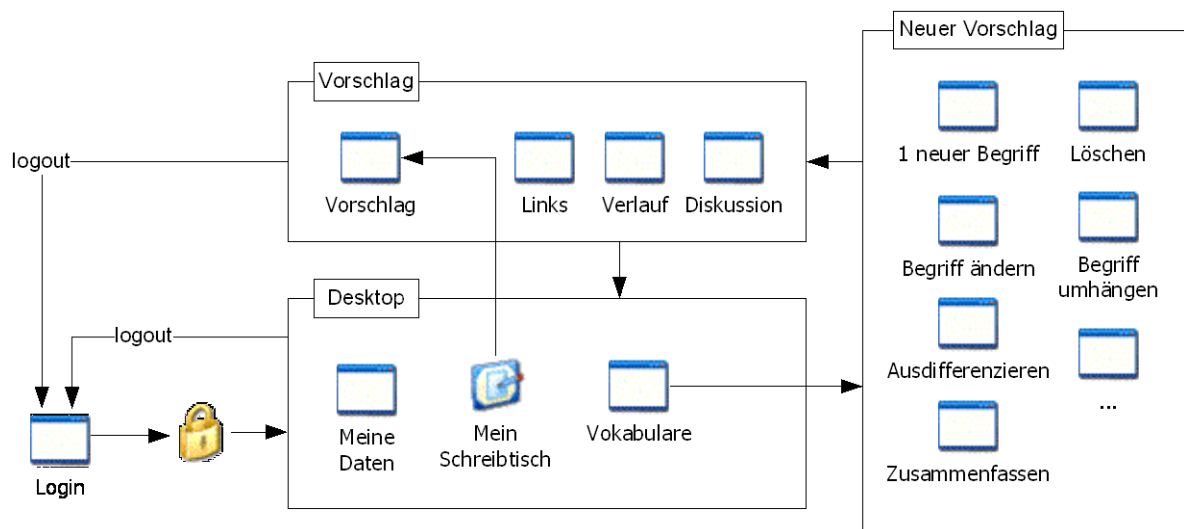


Abbildung 3-3 Programmablauf

In dem Desktop-Bereich sind alle Informationen zu einem Benutzer zu finden. Er kann dort seine eigenen Daten ändern, durch Vokabulare blättern oder kann sich Vorschläge auf seinem Schreibtisch ansehen, an denen er beteiligt ist.

Im Vorschlag-Bereich sind alle Informationen zu einem Vorschlag zu finden. Es kann festgelegt werden, welche Benutzer Zugriff auf diesen Vorschlag haben dürfen und sich an der Diskussion beteiligen können. Die Diskussion stellt einen Kernpunkt der Anwendung dar, da sie Anwender unterstützt, Entscheidungen zu treffen. Zusätzlich lassen sich zu jedem Vorschlag Dokumente in Form von Dateien oder Links anhängen. Der Verlauf zeigt alle Status-Änderungen des Vorschlags. Berechtigte Nutzer können in diesem Bereich den Status eines Vorschlags ändern. Welche Statusänderungen möglich sind, ist unter dem Punkt 3.4 Verhalten/Aktivitäten nachzulesen.

Neue Vorschläge lassen sich von allen Benutzern erstellen. Nach einer Prüfung des Inhaltsverwalters erscheinen diese auf dem Schreibtisch der berechtigten Nutzer. Es gibt mehrere Typen von Vorschlägen wie „Neuer Begriff“, „Begriff umhängen“ usw.

3.4 Verhalten/Aktivitäten

Die Kollaborationsumgebung lebt von Vorschlägen. Ein Vorschlag besitzt einen Status, welcher vom Akteur „Inhaltsverwalter“ nach bestimmten Regeln geändert werden kann. Die Änderung geschieht im Anwendungsfall „Vorschlag-Status ändern (UC 1.05)“. Welche Statusänderungen möglich sind, muss in der Datenbank bzw. der Anwendung parametrisiert werden können. Für die Kollaborationsumgebung kann folgend dargestelltes Statusdiagramm angegeben werden:

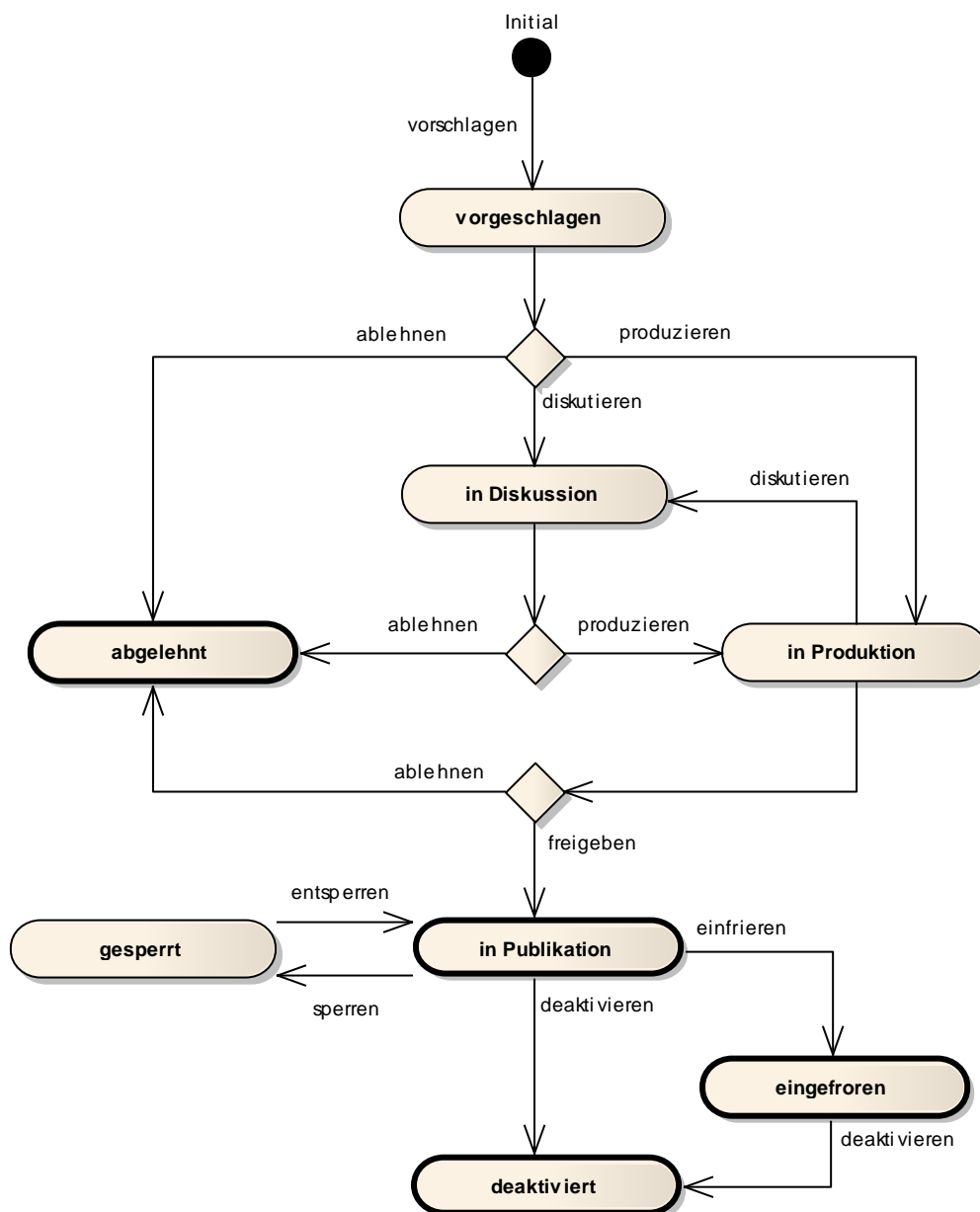


Abbildung 3-4 Aktivitätsdiagramm: Objektlebenszyklus

3.4.1 Status

Im Folgenden werden alle Status aus der Abbildung 3-4 Aktivitätsdiagramm: Objektlebenszyklus näher erläutert. Status-Objekte werden in dem Diagramm in Form von Rechtecken mit abgerundeten Ecken dargestellt. Status-Objekte mit dicken Linien sind Final-Status, d.h. von ihm aus kann der Status-Verlauf enden.

Wenn im Folgenden von Vorschlägen gesprochen wird, sind die Vorschläge inklusive ihrer Objekte, also Vokabulare, Begriffe oder Beziehungen, gemeint.

3.4.1.1 Vorgeschlagen

Der Status „Vorgeschlagen“ ist noch nicht für alle Benutzer sichtbar. Der Inhaltsverwalter prüft diesen Vorschlag und gibt ihn entweder frei oder lehnt ihn ab. Die Freigabe erfolgt entweder direkt zur Diskussion oder zur Produktion, wenn die Diskussion übersprungen werden soll.

3.4.1.2 Abgelehnt

Der Vorschlag erhält diesen Status, wenn er abgelehnt wurde.

3.4.1.3 In Diskussion

Bei diesem Status ist die Diskussion freigeschaltet. Alle privilegierten Benutzer können sich an der Diskussion in Form eines Forums beteiligen. Dieser Status hat meist eine bestimmte Gültigkeitsdauer. Danach muss entschieden werden, ob der Vorschlag angenommen oder abgelehnt wird.

3.4.1.4 In Produktion

Die Diskussion ist beendet. Dieser Status ist eine Vorstufe zu „Veröffentlicht“. Der Vorschlag ist akzeptiert, jedoch noch nicht veröffentlicht. Er kann zusätzlich erneut zur Diskussion gestellt werden und bearbeitet werden.

3.4.1.5 In Publikation

Der Vorschlag ist publiziert. Er ist öffentlich zugänglich. Er befindet sich in einem Finalstatus. Dieser Zustand tritt ein, wenn ein Vokabular veröffentlicht wird und nicht mehr geändert werden kann.

3.4.1.6 Gesperrt

Ein Vorschlag ist gesperrt und nicht nach außen hin öffentlich zugänglich.

3.4.1.7 Eingefroren

„Eingefroren“ ist eine Vorstufe von Deaktivieren. Der Status kann verwendet werden, wenn es bereits neuere Versionen gibt und diese als „veraltet“ dargestellt wird. Anschließend ist nur noch ein „Deaktivieren“ möglich.

3.4.1.8 Deaktiviert

Bei diesem Status ist das Vokabular, der Begriff oder die Beziehung für immer gesperrt und nicht mehr nach außen zugänglich.

3.4.2 Aktionen

Aktionen führen zu Statusübergängen. Alle Aktionen werden in der Anwendung in Form von Buttons realisiert. Der Benutzer schaltet somit manuell von einem Status zum nächsten Status. In der Abbildung 3-4 Aktivitätsdiagramm: Objektlebenszyklus sind Aktionen an den Pfeilen als Text eingetragen. Sie sind in Pfeilrichtung von einem Status zum anderen Status zu lesen.

3.4.3 Statuswechsel

Die *Abbildung 3-4 Aktivitätsdiagramm: Objektlebenszyklus* zeigt die Vorgehensweise der Statusabläufe. Die folgende Tabelle zeigt alle möglichen Statuswechsel der Kollaborationsumgebung, die initial parametrisiert wurden.

Nr.	Status Von	Status Nach	Aktion	Rollen
1	Initial	Vorgeschlagen	Vorschlagen	Antragsteller
2	Vorgeschlagen	In Diskussion	Diskutieren	Inhaltsverwalter
3	Vorgeschlagen	Abgelehnt	Ablehnen	Inhaltsverwalter
4	Vorgeschlagen	In Produktion	Produzieren	Inhaltsverwalter
5	In Diskussion	In Produktion	Produzieren	Inhaltsverwalter
6	In Diskussion	Abgelehnt	Ablehnen	Inhaltsverwalter
7	In Produktion	In Diskussion	Diskutieren	Inhaltsverwalter Rezensent
8	In Produktion	In Publikation	Freigeben	Inhaltsverwalter
9	In Produktion	Abgelehnt	Ablehnen	Inhaltsverwalter
10	In Publikation	Gesperrt	Sperren	Inhaltsverwalter
11	In Publikation	Deaktiviert	Deaktivieren	Inhaltsverwalter
12	In Publikation	Eingefroren	Einfrieren	Inhaltsverwalter
13	Gesperrt	In Publikation	Entsperren	Inhaltsverwalter
14	Eingefroren	Deaktiviert	Deaktivieren	Inhaltsverwalter

Tabelle 3-1 Mögliche Statusänderungen

In der Tabelle sind Rollen angegeben, welche den Status ändern dürfen. In der Datenbank werden diese Transitionen mit ihren Berechtigungen einmalig eingegeben, können aber auch je nach Verwendungszweck angepasst werden, so dass der Betreiber eine möglichst große Freiheit zur Abbildung der Ablauforganisation zur Verfügung hat.

In welcher Form diese Tabelle in der Datenbank abgebildet ist, wird im Kapitel 4 Design erläutert.

3.5 Benutzerfunktionen / Pflichtenheft

3.5.1 Zielbestimmung

3.5.1.1 *Muss-Kriterien*

- Analyse
 - ... der vorhandenen Workflows bzw. Organisationsprinzipien in den einschlägig national tätigen Organisationen bzgl. derer Terminologieentwicklung
 - ... der Anforderungen und Wünsche der potentiellen Endbenutzer (was, wie, wann) → Use Case Beschreibungen
 - ... Analyse ggf. vorhandener Lösungsansätze (Literatur, DIMDI usw.)
 - ... Identifikation ggf. zu berücksichtigender Schnittstellen
- Werkzeuggestützte Spezifikation und Design
 - Gesamtkonzept, Architektur
 - Use Cases
 - Klassenmodell und Objektlebenszyklen
 - Oberflächendesign
 - → Anwenderorientierte Entwürfe – hier vor allem die Anwendungsoberfläche - sind mit ausgewählten Handlungsträgern im Rahmen eines Workshops zu diskutieren und zu arrondieren.
- Werkzeuggestützte Implementierung einer Web-Anwendung u.A.
 - ... der notwendigen Datenbankstrukturen
 - ... der notwendigen Anwendungsfunktionen und Algorithmen (Stufe 1)
- Test
 - Test der Anwendung im Labor und mit ausgewählten Anwendern

3.5.1.2 *Kann-Kriterien*

- Inbetriebnahme für Evaluation / laborhafter Evaluationsbetrieb
 - Inbetriebnahme des Lösungsansatzes mit einem kleinen Kreis von Anwendern über 2 Wochen und
 - Evaluation des Einsatzes (Fehler, Akzeptanz, Nutzeneffekte etc.)

3.5.1.3 Abgrenzungskriterien

Es sollen keine Terminologien erstellt werden, da diese bereits vorhanden sind. Auf den Datenschutz und die Sicherheit soll aufgrund der sehr umfangreichen Implementierung zunächst keine hohe Priorität gelegt werden.

3.5.2 Einsatz

3.5.2.1 Anwendungsbereiche

Das Programm ist für die nationale Terminologiearbeit im Gesundheitswesen vorgesehen. Es unterstützt die Entwicklung von Terminologien in Form einer Kollaborationsplattform.

3.5.2.2 Zielgruppen

Primäre Zielgruppe der Anwendung sind alle Personen und Organisationen, welche an einer Terminologie entwickeln oder diese nutzen und einzelne Vorschläge einbringen wollen. Das können sowohl Herausgeber als auch Nutzer von Terminologien sein. Nutzer können z.B. Ärzte oder Krankenhäuser sein.

3.5.2.3 Betriebsbedingungen

Die Software ist für den Einsatz auf einem Server vorgesehen. Dabei können Terminologieserver sowie Kollaborationsserver auf unterschiedlichen Servern liegen, um Ressourcen zu sparen. Für den performanten Einsatz sollten sich beide Server jedoch in einem gemeinsamen Netzwerk befinden, da zwischen ihnen Kommunikation besteht.

3.5.3 Umgebung

3.5.3.1 Software

Es müssen folgende Komponenten vorhanden sein:

- Java Runtime Environment in der Version 1.6 (oder höher)
- Tomcat-Server (oder anderer Server für Webservices)
- MySQL-Datenbank

3.5.3.2 Hardware

Ein System mit ausreichend Ressourcen für den Betrieb von 2 Webservices inklusive deren Datenbankbindung muss vorhanden sein. Der oder die Server müssen einem erhöhten Datentransfer stand halten.

3.5.4 Funktionalität

Das System muss Anwendern eine Zusammenarbeit an Terminologien ermöglichen. Benutzer sollen über verschiedene Vorschläge diskutieren können. Spezielle Rollen sind für die administrativen Tätigkeiten vorgesehen. Sie koordinieren die Kollaboration.

Desktop-Bereich

Der Desktop-Bereich soll die Bereiche „Meine Daten“, „Mein Schreibtisch“, „Vokabulare“ und, für Administratoren, „Admin“ enthalten. Bei „Meine Daten“ werden eigene Angaben zur Person, Adresse und Organisation verwaltet. Auf dem eigenen Schreibtisch werden alle Vorschläge angezeigt, zu denen der Benutzer zugeordnet wurde. Bei den Vokabularen werden alle Vokabulare des Terminologieservers aufgelistet. Von dort aus können neue Vorschläge eingereicht werden. Der „Admin“-Bereich ist für Betreiber vorgesehen. Es können neue Rollen oder Benutzer angelegt werden.

Vorschlag-Bereich

Der Vorschlag-Bereich zeigt alle Details zu einem Vorschlag an. Er ist aufgeteilt in die Unterbereiche „Vorschlag-Details“, „Links“, „Verlauf“ und „Diskussion“. Bei den Details werden alle Metadaten zu einem Vorschlag angezeigt sowie welche Objekte (Vokabulare, Begriffe, Beziehungen) zu einem Vorschlag gehören. Es können Privilegien vergeben werden. Unter „Links“ können Dokumente oder URLs zu einem Vorschlag angehängt werden. Beim „Verlauf“ werden alle Statusänderungen dargestellt. „Diskussion“ stellt die eigentliche Kollaborations-Aufgabe zur Verfügung. Mittels einer Forenansicht können Benutzer untereinander Meinungen austauschen sowie Entscheidungen treffen.

Neue Vorschläge

In diesem Bereich können neue Vorschläge eingereicht werden. Dabei gibt es unterschiedliche Szenarien wie „1 neuer Begriff“, „Begriff ändern“ oder „Begriff umhängen“.

3.5.5 Daten

Die Daten werden in einer relationalen Datenbank gespeichert werden. Der Datenzugriff erfolgt ausschließlich über Webservices.

Dabei beinhaltet die Kollaborationsumgebung folgende Daten:

- Vorschläge
 - Objekte (Schnittstelle zur Datenbank des Terminologieservers)
 - Privilegien (Rechtezuweisung für Benutzer)
 - Status-Änderungen
 - Links
- Diskussionen
 - Links
 - Zitate
 - Objekte (Schnittstelle zur Datenbank des Terminologieservers)
 - Bewertungen
- Benutzer
 - Personenangaben
 - Organisationsangaben
- Rollen
- Status

3.5.6 Leistungen

Das Programm umfasst alle Leistungen, welche unter 3.5.1 genannt sind. Spezielle Leistungen sind nicht erforderlich.

3.5.7 Benutzeroberfläche

Die Benutzeroberfläche wird mithilfe eines Java-Applets realisiert. Dieses wird nach dem Aufruf aus dem Webbrowser auf dem PC des Anwenders ausgeführt. Die Übertragung der Daten vom Server erfolgt per Webservice.

3.5.8 Qualitätsziele

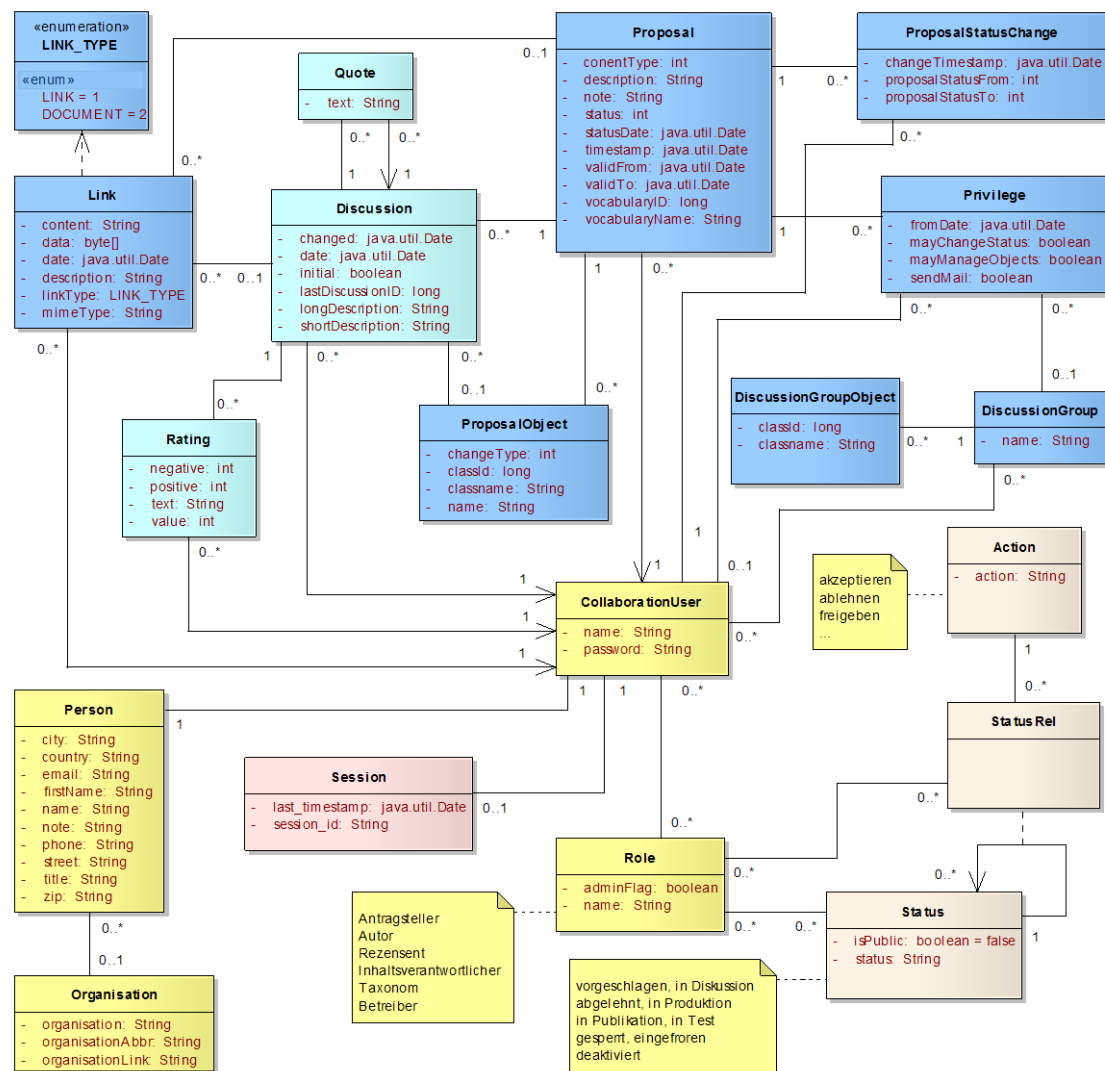
Das Programm soll die Zusammenarbeit von Terminologien vereinfachen. Es ist für den dauerhaften Einsatz gedacht und sollte stabil laufen. Eine hohe Benutzerfreundlichkeit ist obligatorisch.

4 Design

Das Design der Software umfasst die Erarbeitung von Klassenmodellen, Datenbankmodellen und anderen konkreten Modellen. Die meisten Modelle sind auf die gewählte Programmiersprache und die gewählte Datenbank ausgerichtet.

Die Kollaborationsumgebung muss in 2 Bereiche aufgeteilt werden. Zum Einen gibt es das Java-Applet und zum Anderen gibt es die Webservices, welche die Schnittstelle zur Datenbank darstellen. Das Applet verwendet das in Kapitel 4.1 dargestellte Klassenmodell, der Webservice verwendet das in Kapitel 4.2 vorgestellte Datenbankmodell.

4.1 Klassenmodell Fachkonzept

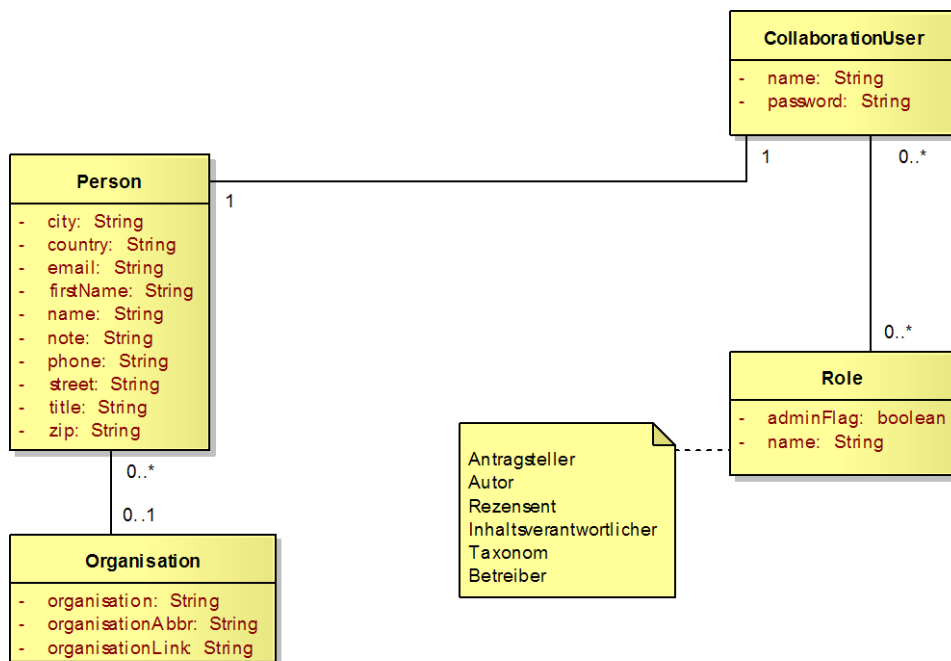


Klassenmodell 4-1 Fachkonzept-Klassenmodell

Das oben abgebildete Modell umfasst die Bereiche „Rollen- und Benutzerkonzept“, „Vorschlag-Konzept“, „Diskussion“, „Status-Verwaltung“ und „Sicherheit“, welche im Folgenden erklärt werden.

4.1.1 Rollen- und Benutzerkonzept

Das Rollen- und Benutzerkonzept umfasst Informationen über Benutzer, welche sich am System anmelden, sowie deren persönliche Informationen und zugehörige Rollen. Die Rollen sind die im Kapitel 3.1 beschriebenen Akteure.



Klassenmodell 4-2 Submodel Rollen- und Benutzerkonzept

Das Klassenmodell 4-2 Submodel Rollen- und Benutzerkonzept zeigt einen Ausschnitt aus dem Gesamtmodell. Die einzelnen Klassen werden im Folgenden näher erläutert:

CollaborationUser	
Beschreibung	Die Klasse beinhaltet Benutzer der Kollaborationsumgebung Die hier hinterlegten Daten werden bei der Anmeldung vom System benötigt.
Beziehungen zu	zu Proposal (N) zu ProposalStatusChange (N) zu Rating (N) zu Discussion (N) zu Link (N)

Attribute	zu Privilege (N)	
	zu Role (N)	
	zu Person (1)	
	zu Session (N)	
	name	Benutzername
	password	Passwort des Benutzers, wird als MD5-Hashwert gespeichert

Person

Beschreibung	„Person“ beinhaltet alle persönlichen Informationen zu einem Benutzer.	
Beziehungen zu	zu CollaborationUser (1)	
	zu Organisation (1)	
Attribute	city	Stadt
	country	Land
	email	Email-Adresse
	firstName	Vorname
	name	Nachname
	note	Notiz
	phone	Telefonnummer
	street	Straße
	title	Titel
	zip	Postleitzahl

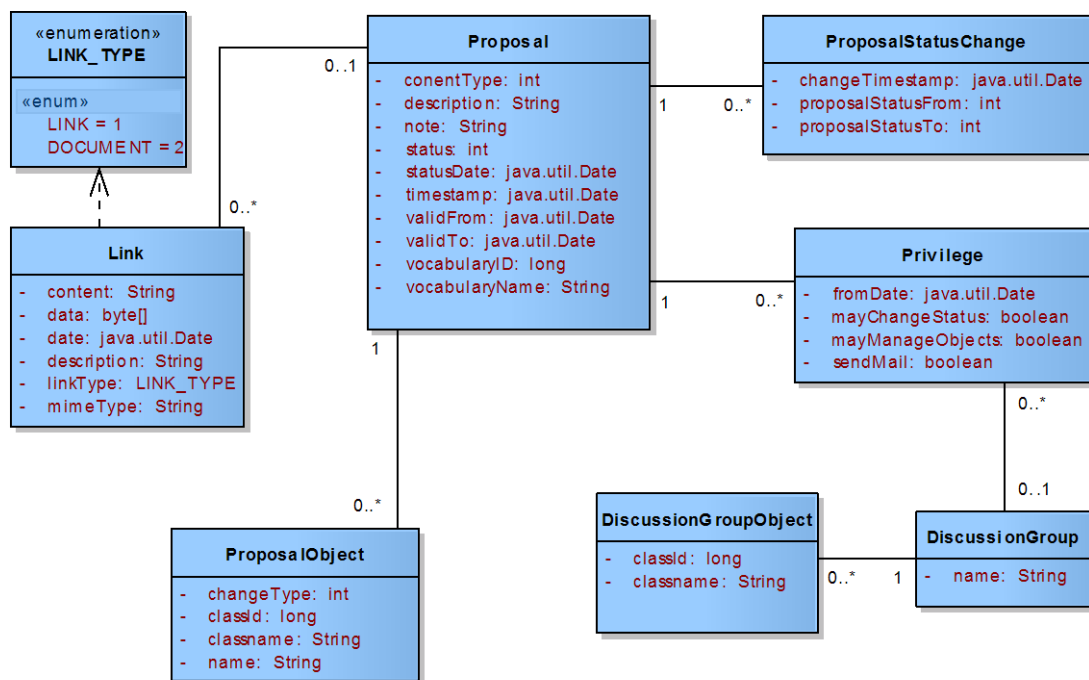
Organisation

Beschreibung	„Organisation“ enthält Informationen über die Firma einer Person.	
Beziehungen zu	zu Person (1)	
	organisation	Name der Firma
Attribute	organisationAbbr	Abkürzung / Kurzversion der Firma
	organisationLink	Homepage der Firma

Role	
Beschreibung	Die Klasse beinhaltet Rollen. Eine Rolle kann mehreren Benutzern zugeordnet werden. Rollen regeln Rechte, sie entscheiden, welche Statusänderungen durchgeführt werden dürfen und welche Status der Benutzer in der Vorschlags-Liste sehen darf. Eine Rolle entspricht einem Akteur, siehe Kapitel 3.1.
Beziehungen zu	zu CollaborationUser (N) zu Status (N) zu StatusRel(N)
Attribute	adminFlag Administrationsflag; entscheidet, ob die Rolle administrative Tätigkeiten durchführen darf oder nicht
	name Name der Rolle

4.1.2 Vorschlag-Konzept

Die Zusammenarbeit verschiedener Benutzer zur Pflege von Terminologien basiert hauptsächlich auf Vorschlägen. Benutzer können Änderungsvorschläge zu bestehenden Terminologien einreichen. Diese können diskutiert werden, es kann über sie abgestimmt werden. Nach dem Ablauf einer Frist muss ein Verantwortlicher eine Entscheidung treffen, ob ein Vorschlag angenommen wird oder abgelehnt wird. Die verknüpften Objekte werden anschließend im Terminologieserver automatisch freigegeben oder gesperrt. Das folgende Modell kann solche Vorschläge verwalten:



Klassenmodell 4-3 Submodel Vorschlag-Konzept

Proposal	
Beschreibung	Ein „Proposal“ ist ein Vorschlag in der Kollaborationsumgebung. Benutzer können Vorschläge geben, welche diskutiert und über welche abgestimmt werden kann. Freigegebene Vorschläge werden anschließend im Terminologieserver veröffentlicht.
Beziehungen zu	zu CollaborationUser (1) zu ProposalObject (N) zu Discussion (N) zu ProposalStatusChange (N) zu Privilege (N) zu Link (N) zu Rating (N)
Attribute	Inhaltstyp des Vorschlags, erlaubte Werte: contentType <ul style="list-style-type: none"> 1 – Begriff 2 – Vokabular 3 – Beziehung description: Kurzbeschreibung des Vorschlags note: Ausführliche Beschreibung

status	Status
statusDate	Datum der letzten Statusänderung
timestamp	Erstelldatum
validFrom	Diskussionszeitraum (Beginn)
validTo	Diskussionszeitraum (Ende)
vocabularyID	die ID des zugehörigen Vokabulars im Terminologieserver
vocabularyName	der Name des zugehörigen Vokabulars

Link

Beschreibung	Ein Link enthält entweder eine Verknüpfung zu einer Webseite oder eine Datei.
Beziehungen zu	zu CollaborationUser (1) zu Proposal (0..1) zu Discussion (0..1)
Attribute	<div>content</div> <div>Inhalt (entweder URL oder Dateiname)</div> <div>data</div> <div>Dateiinhalt</div> <div>date</div> <div>Datum, wann der Link erstellt wurde</div> <div>description</div> <div>Beschreibung</div> <div>linkType</div> <div> Typ des Links, erlaubte Werte: <ul style="list-style-type: none"> 1 – URL 2 – Datei </div> <div>mimeType</div> <div>Definition des Dateiinhalts</div>

ProposalObject

Beschreibung	Diese Klasse stellt eine generische Schnittstelle zum Terminologieserver her. Sie beinhaltet Verknüpfungen zu Objekten im Terminologieserver.
Beziehungen zu	zu Proposal (1) zu Discussion (N)
Attribute	<div>changeType</div> <div> Änderungstyp, erlaubte Werte: <ul style="list-style-type: none"> 1 – Hinzugefügt 2 – Gelöscht </div>

Beschreibung	• 3 – Geändert	
	classId	ID der Klasse „classname“
	classname	Klassenname aus dem Terminologieserver
	name	Bezeichner (intern)
	ProposalStatusChange	
	Diese Klasse stellt den Verlauf von Statusänderungen eines Vorschlags dar.	
	changeTimestamp	Änderungszeitpunkt des Status
Attribute	proposalStatusFrom	vorheriger Status
	proposalStatusTo	neuer Status

Privilege

Beschreibung	Mit Hilfe dieser Klasse können Vorschläge bestimmten Benutzern oder Benutzergruppen zugeordnet werden.	
Beziehungen zu	zu Proposal (1)	
	zu DiscussionGroup (0..1)	
Attribute	fromDate	Datum, ab dem der Benutzer oder die Gruppe privilegiert ist
	mayChangeStatus	vorbehalten
	mayManageObjects	vorbehalten
	sendMail	entscheidet, ob bei neuen Diskussionseinträgen eine Email an den Benutzer geschickt wird

DiscussionGroup

Beschreibung	Verschiedene Benutzer können zu einer Diskussionsgruppe zusammengefasst werden. Somit ist es einfacher, verschiedene Gruppen (z.B. Firmen) zu verwalten.	
Beziehungen zu	zu Privilege (N)	
	zu DiscussionGroupObject (N)	
	zu CollaborationUser (N)	
Attribute	name	Name der Diskussionsgruppe

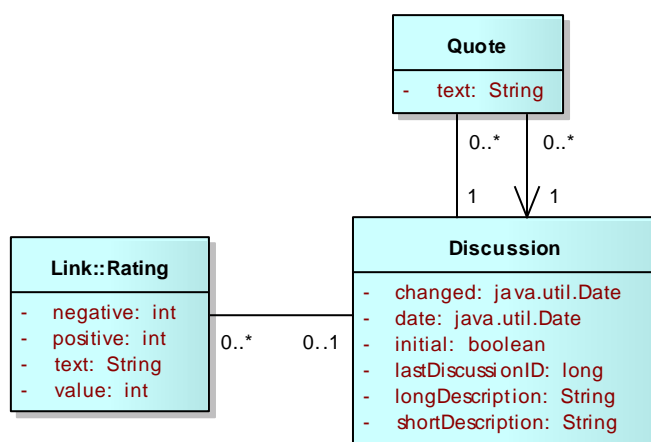
DiscussionGroupObject

Beschreibung	Diese Klasse stellt eine generische Schnittstelle zum Terminologieserver her. Wenn ein neuer Vorschlag erstellt wird, und das zugehörige Objekt (z.B. Vokabular) hier angegeben ist, wird diese Gruppe automatisch mit dem Vorschlag verknüpft (über „Privilege“)	
Beziehungen zu	zu DiscussionGroup (1)	
Attribute	classId	ID der Klasse „classname“
	classname	Klassenname aus dem Terminologieserver

4.1.3 Diskussion

Der folgende Abschnitt erklärt den Teilbereich „Diskussion“. Jeder Vorschlag der Kollaborationsumgebung besitzt eine Diskussion. Sofern sich dieser in dem entsprechenden Status befinden, können sich Benutzer an einer Diskussion beteiligen.

Eine Diskussion ist ähnlich einem Forum aufgebaut. Ein Eintrag wird in der Klasse „Discussion“ festgehalten.



Klassenmodell 4-4 Submodel Diskussion

Discussion

Beschreibung	Die Klasse Discussion ist einer Antwort in einem Forum gleichzusetzen. Einzelne Einträge werden hier gespeichert und mit einem Vorschlag und einem Benutzer verknüpft. Vergleicht man dieses mit einem Forum, ist das Thema eines Forums hier dem Vorschlag
--------------	---

Beziehungen zu	gleichzusetzen.	
	zu CollaborationUser (1)	
	zu Proposal (1)	
	zu Link (N)	
	zu ProposalObject (0..1)	
	zu Quote (N)	
Attribute	zu Rating (N)	
	changed	Datum, an dem der Eintrag zuletzt bearbeitet wurde, oder null
	date	Erstell-Datum des Eintrags
	initial	legt fest, ob dieses der erste Eintrag eines Vorschlags ist bzw. ein neues Unterthema eröffnet
	lastDiscussionID	wenn angegeben, ist dieser Eintrag eine Antwort auf einen bestehenden Eintrag mit der angegebenen Identifikationsnummer
	longDescription	der Text
	shortDescription	zur Zeit nicht benutzt

Quote

Beschreibung	Mit Hilfe der Klasse „Quote“ können Zitate zu Beiträgen erstellt werden.	
Beziehungen zu	zu Discussion (N)	
Attribute	text	die zitierte Textstelle (oder leer, wenn der ganze Beitrag zitiert wird)

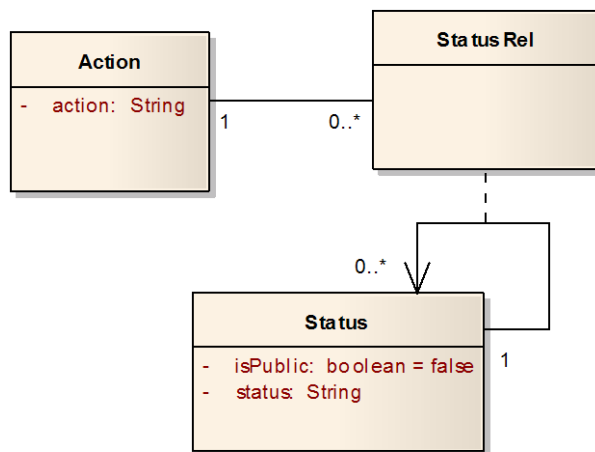
Rating

Beschreibung	„Rating“ (Bewertung) ist entweder mit einer Diskussion oder einem Vorschlag verknüpft. Bei der Verknüpfung mit einer Diskussion wird diese bewertet, bei der Verknüpfung mit einem Vorschlag wird über diesen abgestimmt.	
Beziehungen zu	zu Discussion (0..1)	
	zu Proposal (0..1)	
	zu CollaborationUser (1)	
Attribute	negative	vorbehalten
	positive	vorbehalten

text	optionalen Text, wird bei der Bewertung eines Vorschlags benutzt
value	<p>Bewertung, unterschiedliche Werte, je nachdem womit die Klasse verknüpft wurde</p> <p><i>Diskussion</i></p> <p>Werte von 1-6 (Schulnoten) sind möglich</p> <p><i>Vorschlag</i></p> <p>mögliche Werte:</p> <ul style="list-style-type: none"> • 1 – Ablehnung • 2 – Zustimmung • 3 – Zustimmung mit Modifikation

4.1.4 Status-Verwaltung

Die Status-Verwaltung realisiert das in Kapitel 3.4 beschriebene Verhalten der Status und der Statusänderungen. Es wird festgelegt, welcher Status zu welchem Status mit welcher Aktion wechseln kann. Dieser Vorgang ist somit nicht fest im Programm codiert, sondern per Datenbank parametrierbar. So kann der Anwender das Verhalten der Anwendung zu einem späteren Zeitpunkt leicht ändern.



Klassenmodell 4-5 Submodel Status-Verwaltung

Status

Beschreibung	Beschreibung der möglichen Status, welche im Kapitel 3.4.1 beschrieben sind.	
Beziehungen zu	zu Role (N) zu StatusRel (N)	
Attribute	isPublic	legt fest, ob der Status im Terminologieserver als öffentlich gilt oder nicht
	status	Name des Status

StatusRel

Beschreibung	Dient als Verknüpfung zwischen 2 Status Sie legt das Verhalten der Statusänderungen fest.	
Beziehungen zu	zu Status (N) zu Role (N) zu Action (1)	

Action

Beschreibung	Beschreibt mögliche Aktionen beschrieben in Kapitel 3.4.2, welche der Benutzer ausführen kann.	
Beziehungen zu	zu StatusRel (N)	
Attribute	action	Name der Aktion

4.1.5 Sicherheit

Alle Webservices der Kollaborationsanwendung sind theoretisch für jeden öffentlich. Jedoch sind diese nur nutzbar, wenn sich ein Benutzer angemeldet hat. Nach der erfolgreichen Anmeldung erhält der Benutzer bzw. die Anwendung eine sogenannte Session-ID. Diese Nummer wird nun bei jedem Aufruf an die Kollaborationsanwendung übergeben. Das System prüft diese Nummer und führt den Dienst aus oder nicht.

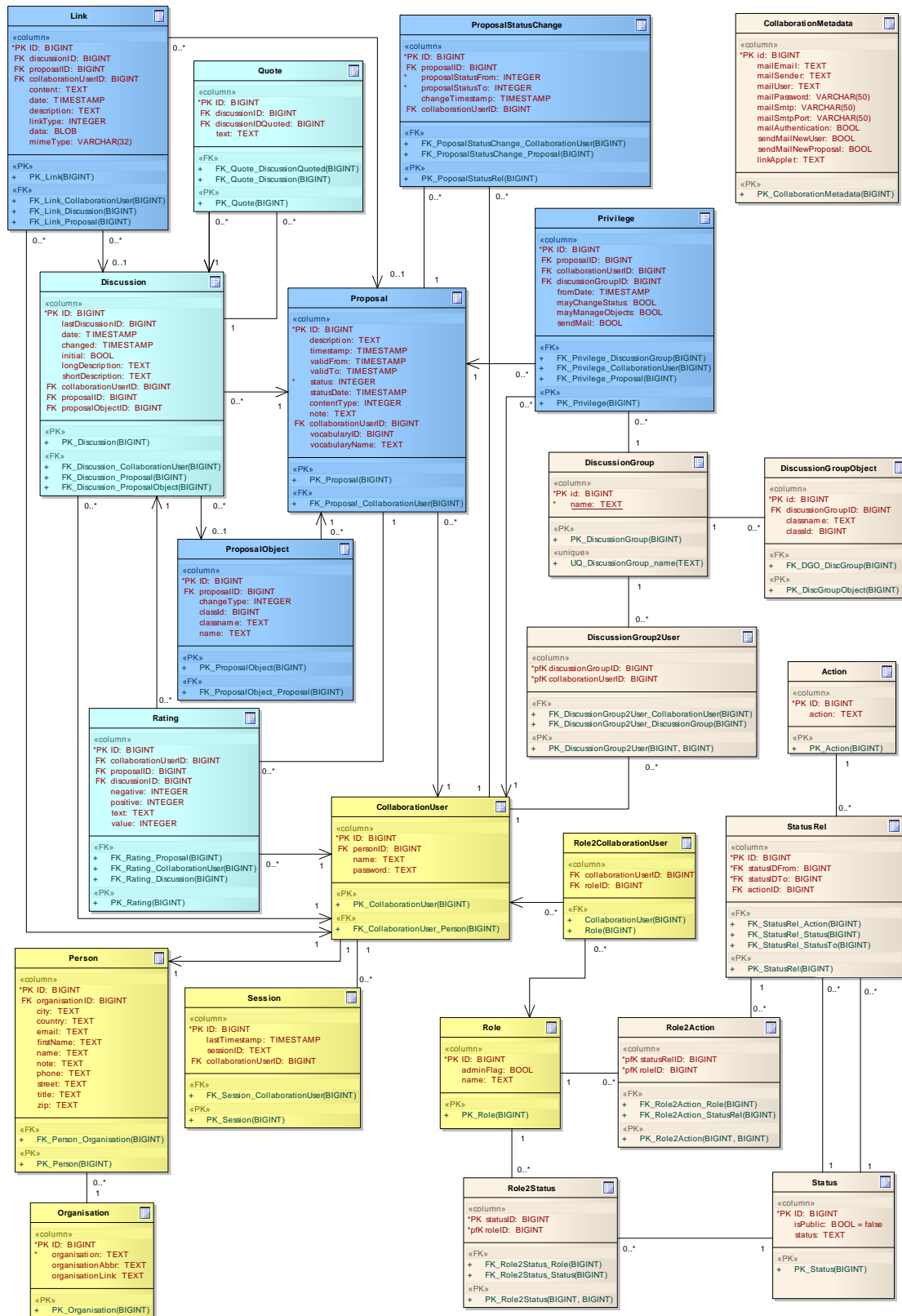
Das folgende Klassenmodell der Sicherheit besteht deswegen nur aus einer Klasse, welche die Aufgabe der Verwaltung dieser Nummern hat:



Klassenmodell 4-6 Submodel Sicherheit

Session	
Beschreibung	Diese Klasse beinhaltet temporäre Informationen über die Anmeldung eines Benutzers in der Kollaborationsumgebung
Beziehungen zu	zu CollaborationUser (1)
Attribute	last_timestamp Zeitstempel der letzten Aktivität eines Benutzers
	session_id eindeutige UUID, welche nach der Anmeldung vergeben wird und bis zur Abmeldung gültig ist

4.2 Datenbankmodell

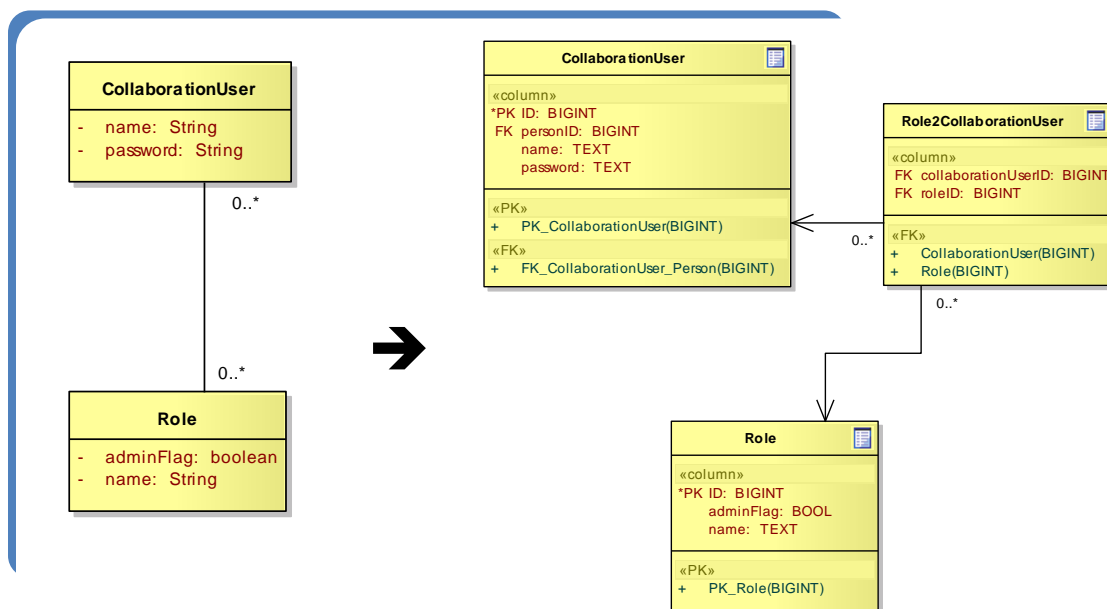


Datenbankmodell 4-1 Kollaborations-Datenbankmodell

Das Datenbankmodell beschreibt die Art und Weise, wie Daten in der Datenbank gespeichert werden. Das Datenbankmodell der Kollaborationsumgebung beruht auf dem relationalen Datenmodell.

Das Datenbankmodell wird aus dem Klassenmodell abgeleitet. Mit Hilfe des Enterprise Architects lässt sich das Grundmodell der Datenbank aus dem Klassenmodell automatisch generieren. Es bedarf anschließend jedoch einiger Anpassungen. Zunächst müssen Tabellen mit Identifikationsnummern (IDs) versehen werden. Aus M:N-Verbindungen muss eine 3. Klasse (hier: Tabelle) erstellt werden, damit die Verbindung in der relationalen Datenbank dargestellt werden kann (siehe dazu Abbildung 4-1 Beispiel Umwandlung M:N-Tabellen). Bei 1:N-Verbindungen müssen Fremdschlüssel in die Tabellen eingefügt werden. Anschließend müssen alle Datentypen in die Typen der Datenbank umgewandelt werden, da das Klassenmodell Java-Typen beinhaltet. Die Datenbanktypen müssen der gewählten Datenbank entsprechen. In diesem Fall ist das die Datenbank MySQL.

Eine Erläuterung der einzelnen Tabellen macht an dieser Stelle wenig Sinn, da alle Attribute bereits im Kapitel 4.1 „Klassenmodell Fachkonzept“ erläutert sind.



Aus den zwei Klassen entstehen im Datenbankmodell drei Tabellen.

Abbildung 4-1 Beispiel Umwandlung M:N-Tabellen

4.3 Webservice

Der Webservice der Kollaborationsumgebung stellt die Schnittstelle vom Applet zur Datenbank dar. Das Java-Applet wird bei der Ausführung auf dem PC des Anwenders ausgeführt, so dass dieser keine direkte Verbindung zur Datenbank hat. Um die Datenbank gegen Angriffe nach außen zu schützen, sollte diese nicht nach außen geöffnet werden. Deshalb empfiehlt sich die Verwendung eines Webservices.

Die Bereitstellung der Daten erfolgt in mehreren Diensten. Alle Dienste, bis auf den Dienst „Login“, sind geschützt und nur mit einer Session-ID erreichbar. Diese ID erhält der Benutzer bzw. das Applet nach dem erfolgreichen Einloggen.

Alle Dienste werden im Folgenden kurz erläutert.

4.3.1 Dienste

Alle Dienste erwarten bestimmte Parameter. Auf diese wird in folgender Tabelle aus Gründen der besseren Übersichtlichkeit nicht eingegangen. Alle Parameter sind jedoch in der Dienstmatrix nachzulesen.

Dienst	Beschreibung
CreateDiscussion	Erstellt eine neue Diskussion (Beitrag)
CreateProposal	Erstellt einen neuen Vorschlag
CreateRole	Erstellt eine neue Rolle
CreateUser	Erstellt einen neuen Benutzer
GetProposalDetails	Alle Details eines Vorschlags erhalten (Links, ProposalObjects, StatusChange, Privileges)
ListActions	Listet alle Aktionen
ListDiscussions	Listet alle Diskussionen
ListProposals	Listet alle Vorschläge
ListRoles	Listet alle Rollen
ListStatus	Listet alle Status
ListUser	Listet alle Benutzer
Login	Meldet einen Benutzer an das System an. Dieser erhält bei Erfolg Benutzerinformationen und eine Session-ID, mit der er andere Dienste aufrufen kann.
Logout	Meldet einen Benutzer vom System ab
MaintainDiscussion	Bearbeitet einen Diskussionseintrag
MaintainProposal	Bearbeitet einen Vorschlag
MaintainRole	Bearbeitet eine Rolle
MaintainUser	Bearbeitet einen Benutzer
UpdateProposalStatus	Aktualisiert den Status eines Vorschlags. Fügt zudem einen Eintrag in der Tabelle „ProposalStatusChange“ hinzu, damit der Status-Verlauf nachvollziehbar bleibt.

4.3.2 Dienstmatrix

Zur Darstellung der Dienste und ihrer erforderlichen Übergabe-Parameter sowie Rückgabewerte empfiehlt sich die Darstellung in Form einer Matrix. Auf der Horizontalen werden alle verfügbaren Dienste, auf der Vertikalen werden die Übergabe- und Rückgabeparameter dargestellt.

Die komplette Matrix passt aufgrund ihrer Komplexität nicht lesbar auf diese Seiten. Sie finden diese im Anhang. Daher ist hier nur ein Ausschnitt der Matrix zu sehen:

Terminologieserver - Kollaboration																		
XML-Node	Typ	Format Element / Attribut	Login (Request)	(Response)	Login (Request)	(Response)	ListProposals (Request)	(Response)	GetProposalDetails (Request)	(Response)	CreateProposal (Request)	(Response)	MaintainProposal (Request)	(Response)	UpdateProposalStatus (Request)	(Response)	ListActions (Request)	(Response)
ReturnInfos																		
ReturnType	ReturnType	E		1		1		1		1		1		1		1		1
overallErrorCategory	Enum	A		1		1		1		1		1		1		1		1
status	Enum	A		1		1		1		1		1		1		1		1
message	String	A		1		1		1		1		1		1		1		1
Login																		
LoginType	LoginType	E	1	1	1		1		1		1		1		1		1	
sessionID	String	A		0..1	1		1		1		1		1		1		1	
username	String	A	1															
password	String (MD5)	A	1															
Role																		
RoleType	List< RoleType>	E		0..n														
id	long	A		1														
name	String	A		1														
adminFlag	boolean	A		1														
Action																		
ActionType	List< ActionType>	E													0..1			0..n
id	long	A													1			1
action	String	A																1
User																		
CollaborationUserType	List< CJType>	E		0..1														
id	long	A		1														
name	String	A		1														
password	String (MD5)	A																
PersonType	PersonType	E		1														
id	long	A		1														
city	String	A		1														
country	String	A		1														
email	String	A		1														
firstName	String	A		1														
name	String	A		1														
note	String	A		1														
phone	String	A		1														
street	String	A		1														
zip	String	A		1														
OrganisationType	OrganisationType	E		0..1														
id	long	A		1														
organisation	String	A		1														
organisationAbbr	String	A		1														
organisationLink	String	A		1														
UserInfo																		
UserInfoType	UserInfoType	E																
id	long	A																
name	String	A																
organisation	String	A																
username	String	A																
Proposal																		
ProposalType	List< ProposalType>	E					0..n	1	0..1	1	1	1	1		1			
id	long	A					1	1	1			1	1		1			
description	String	A					1			1				0..1				
timestamp	java.util.Date	A					1		1									
validFrom	java.util.Date	A					1		1	0..1			0..1					
validTo	java.util.Date	A					1		1	0..1			0..1					

Abbildung 4-2 Dienstmatrix (Auszug)

5 Implementierung

Die Implementierung ist das Erstellen der Software aus den vorher erstellten Modellen. In diesem Projekt müssen die Datenbank, das Java-Applet und der Webservice für die Kollaborationsumgebung inklusive aller Dienste erstellt werden. Zudem muss am Ende der Linux-Server konfiguriert werden.

5.1 Datenhaltung

Alle Daten der Kollaborationsumgebung werden, wie im Kapitel 4.2 beschrieben, in der relationalen Datenbank MySQL abgelegt. Relationale Datenbanken legen Daten in verschiedenen Orten (Tabellen) ab. Verknüpfungen werden dabei über sog. Relationen hergestellt. Bei Daten über eine Person werden z.B. der Name und Vorname in einer Tabelle, der Ort in einer anderen Tabelle gespeichert. Das hat den Vorteil, dass bei mehreren Personen, welche im gleichen Ort wohnen, der Ort auch nur 1x abgespeichert wird. Eine Änderung der Postleitzahlen der Orte müsste dann z.B. nur an einer Stelle erfolgen. Datenbanken sind so pflegeleichter und benötigen weniger Speicherplatz. Zudem wird die Wartbarkeit erhöht.

In Frage würde auch eine Objektrelationale Datenbank kommen. Aufgrund der wenigen Erfahrungen und der kleinen Bekanntheit fiel die Entscheidung in diesem Projekt jedoch auf eine relationale Datenbank, welche sich in der Praxis seit langem bewährt.

Der Zugriff auf die Datenbank erfolgt mittels der Datenbanksprache SQL (Structured Query Language). Sie dient zur Definition, Abfrage und Manipulation von Daten in relationalen Datenbanken. Durch die standardisierte Sprache muss der Anwender nicht die genauen Datenstrukturen kennen. Wie die Datenbank ihre Daten abspeichert, ist für den Programmierer uninteressant, da dieser seine Daten per SQL bezieht. Somit könnte sogar die Datenbank ausgetauscht werden. Leider unterstützen nicht alle Datenbanken alle SQL-Befehle, so dass in diesem Fall kleinere Anpassungen am Quelltext nötig wären.

5.1.1 Ableitung des Datenbankentwurfs

Aus dem bereits erzeugten Datenbankmodell in Kapitel 4.2 wird mithilfe des Enterprise Architects ein Skript generiert. Mit Hilfe des Skripts lassen sich alle Tabellen und deren Relationen in eine MySQL-Datenbank einfügen. Ein manuelles Einfügen der Strukturen ist somit nicht mehr nötig.

5.1.2 Performance-Überlegungen

Die Kollaborationsumgebung speichert Daten in den Bereichen Benutzerverwaltung, Vorschlägen, Diskussionen und Status. Beim DIMDI gehen jährlich ca. 400 Änderungsvorschläge für den ICD ein (Thun, 2009). Wenn bei jedem Vorschlag ca. 20 Personen an der Entwicklung beteiligt sind und jeder davon ca. 5 Diskussionsbeiträge pro Vorschlag gibt, kommt es insgesamt zu folgender Anzahl an Einträgen in der Datenbank:

Tabelle	Anzahl Einträge
CollaborationUser	20
Person	20
Organisation	20
Proposal	400
Privilege	8000
ProposalObject	800
ProposalStatusChange	1600
Discussion	40000
Rating	1000
Link	1500
Quote	10000
Gesamt	63360

Tabelle 5-1 geschätztes Datenbankvolumen

Das oben geschätzte Datenbankvolumen von ca. 64.000 Einträgen pro Jahr und Ordnungssystem ist unkritisch für heutige Datenbanktechnologien.

Eine MySQL-Datenbank in der Version 5.1 kann auf einem Linux-Server Tabellen mit maximal 4TB Daten fassen, diese Einschränkung sollte kein Problem sein. MySQL speichert Daten mit einem Index in Baumstrukturen. Auf diese kann mit logarithmischer Komplexität zugegriffen werden. Für eine Tabelle mit n Datensätzen sind $\log(b,n)$ Zugriffe notwendig, bis ein gesuchter Datensatz gefunden ist. Die Praxis zeigt, dass die Basis b in der Regel zwischen 20 und 40 liegt.

Damit wären zum Finden eines Datensatzes aus 40.000 Datensätzen (größte Tabelle *Discussion*) maximal 4 Vergleiche und Plattenzugriffe notwendig. Zum Vergleich, wenn es z.B. 10.000.000 Datensätze sind, gibt es maximal 6 Vergleiche und Plattenzugriffe.

MySQL kommt daher im Rahmen der Begrenzungen des Betriebssystems problemlos mit großen Tabellen klar.

Die andere Seite der Performance-Überlegungen muss an den Webservice gerichtet sein, welcher die Schnittstelle zwischen dem Applet und der Datenbank realisiert. Der gewählte Server sollte eine ausreichende Internetgeschwindigkeit (Bandbreite) besitzen, um den Zugriff von mehreren Benutzern gewährleisten zu können. Im Rahmen der Masterarbeit konnte noch kein Test mit mehreren Benutzern gleichzeitig durchgeführt werden. Ein Test mit 1-2 Benutzern ist problemlos durchgeführt worden.

5.2 Webservice Kollaborationsumgebung

Der Webservice ist eine für den Benutzer nicht sichtbare Anwendung. Die Funktionsweise des Webservices wurde in Kapitel 4.3 bereits beschrieben und designet.

Bei der Implementierung kommt das sehr leistungsfähige und mächtige Tool Netbeans zum Einsatz. Die Wahl fiel auf Netbeans, weil die Generierung von Webservices sehr gut unterstützt wird.

5.2.1 Projekt und Webservice erstellen

Als erster Schritt wird ein neues (Web-)Projekt erstellt. In diesem wird ein neuer Webservice „Kollaboration“ erstellt.

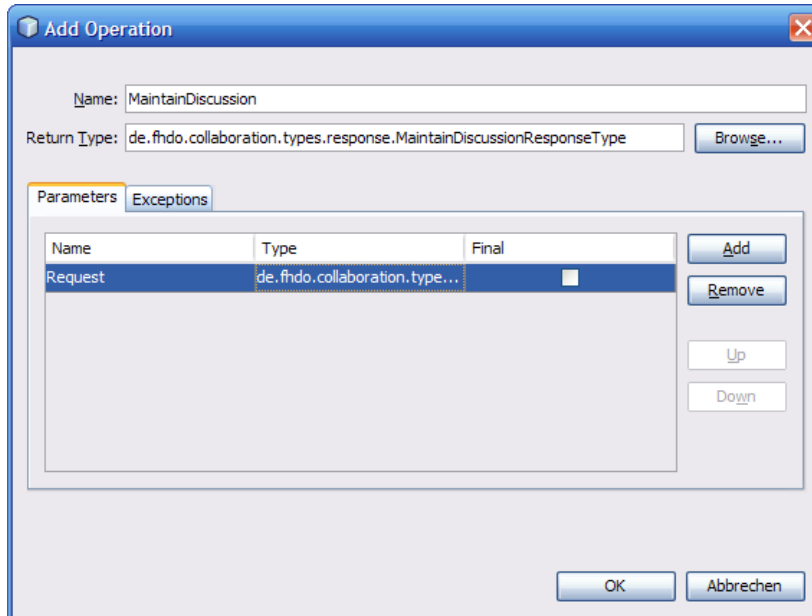
5.2.2 Typen erstellen

Die Dienstematrix aus Kapitel 4.3.2 enthält Parameter, welche an die Dienste übergeben werden müssen und Parameter, welche von den Diensten zurück kommen (Rückgabewerte). Die Parameter (z.B. Benutzer) werden zu Typen zusammengefasst (z.B. UserType). Für jeden Dienst werden anschließend Typen erstellt (je 1 für Request und 1 für Response), welche aus den vorher erstellten Typen bestehen. Dabei können die Typen einfach oder als Liste angegeben werden, je nach Beziehung. Der Dienst „ListProposals“ beinhaltet z.B. eine Liste an Vorschlägen, da hier alle Vorschläge zurückgegeben werden.

Ein Typ wird durch eine Java-Klasse realisiert.

5.2.3 Dienste erstellen

Für jeden Dienst in der Dienstematrix muss nun eine neue Methode erstellt werden. Jeder Methode werden nun die Typen zugewiesen, wie auf folgendem Bild zu sehen ist.



Bildschirmabzug 5-1 Dienst hinzufügen

Dieser Vorgang wird nun für jeden Dienst durchgeführt, so dass alle Dienste erstellt werden. Veröffentlicht man das Projekt zu diesem Zeitpunkt, würde der Webservice von außen ansprechbar sein. Die innere Programmlogik fehlt jedoch noch gänzlich, so dass keine Informationen zurückgegeben würden.

5.2.4 Datenbankverbindung

Die Klassen für die Datenbankverbindung werden in einem neuen Paket „de.fhdo.collaboration.db“ abgespeichert. Die einzelnen Klassen realisieren die Datenbankverbindung sowie die Speicherung und das Auslesen der Daten. Die Parameter der Datenbank werden in einer Konfigurationsdatei festgelegt, so dass beim Umzug auf einen neuen Server oder bei der Vergabe eines neuen Datenbankpassworts der Quelltext nicht neu übersetzt werden muss.

5.2.5 Programmlogik

Alle Dienste müssen nun mit Leben gefüllt werden. Die Dienste laufen in der Regel nach dem gleichen Aufbau ab. Zunächst wird die Datenbankverbindung hergestellt. Es folgt die Prüfung nach Authentizität, also ob der Benutzer am System angemeldet ist.

Anschließend wird die Aufgabe des Dienstes implementiert, welche hauptsächlich aus „Daten auflisten“, „Daten eingeben“ und „Daten ändern“ besteht.

5.2.6 Sicherheit

Die Sicherheit der Kommunikation lässt sich prinzipiell auf Transport- oder der Nachrichtenebene angehen. Sicherheit auf der Transportebene bedeutet, dass Funktionen des Transportprotokolls verwendet werden, um die Übertragung zu schützen. Grenzen hat diese Ebene der Sicherheit, wenn die Kommunikation über zwei Punkte hinausgeht, wenn also z.B. der Webservice einen weiteren Webservice aufruft und beide Webservices nur einen Teil der Daten „sehen“ dürfen. Dann ist zwar der Transportweg geschützt, die Daten sind aber in beiden Services komplett sichtbar.

Das bekannteste Beispiel für die Sicherheit der Transportebene ist SSL, welches im Zusammenspiel mit http häufig zum Einsatz kommt. Da bei der Kollaborationsumgebung das oben genannte Beispiel nicht zum Tragen kommt, ist dieses Verfahren für die Gewährleistung der Sicherheit zunächst einsetzbar.

Die Sicherung der Nachrichtenebene ist durch den Einsatz von WS-Security möglich. Für den Transport kann beim Einsatz von WS-Security auch ein unsicheres Übertragungsprotokoll zum Einsatz kommen, da die Daten auf Nachrichtenebene verschlüsselt werden. Mithilfe der Spezifikation von WS-Security lassen sich Nachrichten also verschlüsseln und signieren.

Laut dem Pflichtenheft wird keine der beiden Sicherheitstechniken in dieser Arbeit verwendet. Der Einsatz einer der beiden oder sogar beider Sicherheitstechniken ist nachträglich jedoch möglich und ratsam.

5.3 Java-Applet

Das Java-Applet ist das Programm, welches der Benutzer aus seinem Browser aufruft. Mittels der graphischen Oberfläche kann er mit der Kollaborationsumgebung interagieren. Der Programmablauf, wie größtenteils in Kapitel 3.3 Anwendungsfunktionen und Navigation abgebildet, ist von diesem Applet realisiert.

Auf eine detaillierte Beschreibung der einzelnen Bildschirmmasken wird an dieser Stelle nicht eingegangen. Diese werden im Kapitel 6.2 in Form eines Probelaufs dargestellt.

Auch für das Java-Applet kommt das Entwicklungswerkzeug Netbeans in der Version 6.7.1 zum Einsatz. Mittels des integrierten GUI-Editors werden die Masken für die grafische Oberfläche umgesetzt.

Dem Applet muss zusätzlich eine Signatur hinzugefügt werden, welche der Benutzer beim Start des Programms einmalig akzeptieren muss. Ohne diese Signatur wäre dem Applet kein Zugriff auf die Webservices möglich.

6 Test und Testergebnisse

Das folgende Kapitel zeigt die Testumgebung, den Probelauf der Kollaborationsumgebung sowie dessen Ergebnisse.

6.1 Testumgebung

Die Testumgebung wird auf dem öffentlichen Server der FH Dortmund installiert. Dieser ist zu erreichen unter der Adresse <http://193.25.22.68/> (oder auch <http://www.term.mi.fh-dortmund.de>).

Das Applet der Kollaborationsumgebung läuft unter der Adresse <http://www.term.mi.fh-dortmund.de/kollaboration/index.html>.

Der Terminologie-Webservice läuft unter der Adresse <http://193.25.22.68:8080/Terminologieserver/>, der Kollaboration-Webservice läuft unter der Adresse <http://193.25.22.68:8080/Terminologieserver/Kollaboration?wsdl>.

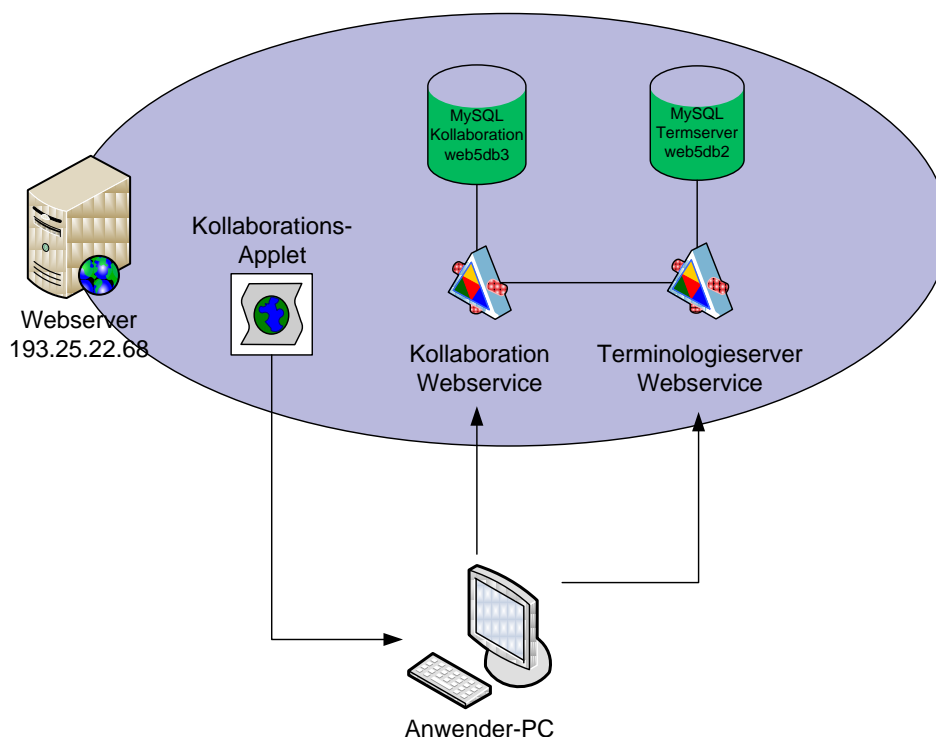


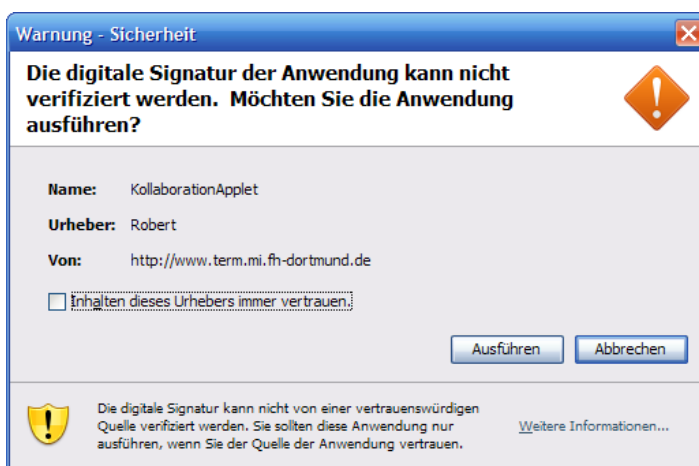
Abbildung 6-1 Testumgebung Überblick

Das Applet befindet sich zwar auf dem Server, wird jedoch beim Start auf den Anwender-PC geladen und von dort ausgeführt. Das Applet greift vom Anwender-PC auf die Webservices des Terminologieservers und der Kollaboration zu.

6.2 Probelauf

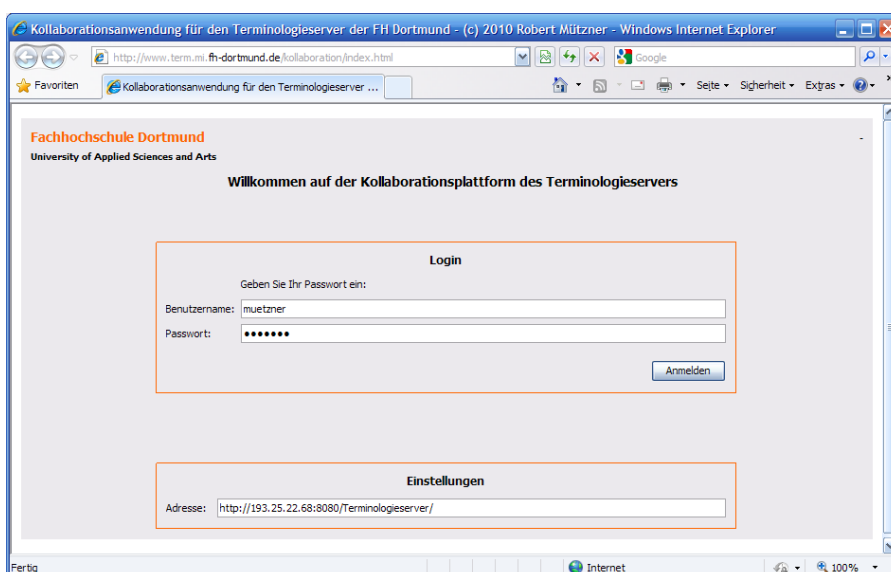
Der Probelauf demonstriert den erfolgreichen Einsatz der Software auf dem Server der FH Dortmund. Der Betreiber der Kollaborationsumgebung legt zunächst einen neuen Benutzer an, welcher einen neuen Vorschlag einreicht. 2 weitere Benutzer diskutieren nun über diesen Vorschlag und nehmen an der Abstimmung teil. Der Inhaltsverwalter schaltet bei dem Vorgang die Status frei, bis der Vorschlag letztendlich veröffentlicht wird.

Nachdem das Applet gestartet wurde, muss zunächst die digitale Signatur akzeptiert werden:



Bildschirmabzug 6-1 Probelauf: Applet-Signatur

Ohne diese Signatur könnte das Applet nicht auf die Webservices zugreifen.



Bildschirmabzug 6-2 Probelauf: Anmeldung

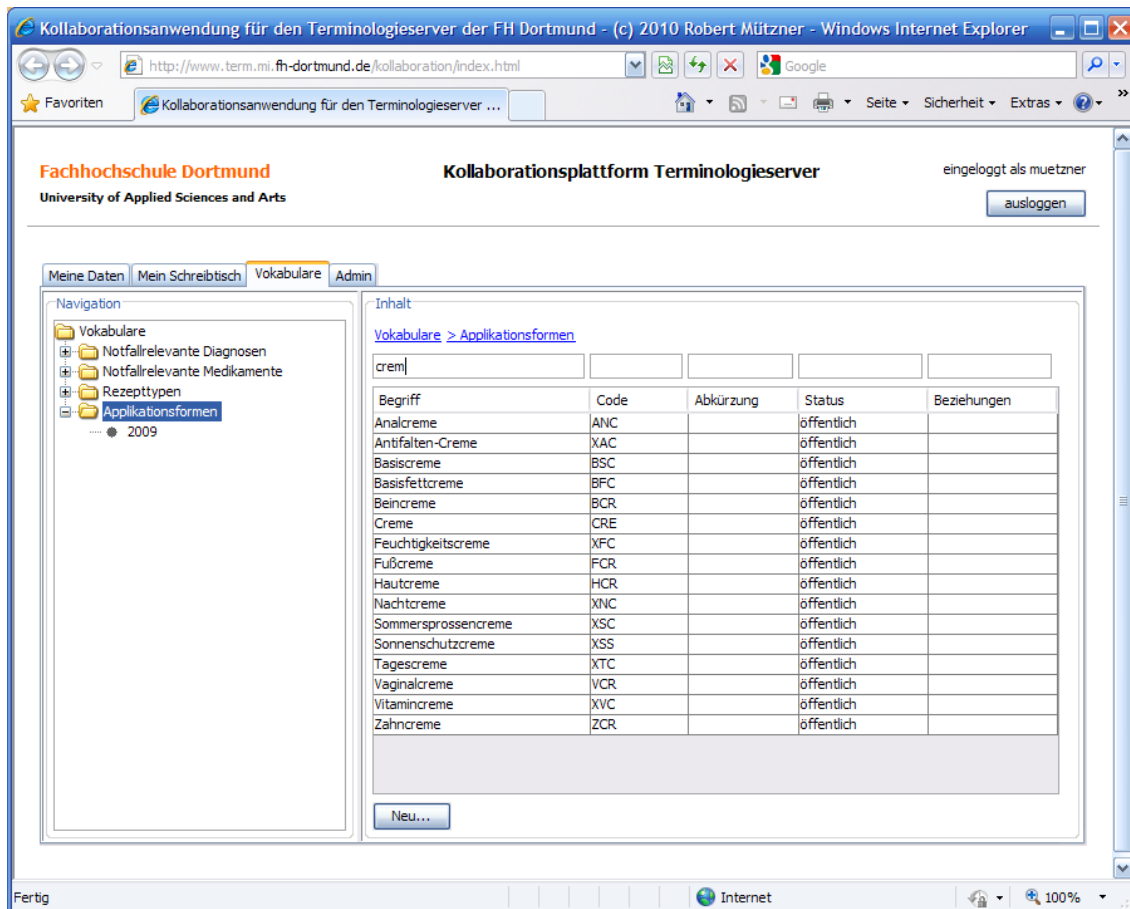
Der Anmeldungs Bildschirm ist der Startpunkt des Applets. Ohne einen gültigen Benutzernamen und dem zugehörigen Passwort ist eine Anmeldung nicht möglich.

Als erster Benutzer wählt sich „muetzner“ ein, welcher Administrator, Betreiber und Inhaltsverwalter der Plattform ist.

Der folgende Bildschirm zeigt den Reiter „Meine Daten“. Dort kann der Benutzer seine persönlichen Daten vervollständigen oder sein Passwort ändern.

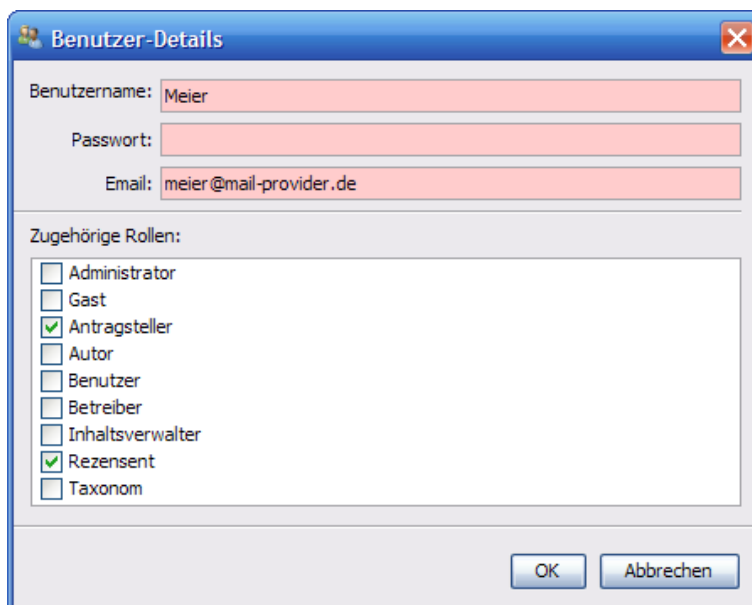
Bildschirmabzug 6-3 Probelauf: Meine Daten

Weiter geht es zu der Vokabularansicht, eine Übersicht von allen Vokabularen. Die Vokabulare selbst sind links inklusive ihren Versionen aufgelistet. Im rechten Teil werden die Begriffe zu einem Vokabular angezeigt. Über die Filterfelder über jeder Spalte kann die Liste nach Belieben gefiltert werden. Von diesem Bildschirm aus können neue Vorschläge eingereicht werden. Zur Zeit ist es möglich, neue Begriffe für ein Vokabular über den Button „Neu...“ unter der Begriffsliste zu erstellen.



Bildschirmabzug 6-4 Probelauf: Vokabulare

Im Administrations-Bereich wird nun ein neuer Benutzer „Meier“ hinzugefügt. Herr Meier besitzt die Rollen „Antragsteller“ und „Rezensent“.



Bildschirmabzug 6-5 Probelauf: Neuer Benutzer

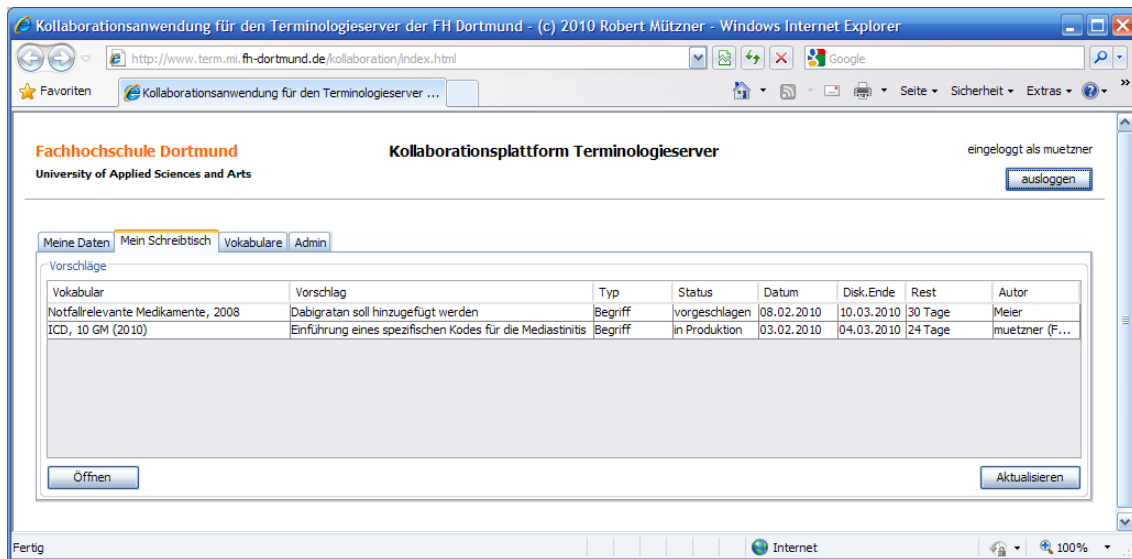
Der neue Benutzer „Meier“ meldet sich nun am System an, wechselt in die Vokabular-Übersicht und reicht einen neuen Vorschlag ein:

The screenshot shows a Windows-style dialog box titled "Neuer Vorschlag" (New Suggestion). It contains the following fields and controls:

- Aktuelles Vokabular:** Notfallrelevante Medikamente, 2008
- Diskussionszeitraum:** 08.02.2010 - 10.03.2010 (with calendar icons for date selection)
- Vorschlag:** Dabigratan soll hinzugefügt werden (text area)
- Begriff:** Dabigratan (text field)
- Abkürzung:** Dbgrtn (text field)
- Code:** (empty text field)
- ☒ Vorzugsbegriff (isPreferred)
- ☐ Achse
- Buttons:** OK and Abbrechen

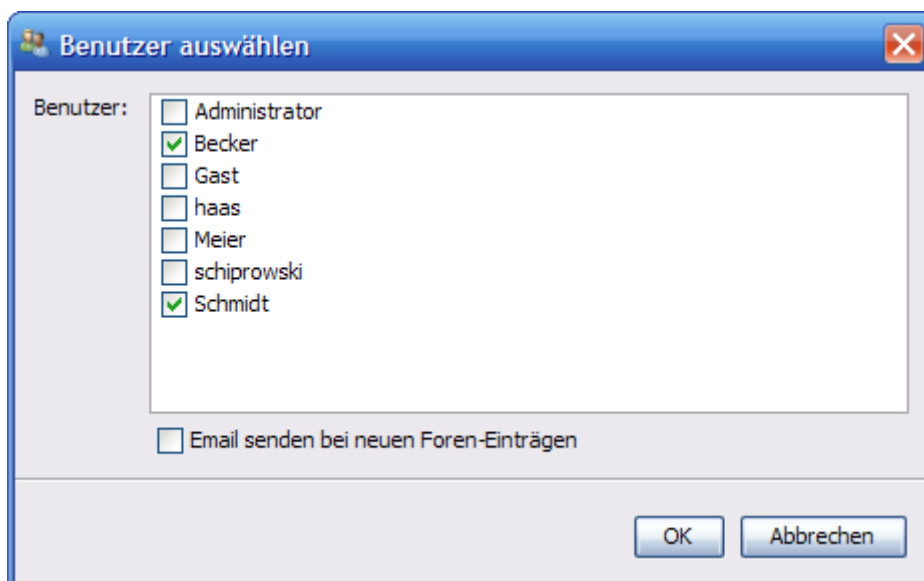
Bildschirmabzug 6-6 Probelauf: Neuer Vorschlag

Nun schaut der Benutzer „muetzner“ auf die Ansicht „Mein Schreibtisch“ und entdeckt den neuen Vorschlag „Dabigratan soll hinzugefügt werden“. Diesen Eintrag sieht er, da er Inhaltsverwalter des entsprechenden Vokabulars ist.



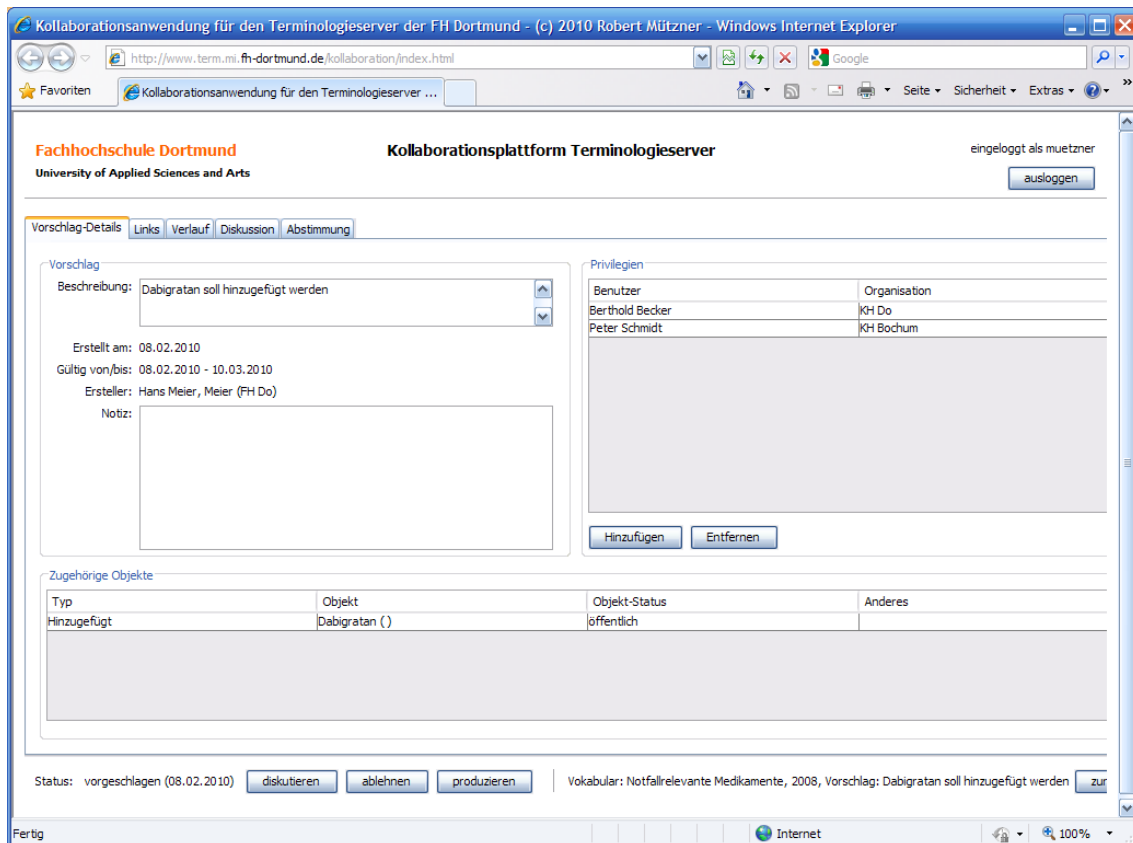
Bildschirmabzug 6-7 Probelauf: Mein Schreibtisch

Als Inhaltsverwalter fügt er nun 2 Benutzer als Rezensenten dem Vorschlag hinzu, welche über diesen mitdiskutieren sollen.



Bildschirmabzug 6-8 Probelauf: Privilegien hinzufügen

Zudem ändert er den Status des Vorschlags auf „in Diskussion“.



Bildschirmabzug 6-9 Probelauf: Vorschlag-Details

Im Bildschirmabzug 6-9 Probelauf: Vorschlag-Details ist der Vorschlag dargestellt. Zugehörige Objekte sind Objekte des Terminologieservers, in diesem Fall ist es ein Begriff, bei anderen Szenarien können dieses auch mehrere Begriffe, Vokabulare, Beziehungen oder ValueSets sein.

Im unteren Bereich ist der Status zu erkennen. Der Inhaltsverwalter kann mit Hilfe der Buttons den Status ändern, wie im Objektlebenszyklus angegeben bzw. in den Datenbanktabellen entsprechend parametrisiert.

Herr Becker und Herr Schmidt sehen nun, dass sie über den Vorschlag diskutieren können und jeder gibt einen Diskussionsbeitrag ab.

Ihre Antwort:

Ich bin mir nicht sicher...

Datum	Typ	Name	Ersteller	Inhalt

Anlage Hinzufügen Anlage Entfernen

Speichern Abbrechen

Bildschirmabzug 6-10 Probelauf: Neue Antwort

Die Diskussions-Ansicht sieht folgendermaßen aus:

Kollaborationsanwendung für den Terminologieserver der FH Dortmund - (c) 2010 Robert Mützner - Windows Internet Explorer

http://www.term.fh-dortmund.de/kollaboration/index.html

Fachhochschule Dortmund
University of Applied Sciences and Arts

Kollaborationsplattform Terminologieserver

eingeloggt als muetzner
ausloggen

Vorschlag-Details Links Verlauf Diskussion Abstimmung

Aktualisieren Antworten

Becker Berthold Becker KH Do	08.02.2010, 14:16:50 Ich bin mir nicht sicher...	#1
Bearbeiten Zitieren ^		
Schmidt Peter Schmidt KH Bochum	08.02.2010, 14:18:01 Gerade deshalb bin ich mir sicher Anlagen Da kann man alles nachlesen (http://www.google.de)	#2
Bearbeiten Zitieren ^		

Status: in Publikation (08.02.2010) einfrieren sperren deaktivieren Vokabular: Notfallrelevante Medikamente, 2008, Vorschlag: Dabigatran soll hinzugefügt werden zurück

Fertig Internet 100%

Bildschirmabzug 6-11 Probelauf: Diskussion

Jeder Benutzer, welcher einem Vorschlag zugeordnet ist, kann für diesen abstimmen. Er kann nur ein Mal seine Stimme vergeben. „Zustimmung mit Modifikation“ bedeutet, dass er prinzipiell zustimmt, wenn seine Änderungswünsche am Vorschlag akzeptiert werden.

Sie können für jeden Vorschlag 1 Stimme abgeben!

Ihre Stimme: ☐ Ablehnung ☒ Zustimmung ☐ Zustimmung mit Modifikation

Begründung: Siehe Diskussionsbeitrag

OK Abbrechen

Bildschirmabzug 6-12 Probelauf: Abstimmung

Der Inhaltsverwalter kann sich das Ergebnis einer Abstimmung ansehen:

Fachhochschule Dortmund
University of Applied Sciences and Arts

Kollaborationsplattform Terminologieserver

eingelogg als muetznr [ausloggen](#)

Vorschlag-Details Links Verlauf Diskussion **Abstimmung**

Auswertung

Anzahl Teilnehmer: 3
Anzahl Abstimmungen: 3
Anzahl Ablehnungen: 1 (33%)
Anzahl Zustimmungen: 2 (66%)
Anzahl Zustimmungen mit Modifikation: 0 (0%)

Benutzer	Ablehnung	Zustimmung	Zustimmung ...	Begründung
Peter Schmidt		x		Siehe Diskussionsbeitrag
Berthold Becker	x			
Hans Meier		x		

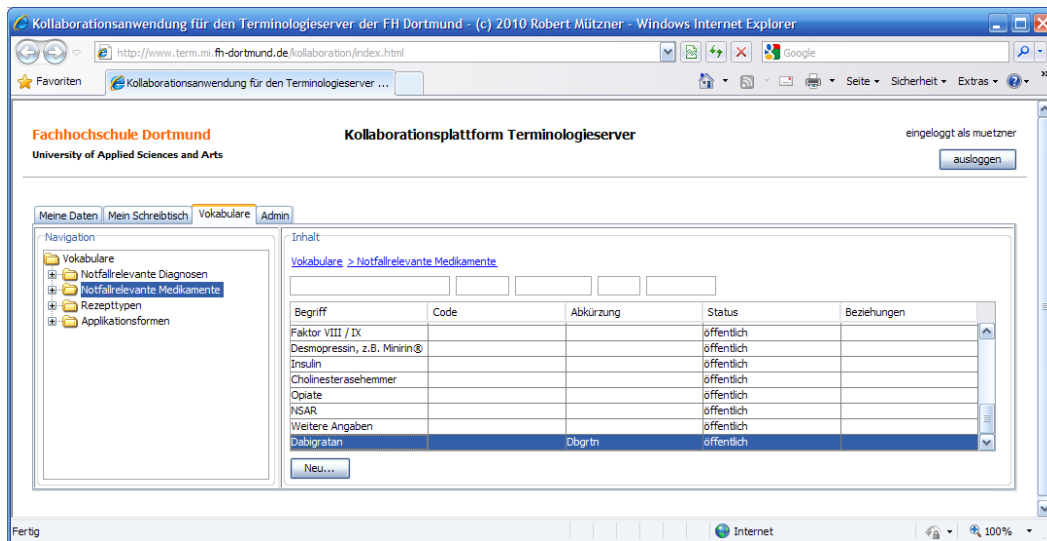
[Abstimmen](#)

Status: in Diskussion (08.02.2010) [ablehnen](#) [produzieren](#) Vokabular: Notfallrelevante Medikamente, 2008, Vorschlag: Dabigratan soll hinzugefügt werden [zurück](#)

Fertig

Bildschirmabzug 6-13 Probelauf: Abstimmung-Ergebnis

Anhand des Ergebnisses von 66% Zustimmung akzeptiert er nun den Vorschlag und gibt ihn anschließend frei („In Publikation“). Die zugehörigen Objekte sind nun im Terminologieserver freigeschaltet und sofort verfügbar.



Bildschirmabzug 6-14 Probelauf: Vokabulare mit neuem Begriff

Das beschriebene Szenario fügt einen neuen Begriff hinzu. Andere Szenarien können sein: „Begriff ändern“, „Begriff ausdifferenzieren“ oder „Begriff löschen“. Die gezeigten Vorgänge beschreiben einen Teilbereich der Software. Auf eine komplette Darstellung muss wegen der Komplexität an dieser Stelle verzichtet werden. Das Programm lässt sich jedoch auch auf dem lokalen PC benutzen, so dass es ausprobiert werden kann.

6.3 Testergebnisse

Der Probedurchlauf zeigt, dass die Kollaborationsumgebung funktionsfähig und anwendbar ist. Die Ladezeiten sind akzeptabel. Da der Datenzugriff ausschließlich per Webservice über das Internet funktioniert, muss eine gewisse Verzögerung, wie es auch bei Webseiten bekannt ist, in Kauf genommen werden.

Bei der Einrichtung der Datenbank muss darauf geachtet werden, dass Tabellen und Attribute korrekt geschrieben sind. Dabei kommt es vor allem auf die Groß- und Kleinschreibung an, wenn das System auf einem Linux-Server läuft. Die MySQL-Datenbank unterscheidet die Groß- und Kleinschreibung im Gegensatz zu Windows-Systemen auf Linux-Systemen.

7 Diskussion und Ausblick

7.1 Diskussion

Vor dem Hintergrund der Zusammenarbeit an Terminologien gilt es abschließend die Kollaborationsumgebung für den Terminologieserver zu diskutieren. Dabei muss zwischen dem Terminologieserver und der Kollaborationsanwendung unterschieden werden. Der Terminologieserver und dessen Dienste werden von der Kollaborationsumgebung benutzt und sind Voraussetzung für dieses Projekt. Damit wird gewährleistet, dass die Kollaborationsanwendung immer mit den neuesten Daten des Terminologieservers arbeitet und dass neu eingepflegte Begriffe oder Terminologien sofort verfügbar sind. Ein weiterer Vorteil ist, dass keine 2. Datenhaltung existieren muss, somit entfällt eine aufwendige Pflege. Damit Begriffe oder Terminologien, welche in der Entwicklung sind, nicht nach außen hin sichtbar sind, liegt dem Terminologieserver eine aufwendige Versionisierung zu Grunde.

Das angestrebte Ziel, Terminologien gemeinsam zu entwickeln und anderen Teilnehmern in der Telematikplattform zeitnah zur Verfügung zu stellen, kann mit Hilfe der Kollaborationsumgebung erreicht werden. Der Ansatz der Zusammenarbeit über das Internet zeigt den Weg in die Zukunft. Mehrere kompetente Fachkräfte können so zusammen an Terminologien entwickeln. Mittels des Diskussionsmoduls kann über fachliche Themen sinnvoll diskutiert und abgestimmt werden – auch unter Berücksichtigung angegebener Quellen wie Links oder Dateien. Antworten können auf Vorredner bezogen werden, so dass der Zusammenhang immer hergestellt ist. Ein Abstimmungsmodul sorgt dafür, dass der Verwalter einen Überblick über die Meinungen der Diskutanten im Forum erhält.

Es können sowohl neue Terminologien erstellt als auch vorhandene Terminologien gepflegt werden. Als Beispiel der Pflege von Terminologien kann die Weiterentwicklung des ICD GM genommen werden. Jährlich gibt es dort ungefähr 400 Aktualisierungsvorschläge, welche bearbeitet werden müssen. Ohne eine sinnvolle Software kommt es schnell zur Unübersichtlichkeit oder gar zu Redundanzen in der Terminologie.

Am Beispiel der Schweinegrippe ist zu erkennen, dass Aktualität einer Terminologie eine wichtige Rolle spielt. Begriffe, welche aktuell nötig wären, sind zurzeit erst in der nächsten Veröffentlichung enthalten. Mittels einer Telematikplattform wäre die neue

Krankheit in vielen Primärsystemen schnell verfügbar, wenn diese eine Synchronisation mit dem Terminologieserver über die Webservices nutzen.

Die Verfügbarkeit über das Internet per Webservice revolutioniert die bisherigen, zum Teil veralteten Verfahren. Ein Dienst, welcher alle neuen Entitäten seit einem bestimmten Zeitpunkt übermittelt, bildet die Grundlage für eine Synchronisation von Terminologien auf moderne Art und Weise.

7.2 Ausblick

Das Java-Applet zeigt, wie die Online-Zusammenarbeit von Terminologien funktionieren kann. Das Rollen- und Rechtekonzept bietet der Anwendung die Sicherheit, dass verschiedene Benutzer zusammenarbeiten können, ohne dass sie sich gegenseitig behindern.

Bevor die Anwendung in den Echtbetrieb aufgenommen werden kann, müssen noch einige Implementierungen durchgeführt werden, welche im Rahmen der Stufe 1 nicht erledigt werden konnten. Das betrifft unter anderem die Diskussionsgruppen, welche verschiedene Nutzer zu einer Gruppe zusammenfassen. Dadurch wird erreicht, dass bei einem neuen Vorschlag automatisch eine berechtigte Gruppe an Nutzern zur Diskussion freigegeben werden.

Der nächste Schritt wird ein Test mit ausgewählten Nutzern sein, welche über eine neue Terminologie diskutieren, beraten und diese mit Inhalten füllen. Der Test mit ca. 15 Personen wird zeigen, in wie fern die Software für den täglichen Gebrauch angepasst werden muss.

Das langfristige Ziel des Terminologieservers soll es sein, dass Primärsysteme die neu erarbeiteten Terminologien mit ihren lokalen synchronisieren können. Zunächst wird es voraussichtlich darauf hinauslaufen, dass Hersteller ihre Terminologien mit dem Server abgleichen und damit ihre Kunden (Primärsysteme) versorgen. Das hat den Vorteil, dass die Terminologie vor der endgültigen Auslieferung nochmals geprüft werden kann.

Das langfristige Ziel der Kollaborationsanwendung soll es sein, dass diese in den Echtbetrieb übergeht. Rechtlich gesehen darf jeder Vorschläge für Terminologien vergeben. Somit wird es auch einen Gast-Account geben müssen, mit dem eine Person ohne Anmeldung neue Vorschläge eingeben kann – wenngleich eigentlich sichergestellt sein sollte, von wem ein Vorschlag stammt.

8 Anlagen

8.1 Installationshinweise

Die Installation des Webservices inklusive der Datenbank ist recht komplex. Eine genaue Beschreibung kann ich an dieser Stelle nicht geben.

Im Folgenden werden die Schritte der Installation genannt:

- Installation von MySQL und Tomcat
- Einrichtung der MySQL-Datenbank mit dem Datenbankskript des Terminologieservers und der Kollaborationsumgebung
- Einrichten der Konfigurationsdateien (term_config.xml und termserver_log4j.xml; diese müssen im Verzeichnis var\www\www.term.mi.fh-dortmund.de\termserver liegen; coll_config.xml im Verzeichnis var\www\www.term.mi.fh-dortmund.de\web\kollaboration)
- Einrichten des Webservices (Deployen der Datei Terminologieserver.war)
- Start des Applets über die Datei index.html

8.2 Datenbankmodell

Das Datenbankmodell ist im Kapitel 4.2 auf der Seite 48 in kleiner Form abgedruckt. Eine große Version befindet sich am Ende der Arbeit in Form eines DIN-A3-Blatts.

8.3 Dienste-Matrix

Die Dienste-Matrix ist ebenfalls als DIN-A3-Blatt am Ende der Arbeit zu finden.

9 Anhang

9.1 Quellenverzeichnis

- Apache. 2010.** Log4j Dokumentation. [Online] 17. Februar 2010. <http://logging.apache.org/log4j/1.2/index.html>.
- Aschhoff, Matthias. 2009.** *Protokoll 1. Workshop Terminologieserver und Kooperationsplattform*. Köln : DIMDI, 2009.
- . **2010.** *Protokoll 2. Workshop Terminologieserver und Kooperationsplattform*. Dortmund : FH Dortmund, 2010.
- Balzert, Heide. 2004.** *Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2*. Heidelberg : Spektrum Akademischer Verlag, 2004. ISBN 3-827-41162-9.
- Balzert, Helmut. 2009.** *Lehrbuch der Softwaretechnik*. Heidelberg : Spektrum Akademischer Verlag, 2009. ISBN 978-3-8274-1705-3.
- DIMDI. 2010.** Das DIMDI - Medizinwissen online. [Online] 16. Februar 2010. <http://www.dimdi.de/>.
- Dugas, Martin. 2003.** *Medizinische Informatik und Bioinformatik*. Heidelberg : Springer-Verlag Berlin, 2003. ISBN 3-540-42568-3.
- Frotscher, Thilo. Java Web Services mit Apache Axis 2. 2007.** Siegen : entwickler.press, Java Web Services mit Apache Axis 2. ISBN 978-3-935042-81-9.
- Gennick, Jonathan. 2006.** *SQL kurz und gut*. Köln : O'Reilly, 2006. ISBN 3-89721-522-5.
- Haas, Peter. 2006.** *Gesundheitstelematik*. FH Dortmund : Springer-Verlag Berlin Heidelberg, 2006. ISBN 3-540-20740-6.
- . **2005.** *Medizinische Informationssysteme und Elektronische Krankenakten*. FH Dortmund : Springer-Verlag Berlin Heidelberg, 2005. ISBN 3-540-20425-3.
- . **2009.** Projektantrag „Entwicklung Prototyp Terminologieserver als Infrastrukturkomponente“. Dortmund : s.n., 2009.
- HL7. 2007.** Version 3 Standard Ballot 2007. [Online] 2007. [Zitat vom: 2. Februar 2010.] <http://www.hl7.org/v3ballot/html/welcome/environment/index.htm>.

HL7, CTS II. HL7 Common Terminology Services 2 Service Functional Model (SFM). [Online]

[http://216.47.173.10/apelcore/index.php/HL7_Common_Terminology_Services_2_Service_Functional_Model_\(SFM\)](http://216.47.173.10/apelcore/index.php/HL7_Common_Terminology_Services_2_Service_Functional_Model_(SFM)).

Johner, Christian und Haas, Peter. 2009. *Praxishandbuch IT im Gesundheitswesen*. München : Carl Hanser Verlag, 2009. ISBN 978-3-446-41556-0.

Kaepfel, Volker. 1998. *Das Wörterbuch medizinischer Fachausdrücke*. Mannheim : Duden, 1998. ISBN 3-411-04616-3.

Sun. 2010. Java 2 Platform Standard Edition 5.0. [Online] 17. Februar 2010. <http://java.sun.com/j2se/1.5.0/docs/api/>.

Systems, Sparx. 2010. Enterprise Architecture Dokumentation. [Online] 17. Februar 2010. <http://www.sparxsystems.com/resources/index.html>.

Thun, Dr. Silvia. 2009. 2. Workshop Terminologieserver. 2. Februar 2009.

Wissensdatenbank, Wikipedia. Knowledge Base. [Online] http://de.wikipedia.org/wiki/Knowledge_Base.

Zeppenfeld, Klaus / Finger, Patrick. 2009. *SOA und WebServices*. s.l. : Springer Berlin, 2009. ISBN 978-3-540-76990-3.

9.2 Verwendete Hilfsmittel und Materialien

Letzter Zugriff auf die Webseiten: 09.02.2010

Produkt	Beschreibung/URL
Enterprise Architect 7.0	Modellierungswerkzeug mit UML und XMI 2.1 Unterstützung http://www.sparxsystems.com.au/
JDK 1.6	Entwicklungswerkzeug für Java http://java.sun.com/javase/downloads/widget/jdk6.jsp
Netbeans 6.7.1	Open-Source-Entwicklungsumgebung http://netbeans.org/
Log4j	Logging-Framework für Java http://logging.apache.org/log4j/docs/index.html
Subversion	Versionverwaltung http://subversion.tigris.org/
MySQL 5.1	Relationales Datenbanksystem http://www.mysql.com/
Tomcat 6	Java Webserver zur Bereitstellung der Webservices http://tomcat.apache.org/
Microsoft Excel	Hilfsmittel zur Erstellung von Tabellen http://office.microsoft.com/de-at/excel/FX100487621031.aspx

9.3 Verzeichnisse

9.3.1 Abbildungen

ABBILDUNG 2-1 TERMINOLOGIESERVER	4
ABBILDUNG 2-2 ENTERPRISE ARCHITECT (AKTEURE)	8
ABBILDUNG 2-3 MICROSOFT EXCEL - KOLLABORATIONSMATRIX	9
ABBILDUNG 2-4 FUNKTIONSWEISE EINES WEBSERVICES	13
ABBILDUNG 2-5 PROJEKT-ZEITPLAN	16
ABBILDUNG 3-1 ÜBERSICHT DER AKTEURE	17
ABBILDUNG 3-2 ÜBERSICHT DER ANWENDUNGSFÄLLE (USE CASES)	20
ABBILDUNG 3-3 PROGRAMMABLAUF	27
ABBILDUNG 3-4 AKTIVITÄTSDIAGRAMM: OBJEKTLEBENSZYKLUS	28
ABBILDUNG 4-1 BEISPIEL UMWANDLUNG M:N-TABELLEN	49
ABBILDUNG 4-2 DIENSTMATRIX (AUSZUG)	52
ABBILDUNG 6-1 TESTUMGEBUNG ÜBERBLICK	59

9.3.2 Tabellen

TABELLE 3-1 MÖGLICHE STATUSÄNDERUNGEN	31
TABELLE 5-1 GESCHÄTZTES DATENBANKVOLUMEN	54

9.3.3 Modelle

KLASSENMODELL 4-1 FACHKONZEPT-KLASSENMODELL	36
KLASSENMODELL 4-2 SUBMODEL ROLLEN- UND BENUTZERKONZEPT	37
KLASSENMODELL 4-3 SUBMODEL VORSCHLAG-KONZEPT	40
KLASSENMODELL 4-4 SUBMODEL DISKUSSION	43
KLASSENMODELL 4-5 SUBMODEL STATUS-VERWALTUNG	45
KLASSENMODELL 4-6 SUBMODEL SICHERHEIT	47
DATENBANKMODELL 4-1 KOLLABORATIONS-DATENBANKMODELL	48

9.3.4 Bildschirmabzüge

BILDSCHIRMAZUG 5-1 DIENST HINZUFÜGEN	56
BILDSCHIRMAZUG 6-1 PROBELAUF: APPLLET-SIGNATUR.....	60
BILDSCHIRMAZUG 6-2 PROBELAUF: ANMELDUNG.....	60
BILDSCHIRMAZUG 6-3 PROBELAUF: MEINE DATEN	61
BILDSCHIRMAZUG 6-4 PROBELAUF: VOKABULARE	62
BILDSCHIRMAZUG 6-5 PROBELAUF: NEUER BENUTZER.....	62
BILDSCHIRMAZUG 6-6 PROBELAUF: NEUER VORSCHLAG	63
BILDSCHIRMAZUG 6-7 PROBELAUF: MEIN SCHREIBTISCH	64
BILDSCHIRMAZUG 6-8 PROBELAUF: PRIVILEGIEN HINZUFÜGEN	64
BILDSCHIRMAZUG 6-9 PROBELAUF: VORSCHLAG-DETAILS	65
BILDSCHIRMAZUG 6-10 PROBELAUF: NEUE ANTWORT	66
BILDSCHIRMAZUG 6-11 PROBELAUF: DISKUSSION	66
BILDSCHIRMAZUG 6-12 PROBELAUF: ABSTIMMUNG	67
BILDSCHIRMAZUG 6-13 PROBELAUF: ABSTIMMUNG-ERGEBNIS	67
BILDSCHIRMAZUG 6-14 PROBELAUF: VOKABULARE MIT NEUEM BEGRIFF	68

9.4 Abkürzungsverzeichnis

Abkürzung	Beschreibung
BLOB	B inary L arge O bject
CT	C omputer t omographie
DB	D aten b ank
DIMDI	D eutsches I nstitut für M edizinische D okumentation und I nformation
FH	F ach h ochschule
GUI	G raphical U ser I nterface
HTTP	H ypertext T ransfer P rotocol
ICD	I nternational C lassification of D iseases
ID	I dentifikationsnummer
IT	I nformation s technik
JAR	J ava A rchive
JIT	J ust I n T ime (gerade rechtzeitig)
MDA	M odel D riven A rchitecture
OOD	O bjekt- O rientiertes D esign
OPS	O perationen- und P rozeduren s chlüssel
PC	P ersonal C omputer
PHP	P HP: H ypertext P reprocessor
SDK	S oftware D evelopment K it
SOAP	S imple O bject A ccess P rotocol
SP	S ervice P ack
SQL	S tructured Q uery L anguage
SSL	S ecure S ockets L ayer
u.U	u nter U mständen
UDDI	U niversal D escription, D iscovery and I ntegration
UML	U nified M odeling L anguage
URI	U niform R esource I dentifier
URL	U niform R esource L ocator
UUID	U niversally U nique I dentifier
WSDL	W eb S ervice D escription L anguage
WWW	W orld W ide W eb
XML	E xtensible M arkup L anguage

9.5 Glossar

Begriff	Erklärung
Akkreditierung	Der Begriff Akkreditierung stammt vom lateinischen Wort „accredere“, welches als „Glauben schenken“ übersetzt wird. Es beschreibt den Zustand, dass eine anerkannte Instanz (z.B. eine Person) einer anderen das Erfüllen einer besonderen Eigenschaft bescheinigt. Im Allgemeinen ist eine akkreditierte Person stark in der Materie eingearbeitet und diese kann als vertrauenswürdig eingestuft werden.
Akteur	Teilnehmer an einem System
Applet	In einem Webbrowser laufendes Java-Programm (Java-Applet).
Begriff	Ein Begriff ist ein Wort in einem Vokabular.
Datenbank	Ein Datenbanksystem ist ein System zur elektronischen Verwaltung von Daten. Ein System kann mehrere Datenbanken beinhalten.
ICD	ICD steht für „Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme“ und ist ein Klassifikationssystem für Diagnosen in der Medizin.
Klassenmodell	Darstellung von Objekten und Beziehungen der realen Welt inklusive Attributen.
Klassifikation	Fasst Begriffe zusammen, die in mindestens einem klassenbildenden Merkmal übereinstimmen.
Kollaboration	Unter einer Kollaboration (lat. co = „mit“, laborare = „arbeiten“) ist die Mitarbeit bzw. Zusammenarbeit mit mehreren Gruppen und/oder Personen gemeint. Heutzutage wird es als Synonym für die Kooperation verwendet.
Logging	Aufzeichnung von Werten oder Meldungen
MimeType	Der Internet Media Type klassifiziert die Daten im Rumpf einer Nachricht im Internet. Dateiendungen werden üblicherweise einem MimeType zugeordnet, damit die Verwendung der Datei eindeutig ist.
Primärsystem	Ein Anwendungssystem bei einem Leistungserbinger (z.B. Arztpraxissystem).
Semantische Interoperabilität	Fähigkeit von 2 Maschinen, Daten auszutauschen und zu verstehen.

SQL	SQL ist eine Datenbanksprache zur Definition, Abfrage und Manipulation von Daten in relationalen Datenbanken. SQL ist von ANSI und ISO standardisiert und wird von fast allen gängigen Datenbanksystemen unterstützt.
Taxonomie	Eine generisch-hierarchische Klassifikation
Telematikplattform	Eine Telematikplattform ist durch eine flächendeckende Infrastruktur definiert. Sie legt einheitliche Rahmenbedingungen zur Nutzung der Plattform und der Daten fest. Diese Bedingungen sind auch politisch.
Terminologie	Als Terminologie wird die Gesamtheit aller Begriffe und Benennungen einer Fachsprache bezeichnet. Sie können als Wörterbuch, Glossar oder Thesaurus vorliegen.
Use Case	Ein Anwendungsfall; beschreibt die Interaktion eines Akteurs mit dem System
UUID	Universally Unique Identifier ist ein Standard für Identifikatoren. Die Zeichenlänge einer Nummer beträgt in der Regel 16 Byte.
Vokabular	Ein Vokabular ist eine Sammlung von Wörtern eines Fachgebiets.
Webservice	Ein Webservice ist ein Programm/Dienst im Internet, welches per URI eindeutig gekennzeichnet ist. Als Schnittstelle dienen XML Nachrichten.

9.6 Index

A

Akteur..... 8, 15, **17**, 18, 20, 37
 Aktionen**26**, 30
 Anwendungsfall 2, 15, **20**, 32

B

Benutzer**18**
 Benutzeroberfläche35

D

Datenbankmodell.....**48**
 Dienste 6, 9, 15, 50, **51**, 55
 Dienstematrix.....55, 56
 Diskussion **23**, 27, 29, 43, 69

E

Enterprise Architect **7**, 8, 15, 49, 53, 75

J

Java 2, **10**, 14, 33
 Java-Applet 2, 6, **11**, 50, 58

K

Kollaborationsumgebung... 2, 6, 15, 17, 28,
 35, 69

L

Log4j**11**, 75

M

Microsoft Excel.....**9**, 75
 MySQL **12**, 33, 53, 75

N

Netbeans **8**, 55, 58, 75

P

Pflichtenheft32
 Probelauf59, 60

R

Rolle
 Admin..... 18
 Antragsteller 18
 Autor 18
 Betreiber..... 19, 24, 25
 Inhaltsverwalter **18**, 22, 28
 Rezensent..... 18
 Taxonom..... 19
 Rollen..... 31, 34, 37, 39

S

Schnittstelle 13, 41, 43, 50
 Sicherheit.....7, 10, 11, 33, 47
 Signatur 60
 SQL 53
 Status21, 22, 26, 27, 28, **29**, 30, 31, 39, 45
 Statuswechsel 31

T

Telematikplattform..... 1
 Terminologie..... 2, 3
 Terminologieserver 1, 4, 5, 6, 15, 33, 39
 Test..... 2, 32, 59
 Tomcat **14**, 33, 75

U

UML.....7

W

Webservice2, 4, **13**, 50, 55
 Wissensdatenbank 17

X

XML..... 13, 14

Eidesstattliche Erklärung

Ich versichere an Eides statt, dass ich die vorliegende Arbeit selbständig angefertigt und mich keiner fremden Hilfe bedient sowie keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichten oder nicht veröffentlichten Schriften und anderen Quellen entnommen sind, habe ich als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, 17.02.2010

Robert Mützner

Erklärung

Mir ist bekannt, dass nach § 156 StGB bzw. § 163 StGB eine falsche Versicherung an Eides Statt bzw. eine fahrlässige falsche Versicherung an Eides Statt mit Freiheitsstrafe bis zu drei Jahren bzw. bis zu einem Jahr oder mit Geldstrafe bestraft werden kann.

Dortmund, 17.02.2010

Robert Mützner