

Endbericht



Entscheidungs-Unterstützung: **R**ollengerecht,
Effizient, **K**ooperativ, **A**llgegenwärtig

Autoren:

Youssef Belkhadir, Michael Dimov, Minye Dong, Michael Drazek,
Jaouad El Jerroudi, Christian Kuete-Tsatedem, Daniel Pierlings,
Mohamed Said, Dennis Saßmannshausen, Holger Willebrandt, Jing Zhou

PG 440
am Fachbereich Informatik
der Universität Dortmund
Lehrstuhl V

14. September 2004

Betreuer:

Dipl. Inform. Markus Bajohr
Dipl. Ing. Martin Karuseit
Prof. Dr. Bernhard Steffen

Inhaltsverzeichnis

1	Einleitung	1
2	Online Conference Service	3
2.1	Was ist OCS?	3
2.1.1	Das personalisierte, internetbasierte Entscheidungssystem	3
2.1.2	Einsatzfelder des OCS	4
2.2	User-/Rollenmanagement	4
2.3	Features	6
2.3.1	Article Feature	6
2.3.2	Bidding Feature	6
2.3.3	Conflict Feature	6
2.3.4	Delegation Feature	6
2.3.5	Email Feature	7
2.3.6	Evaluation Feature	7
2.3.7	Preferences Feature	7
2.3.8	Profile Feature	7
2.3.9	Report Feature	8
2.3.10	Role Feature	8
2.3.11	Setup Feature	8
2.3.12	Staff Feature	9
2.3.13	Tasklist Feature	9
2.3.14	User Feature	9
2.4	Technische Konzepte	10
2.4.1	Client-Server-Architektur	10
2.4.2	Verteiltes System	11

3	Entwicklung und Modifikation	13
3.1	Makrofizieren	13
3.1.1	Vorteile des Makrokonzepts	13
3.1.2	Modifikation des Dienstes	15
3.1.2.1	Voraussetzung	15
3.1.2.2	Vorgehen	15
3.2	Einzelkurzthemen	15
3.2.1	Article Feature	15
3.2.2	Delegation Feature	23
3.2.3	Email Feature	26
3.2.4	Evaluation Feature	31
3.2.5	News Feature 1	37
3.2.6	News Feature 2	43
3.2.7	Staff Feature	48
3.2.8	Setup Feature 1	54
3.2.9	Setup Feature 2	57
3.2.10	Sub-Delegation Feature	63
3.2.11	User Feature	69
3.3	Gruppenkurzthemen	75
3.3.1	Abstract submission workflow	75
3.3.2	Bidding workflow	89
3.3.3	File download workflow	100
3.3.4	Email Template Setup Workflow	107
4	PG Hauptthemen	113
4.1	Management Overview System (MOS)	113
4.2	Support- und Feedbacksystem	136
4.3	Production Feature	149
4.4	Service Status Report Feature	156
5	Ausblick	171
6	Literaturverzeichnis	175
	Literaturverzeichnis	177

Abbildungsverzeichnis

2.1	Verhältnis User, Rollen, Permissions	5
2.2	Ablauf zwischen Client und Server	11
3.1	Makrofizierter Graph	14
3.2	Use Case „Submit New Article“	16
3.3	Aktivitätsdiagramm „Submit New Article 1“	17
3.4	Activity Chart „Submit New Article 2“	18
3.5	Makro „Old Article Macro“	19
3.6	Makro „New Article Macro“	20
3.7	Conference Graph „successful submission 1“	21
3.8	Conference Graph „successful submission 2“	22
3.9	Activity Chart ”Delegation Feature”	24
3.10	Activity Chart 2 ”Delegation Feature”	25
3.11	Use Case des Istzustands ”Write new Email”	27
3.12	Use Case des Sollzustands ”Write new Email”	28
3.13	Activity Chart des Istzustands ”Write new Email”	29
3.14	Activity Chart des Sollzustands ”Write new Email”	30
3.15	Activity Chart ”Evaluation Feature”	32
3.16	Use Case ”Evaluation Feature”	33
3.17	Screenshot old page ”Evaluation Feature”	34
3.18	Screenshot new page ”Evaluation Feature”	35
3.19	Screenshot ”showinfo-all” beim ”Evaluation Feature”	36
3.20	Use Case des Ist-Zustands ”News Feature 1”	38
3.21	Use Case des Soll-Zustands ”News Feature 1”	39
3.22	Activity Chart "Show messages" Ist-Zustand des ”News Feature 1”	40

3.23	Activity Chart "Show messages" Soll-Zustand "News Feature 1"	40
3.24	Use Case "News Feature 2" Ist-Zustand	43
3.25	Use Case "News Feature 2" Soll-Zustand	44
3.26	Activity Chart "News Feature 2"	45
3.27	Activity Chart "reply message" und "post message"	46
3.28	Abschnitt des Graphen, der im News Feature 2 eingesetzt wurde	47
3.29	Checkbox im HTML-Dokument nach Modifizierung des News Feature Workflows	47
3.30	Use Case "AddStaffMember" des Staff Feature	49
3.31	Activity Chart "AddStaffMember" vorheriger Zustand des Staff Feature	50
3.32	Activity Chart "AddStaffMember" Soll Zustand des Staff Feature	51
3.33	Ein Screenshot des modifizierten "AddStaffMember"-Formulars	53
3.34	Zustand vor der Bearbeitung des Kurzthemas "Setup Feature 1"	55
3.35	Zustand vor der Bearbeitung des Kurzthemas "Setup Feature 1"	56
3.36	Zustand nach der Bearbeitung des Kurzthemas "Setup Feature 1"	57
3.37	Zustand nach der Bearbeitung des Kurzthemas "Setup Feature 1"	58
3.38	Use Case Diagramm des Zielzustands "Setup Feature 2"	59
3.39	Ablauf des Unterpunktes "Role Configuration" im Setup, Zielzustand	61
3.40	Activity Chart der Registration, Zielzustand	62
3.41	Use Case Diagramm "Sub-Reviewer anzeigen"	64
3.42	Use Case Diagramm "Rolleneigenschaften ändern"	64
3.43	Activity Chart "erweiterte Report List" des "Sub-Delegation Feature"	65
3.44	Activity Chart "Rolle modifizieren" des "Sub-Delegation Feature"	66
3.45	Screenshot "Sub-Reviewer anzeigen"	68
3.46	Screenshot "CSV-Ansicht der Sub-Reviewer anzeigen"	68
3.47	Use Case „Remove User“	69
3.48	Activity Chart "Remove User" Ist-Zustand	70
3.49	Activity Chart "Remove User" Soll-Zustand	72
3.50	Macro „user“_alt "User Feature"	73
3.51	Macro „user“_neu "User Feature"	74
3.52	Use Case SubmitArticle "Abstract submission workflow"	76
3.53	Use Cases: Mögliche Aktionen die für einen Artikel ausgeführt werden können (in der Actions-Spalte der Artikel-Tabelle)	77

3.54	Abstract-Submission im Setup	79
3.55	Activity Chart "SubmitAbstract"	80
3.56	Activity Chart "ReadArticleShouldBeState"	81
3.57	Activity Chart "ModifyDeadlines"	82
3.58	Activity Chart "ModifyRole"	83
3.59	Activity Chart "ModifyArticle"	85
3.60	Activity Chart "ArticleStateChanges", Ist-Zustand	86
3.61	Activity Chart "ArticleStateChanges", Soll-Zustand	87
3.62	Auslesen von Konflikten aus der Conflict Matrix	92
3.63	Bidding Matrix mit Anzeige eines Konfliktes aus der Conflict Matrix und Verwendung der Farbcodes aus dem Delegation Feature	93
3.64	Übertragen von Bidding - Konflikten in die Conflict Matrix	94
3.65	Activity Chart:Anzeige eigener Artikel	95
3.66	Bidding Liste mit Anzeige eigener Artikel	96
3.67	Bidding Matrix mit Farbcodes aus der Delegation	97
3.68	Anzeige eines zurückgewiesenen Artikels in der Bidding Matrix	98
3.69	Speichern von Biddings, ohne Datenfelder des PC Chair zu überschreiben	99
3.70	Activity Chart "Download Article Files as Zipfile"	102
3.71	Makro "Article Macro 1"	103
3.72	Makro "Article Macro 2"	104
3.73	Editieren der Email Templates	108
4.1	Use Case: „MOS Features“	114
4.2	Use Case: „Conference-Features“	115
4.3	MOS: „Datenbankmodell“	116
4.4	Activity Chart: „Neue Konferenz hinzufügen“	117
4.5	Activity Chart: „Login eines MOS-Users in einen OCS-Dienst“	118
4.6	Activity Chart: „Login zum Statusreport (OCS-Seite)“	119
4.7	Activity Chart: „Konferenzdaten modifizieren“	120
4.8	Activity Chart: „Bearbeiten der Zugangsberechtigungen zur Konferenz“ .	121
4.9	Activity Chart: „User-Registrierung ohne Shadow-Passwort“	122
4.10	Activity Chart: „User-Registrierung mit Shadow-Passwort“	124
4.11	„Konferenz-Liste“	125

4.12	„Neue Konferenz hinzufügen“	126
4.13	„Zugangsberechtigungen zur Konferenz modifizieren“	127
4.14	Use Case des OCS Support Dienstes	136
4.15	Activity Chart des gesamten Dienstes	144
4.16	Eingabemaske für die FAQ Kategorien und Sektionen	145
4.17	Eingabemaske für die FAQ	145
4.18	Übersicht der eingegangenen Supportanfragen	146
4.19	Generierte FAQ	147
4.20	Übersicht der abgelegten Support Templates	147
4.21	Eingabemaske für die XML Template	148
4.22	Das Use Case des Production Features	150
4.23	Das Activity Chart des Production Features	151
4.24	Use Case des interaktiven Arbeitsablaufs	157
4.25	Use Case des automatischen Arbeitsablaufs	157
4.26	Activity Chart des interaktiven Arbeitsablaufs	158
4.27	Activity Chart des automatischen Arbeitsablaufs	158
4.28	Zum Report gehörende Features	159
4.29	Setup des Reports	160
4.30	User Seite zum Report	161
4.31	Administratoren und PC Chairs	161
4.32	Legende der Artikel IDs	162
4.33	Anzahl von Artikeln pro Status	163
4.34	Informationen zu delegierten Artikeln	163
4.35	Informationen zu Reports	164
4.36	Im CVS gespeicherte Reports	165
4.37	Service Graph: Setup des Status Reports	168
4.38	Service Graph: Der SIB <i>Counter</i>	168

Kapitel 1

Einleitung

Internetbasierte Entscheidungssysteme ermöglichen eine kooperative Entscheidungsfindung unabhängig von Zeit und Ort. Durch die systemgestützte Steuerung komplexer Entscheidungsprozesse mit Zeitplanungen, Aufgabenverteilungen, Entscheidungsprotokollen und Ausnahmebehandlungen, ist es möglich, Entscheidungssysteme konkreten Situationen, Rollenprofilen und Echtzeitanforderungen anzupassen. Derartige Kooperations- und Entscheidungssysteme werden typischerweise durch mehrstufige Web-Applikationen als Online-Dienste realisiert, die auf mehreren, unterschiedlichen Systemen verteilt ausgeführt werden.

Sind heterogene Nutzerprofile zu "bedienen", so erfordert dies den Einsatz personalisierter Web-Applikationen, die jedem Benutzer eine massgeschneiderte Sicht auf die Dienstfunktionalitäten und Daten geben. Dies wird typischerweise dadurch gewährleistet, dass dem Benutzer eine oder mehrere Rollen zugewiesen werden, die jeweils mit einem spezifischen Kompetenzzentrum versehen sind. Je nach Komplexität der Rollen- und Rechtestruktur, sowie ihrer Wandel- und Erweiterbarkeit erfordert dies ein flexibles Rollen- und Rechte-Management.

Ein Beispiel für ein solches personalisiertes und internetbasiertes Entscheidungssystem ist der Online Conference Service (OCS), der wissenschaftliche Begutachtungsprozesse koordiniert und Nutzer weltweit, rund um die Uhr, mit aktueller, auf sie zugeschnittener Information versieht und sie bei ihren Aktivitäten unterstützt.

Das vorhandene, modulare Entscheidungssystem, das auch dem OCS zugrunde liegt, wird im Rahmen der Projektgruppe zu einem Meta-Entscheidungssystem erweitert werden, mit dessen Hilfe sich anwendungsspezifische Entscheidungssysteme erzeugen lassen.

In diesem Endbericht wird die Arbeit der Projektgruppe erläutert. Als Vorbereitung auf die anstehenden Aufgaben fand am Anfang eine Seminarphase statt. In dieser Phase wurde das benötigte Wissen in Form von Einzelvorträgen erworben. Diese Ausarbeitungen sind im Literaturverzeichnis unter [1], [2], [3], [4], [5], [8], [9], [10] und [12] zu finden. Auf diesen Kenntnissen sind die folgenden Kapitel aufgebaut. Zunächst wird der OCS Dienst, der von den PG-Teilnehmern modifiziert und erweitert wurde, näher beschrieben. Es werden die technischen Konzepte, das Rollenmanagement, sowie die Möglichkeiten des OCS anhand des Featurekonzepts erläutert. Das dritte Kapitel beschäftigt sich zuerst mit dem Makrofi-

zieren. Dies war der erste Schritt, den die Projektgruppe bewältigen mußte. Anschliessend werden die ersten Kurzaufgaben beschrieben, die jeweils von einem PG-Teilnehmer bearbeitet wurden. Bei diesen Aufgaben handelte es sich um Erweiterungen eines Feature des OCS. Nach den Kurzthemen folgten die Gruppenkurzthemen. Diese wurden in Dreiergruppen bearbeitet und bestanden aus der Entwicklung oder Erweiterung neuer Features. Das vierte Kapitel beschreibt die PG Hauptthemen, welche im zweiten Semester von den Gruppen bearbeitet wurden. Das fünfte Kapitel gibt einen kurzen Ausblick auf mögliche Verbesserungen, die an der in der Projektgruppe erstellten Software möglich wären und ein allgemeines Fazit.

Kapitel 2

Online Conference Service

Die Einführung stellt die Beschreibung des Online Conference Service dar, in der das Konzept des personalisierten Entscheidungssystems näher erläutert wird.

2.1 Was ist OCS?

Beim OCS handelt es sich um ein internetbasiertes, kooperatives Entscheidungssystem, das rollenbasiert komplexe Entscheidungsprozesse, wie sie bei Kommissions-, Gremien- oder Komiteesitzungen entstehen, unterstützt. Es ermöglicht eine kooperative Entscheidungsfindung unabhängig von Ort und Zeit und stellt den Zustand und Verlauf der Entscheidungsfindung in Echtzeit dar. Dabei reagiert es selbständig auf Veränderungen, wie das Auslaufen einer Deadline oder neue Tasklist-Einträge der Benutzer.

2.1.1 Das personalisierte, internetbasierte Entscheidungssystem

Zugeschnitten ist der OCS auf Kongresse oder Tagungen, zu denen Artikel zu bestimmten Themengebieten von Autoren eingereicht werden. Diese durchlaufen einen Begutachtungsprozess und werden entsprechend dem Urteil der Begutachter bewertet. Ausgehend von den abgegebenen Bewertungen, werden die Artikel schließlich zugelassen oder abgelehnt.

Die Autoren reichen zunächst eine Zusammenfassung (Abstract) ihres Artikels ein, der dann von anderen Mitgliedern begutachtet und bewertet wird. Die Begutachter geben Empfehlungen oder Ablehnungen ab, die den Programmkomiteevorsitzenden bereitgestellt werden. Diese diskutieren mit den Begutachtern über die Zulassung des Artikels. Sowohl der Prozess des Einreichens eines Abstracts, wie des Einreichens des endgültigen Artikels, können mit Fristen belegt sein.

Der OCS unterstützt die gesamten Begutachtungs- und Bewertungsprozesse und macht sie transparent für alle Rollen, die darin involviert sind. Desweiteren unterstützt es die Kommunikation zwischen den beteiligten Personen durch Email und News. Jedes Committee

Member kann einen Stab (Staff) von Mitgliedern besitzen, an die Aufgaben delegiert werden können. Die jeweiligen Mitglieder werden automatisch vom Dienst benachrichtigt und auf ihre neuen Aufgaben hingewiesen. Dies geschieht sowohl durch Emails, als auch durch entsprechende Einträge in der Tasklist der Mitglieder.

Darüberhinaus ist es komplett online konfigurierbar. Dies reicht vom Aussehen der Internetseiten, über das Dateiformat eingereichter Artikel oder Abstracts sowie den Funktionen, auf die einzelne Rollen Zugriff haben, bis hin zu den vom Dienst versandten Emails. Ebenso können die bereits erwähnten Fristen während des laufenden Betriebs, verändert werden.

2.1.2 Einsatzfelder des OCS

Wie bereits erwähnt, sind die Einsatzfelder des OCS Kommissions- oder Gremiensitzungen, zu denen Artikel von Autoren zu Fachthemen eingereicht werden. Besonders solche Kommissionen, die aus örtlich verteilten oder internationalen Mitgliedern bestehen, profitieren von dem ortsunabhängigen Zugriff auf das Echtzeitsystem.

2.2 User-/Rollenmanagement

Mittels des User-/Rollenmanagement Features wird für den jeweiligen Benutzer eine eingeschränkte, personalisierte Sicht auf den Dienst und dessen Komponenten zugeschnitten. Verschiedene Rollen können verschiedene Aufgaben in der Konferenz übernehmen. Durch das User-/Rollenmanagement ist ein System vorhanden, mit dem Berechtigungen an User gezielt vergeben und entzogen werden können.

Im OCS kann ein einzelner User mehrere Rollen bekleiden, zum Beispiel Autor und PC Chair zu gleich (siehe Abbildung 2.1). Es ist möglich während einer Sitzung zwischen den Rollen zu wechseln und daher nicht nötig bei mehreren Rollen zwei verschiedene Logins für eine Person einzurichten. Die Verwaltung der Rechte der User wird erheblich vereinfacht, indem die Rechte von Usern nicht einzeln eingerichtet werden, sondern diese an die Rolle gebunden sind. Features und Permissions werden den einzelnen Rollen gewährt und die Benutzer verfügen dann über die Rechte, die ihrer Rolle zugewiesen sind.

Wird ein neuer User in den OCS aufgenommen, wird in dem hierzu angezeigten Formular neben einer Reihe von personenbezogenen Informationen, wie Name und Email-Adresse, auch eine Rolle ausgewählt. Später ist es über das User-Management möglich dem User weitere Rollen zuzuweisen und auch zu entziehen.

Die verfügbaren Rollen können zum Teil vom System vorkonfigurierte Rollen sein, wie "Administrator", "PC Chair", "PC Member" und "Author". Es ist allerdings auch möglich vollkommen neue und eigenständige Rollen zu erstellen - Hauptwerkzeug hierbei ist die Role-/Rights-Matrix. Bei der Erstellung einer neuen Rolle wird eine Liste von Optionen angezeigt, die in verschiedene Features gruppiert ist, damit die Übersichtlichkeit gewahrt

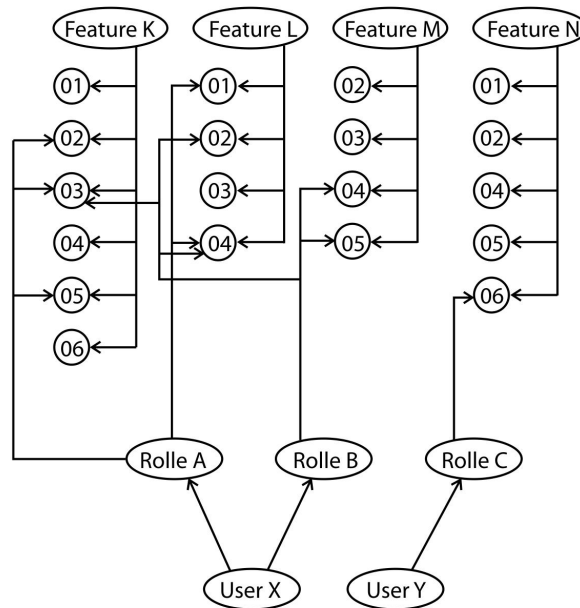


Abbildung 2.1: Verhältnis User, Rollen, Permissions

bleibt. So sind zum Beispiel alle Rechte, die das Artikel-Feature betreffen, in einem Abschnitt zusammengefasst. Die einzelnen Rechte, die hier vergeben werden können werden als "Permissions" bezeichnet. Das Recht einen eigenen Artikel einreichen zu dürfen ist zum Beispiel die Permission F-ART-03 (Feature Article, Permission 3). Auf diese Weise sind alle Artikel-Feature-Permissions durchnummeriert. Features, die das User-Feature betreffen, sind in der Form F-USER-04 ("create new user") durchnummeriert. Bei Modifikation einer Rolle können alle diese Rechte einzeln zugeteilt oder entzogen werden. Manche Permissions verfügen zusätzlich über einen Suffix, um die Berechtigungen feingranularer zu spezifizieren. Ein Beispiel hierfür ist die Erlaubnis Artikel zu lesen (F-ART-05). In diesem Fall gibt es zusätzlich die Unteroption, ob nur die eigenen Artikel (F-ART-05.own), nur die delegierten (F-ART-05.delegated) oder alle Artikel (F-ART-05.all) gelesen werden dürfen.

Mittels der Permissions wird für den jeweiligen Benutzer eine eingeschränkte, personalisierte Sicht auf den Dienst und dessen Komponenten zugeschnitten.

Die für die jeweilige Konferenz gesetzten Fristen für die Einreichung von Abstracts, Artikeln, Biddings und Begutachtungen können die Berechtigungen von Rollen im Dienst beeinflussen. Wenn zum Beispiel die Frist für die Einreichung neuer Artikel abgelaufen ist, wird der Rolle "Author" die Berechtigung zum Einreichen neuer Artikel entzogen.

2.3 Features

2.3.1 Article Feature

Im Article Feature kann der Benutzer zum einen neue Artikel einreichen, zum anderen bereits eingereichte Artikel einsehen. Die bereits eingereichten Artikel werden in einer Tabelle dargestellt und entsprechend der gesetzten Berechtigungen für die derzeit aktive Rolle können die Artikel gelesen, modifiziert oder entfernt werden. Wenn der Artikel schon begutachtet wurde, kann von hier aus auch der Endbericht eingesehen werden. Die Tabelle kann nach Artikel ID, Titel, Autor und Typ sortiert und bei Bedarf als ASCII Text exportiert werden.

2.3.2 Bidding Feature

Das Bidding Feature dient zur strukturierten Verteilung der Begutachtungen an die PC Member. Das Feature unterteilt sich in die Bidding Liste und die Bidding Matrix. Die Bidding Liste zeigt dem PC Member alle eingereichten Artikel an und ermöglicht es ihm zu jedem einzelnen Artikel sein Bidding abzugeben. Es kann angegeben werden, ob man den Artikel begutachten möchte, ob man dem Artikel undifferenziert gegenübersteht, ob man ihn nicht begutachten möchte, oder ob man mit dem Artikel in Konflikt steht, weil man eventuell den Autor persönlich kennt oder andere Gründe gegen eine Begutachtung sprechen. Die Bidding Matrix zeigt dem PC Chair die abgegebenen Biddings von allen PC Mitgliedern zu jedem eingereichten Artikel. Der PC Chair kann aus der Bidding Matrix heraus die Artikel entsprechend den eingereichten Biddings zur Begutachtung an PC Member verteilen. Unterstützt wird er in der Bidding Matrix durch die Anzeige der vergebenen Begutachtungen pro Artikel und pro PC Member, um eine ausreichende und gerechte Verteilung der Begutachtungen zu gewährleisten.

2.3.3 Conflict Feature

Mittels des Conflict Feature kann der PC Chair Konflikte für bestimmte Artikel- / Begutachterkombinationen festlegen. Ein Konflikt tritt immer dann auf, wenn ein PC Member selbst Autor oder Co-Autor eines Artikels ist oder eine persönliche Beziehung zwischen PC Member und dem Autor des Artikels besteht. Die hier eingetragenen Konflikte können beim späteren Delegieren der Begutachtungsaufgaben berücksichtigt werden.

2.3.4 Delegation Feature

Das Delegation Feature dient dem PC Chair zum Delegieren der Artikelbegutachtungen an PC Member. Für die vergebenen Delegationen können den PC Mitgliedern eine Entscheidungsfrist über die Annahme der Delegation und eine Frist zur Abgabe des Reports

aufgelegt werden. Diese Fristen können später bei Bedarf noch verschoben werden. Mittels der Delegationsliste kann der Begutachtungsprozess verfolgt und nachvollzogen werden.

2.3.5 Email Feature

Das Email Feature besteht im Wesentlichen aus drei Unterfeatures um das Versenden von Emails komfortabler zu gestalten.

- Dem Benutzer wird die Möglichkeit zur Verwaltung und Erstellung von Email Templates geboten. Dabei können neue Templates erstellt und Existierende für die eigenen Bedürfnisse angepasst werden.
- Die Nutzung von Textmakros wird erlaubt. Es dürfen globale, vordefinierte Textmakros des OCSs, sowie eigene Textmakros in Emails benutzt werden.
- Um das Versenden neuer Emails zu vereinfachen werden die Empfänger, aufgeteilt nach allen Rollen des Dienstes, sowie nach allen Rollen des eigenen Personalstabs präsentiert.

2.3.6 Evaluation Feature

Die Evaluationmatrix gibt eine Übersicht über alle PC Member Reports, welche dem Komitee als Basis für die Diskussion dienen. Das Komitee besteht aus den PC Chairs und den PC Mitgliedern. Bei der Evaluationmatrix haben nur die PC Chairs das Recht zu editieren, während die PC Member nur lesenden Zugriff haben. Anhand der Evaluationmatrix kann man mittels eines Farbcodes feststellen, in welchem Diskussionsstatus sich der einzelne Artikel befindet (accepted, propose accepted, discussion, propose rejected, rejected). Über die Evaluationmatrix kann das Komitee auf die Diskussionsforen zugreifen, welche mittels der Funktionen OpenForum und CloseForum vom PC Chair geöffnet, bzw. geschlossen werden können. Über Modify kann der PC Chair die einzelnen Bewertungen der Artikel (scores/confidence) für einen oder mehrere Artikel ändern.

2.3.7 Preferences Feature

Über die Preferences hat der User die Möglichkeit, sofern er über mehrere Rollen im OCS verfügt, eine bestimmte auszuwählen, die nach dem Einloggen als Standard aktiv sein soll. Diese Einstellung wird hier gespeichert und gilt dann für jedes weitere Einloggen.

2.3.8 Profile Feature

Das Profile Feature bietet jedem im Dienst angemeldeten Benutzer die Möglichkeit, die bei der Anmeldung angegebenen eigenen Profildaten zu bearbeiten oder das Passwort zu

ändern. Unbedingt auszufüllende Felder sind: Email Adresse, Anrede, Vorname, Nachname, Straße, Postleitzahl, Stadt, Land, Frage für den Fall der Anforderung des eigenen Passwortes und die dazugehörige Antwort. Optional können Titel, Initiale, Institut, Abteilung, Telefon und Faxnummer angegeben werden. Zur Änderung des Passwortes muss das Alte einmal und das Neue zur Vermeidung von Tippfehlern zweimal angegeben werden.

2.3.9 Report Feature

Im "Report Management" werden diverse Features, die die Reports betreffen, verwaltet. Dazu gehören das Einreichen und Löschen von Reports, das Lesen der eigenen bzw. der anderen Reports, das Bearbeiten eines (Final-) Reports sowie das Ansehen aller Sub-Reports.

2.3.10 Role Feature

Der OCS stellt die Möglichkeit bereit Rollen zu erstellen. So wird jedem User eine oder mehrere Rollen zugewiesen. Diese Rollen werden dann verschiedene Rechte zugewiesen. Die Rollen im jetzigen OCS sind PC Chair, PC Member, Administrator und Autor. Rollen zu erstellen oder deren Rechte zu modifizieren ist im OCS nur dem PC Chair und dem Administrator erlaubt.

2.3.11 Setup Feature

Das Setup Feature ermöglicht die Konfiguration des OCS. Die Konfigurationsmöglichkeiten reichen vom Aussehen des Dienstes, über die Rollenkonfiguration, bis hin zur Verwaltung des Programm Committees. Hier eine vollständige Liste:

1. Look & Feel of the Conference Service

- the Navigationbar
- the Content Page
- the Sponsors
- the Contact Addresses
- the Email Templates
- Preview of the Look & Feel

2. Papers and Reports

- the article-specific pages
- the report-specific pages

3. Deadlines of the Conference

- setup the conference deadlines

4. Configuration of Roles

- PC Member
- PC Chair
- Author

5. Setup of the Committee

- setup the Program Committee

6. Finish the Setup

- view the finish setup page

2.3.12 Staff Feature

Der PC Chair arbeitet in der Regel nur mit einem kleinen Teil der im Dienst angemeldeten PC Member zusammen. Damit beim Delegieren von Artikelbegutachtungen nur die PC Member angezeigt werden, die für den PC Chair relevant sind, kann der PC Chair mittels des Staff Feature einen eigenen Mitarbeiterstab als Untermenge der im Dienst angemeldeten Benutzer definieren. Es können dazu bereits angemeldete Benutzer ausgewählt oder neue Benutzer angemeldet und zum Stab hinzugefügt werden.

2.3.13 Tasklist Feature

Das Tasklist Feature stellt dem User optional zwei verschiedene Tasklisten zur Verfügung. Die Tasklist ermöglicht dem PC Member eine Übersicht über die zu erledigenden Aufgaben und bietet somit einen Leitfaden, an dem sich der PC Member während des Bewertungsprozesses eines Artikels orientieren kann.

2.3.14 User Feature

Der PC Chair ist für die Verwaltung aller Benutzer des Dienstes zuständig. Nach dem Einloggen und Klicken des „Users“ Button ist ein Überblick der Benutzer einzusehen, der nach den Nachnamen entsprechend alphabetisch organisiert ist. Das Klicken auf einen Buchstaben führt zum Anzeigen der entsprechenden zugelassenen Benutzer.

Es gibt zwei Verfahren, mit denen der PC Chair einen neuen Benutzer erstellen kann:

- Gastbenutzer, d.h. in einer verkürzten Form (ohne Profil).

- in einer kompletten Form, also mit Benutzerprofil.

Der PC Chair kann einem Benutzer Rollen zuweisen oder entziehen. Außerdem kann er einen Benutzer sperren, ein Benutzerprofil modifizieren, ein Benutzerpasswort ändern oder einen Benutzer aus der Datenbank entfernen.

2.4 Technische Konzepte

2.4.1 Client-Server-Architektur

Als Client-Server¹-Architektur bezeichnet man eine arbeitsteilige Zusammenarbeit zweier Computer oder Programme in einem Computer-Netzwerk. Es ist ein verteiltes Computer-System, in dem mehrere Clients und mindestens ein Server mitwirken. Der Server liefert Daten und Dienste für die angeschlossenen Clients. Clients stellen Anfragen an den Server und dieser sendet das Ergebnis der Anfrage zurück an die Clients. Die meisten Computer-Netze, insbesondere das Internet, sind nach diesem Prinzip aufgebaut. Der Client bildet die Schnittstelle zum Benutzer, sorgt für die Darstellung der vom Server gelieferten Daten. In einer solchen Architektur werden Daten zentral gespeichert und können von verschiedenen Stellen abgerufen werden.

Die Begriffe Client und Server bezeichnen nicht nur die jeweiligen Computer, sondern beziehen sich auf entsprechende Programme. Die bekanntesten Beispiele für Client-Programme sind:

- Internet Browser für das World Wide Web (WWW)
- E-Mail Programme

Bekannt Server-Programme sind:

- HTTP Server (Hypertext Transfer Protocol) zum Bereitstellen von HTML (Hypertext Markup Language)
- POP Server zum Bereitstellen von E-Mail Diensten

Die Verarbeitung der Daten kann unterschiedlich verteilt werden. Die beteiligten Programme können entweder auf dem Rechner des Nutzers (client-seitig) oder auf dem zentralen Netzrechner (server-seitig) ausgeführt werden. Im World Wide Web werden zum Beispiel Anfragen nach einem Java-Applet an den Server geschickt, der wiederum den entsprechenden Code an den Client zurückschickt. Erst auf dem Client wird der Code ausgeführt, was den Server entlastet. Anders ist es zum Beispiel bei Skriptsprachen wie die Java Server Pages (JSP). Der Client schickt die Anfrage (evtl. mit abgesendeten Formulardaten)

¹client (engl.) = Kunde, server (engl.) = Diener

an den Server, der die Anfrage annimmt. Dort wird dann die gesamte Verarbeitung durchgeführt und nur das Ergebnis (HTML-Code) an den Client zurückgeschickt. Beim OCS wird ebenfalls die Verarbeitung der Clientanfrage auf dem Server durchgeführt. Ein Beispiel für einen Ablauf zwischen Client und Server sieht man in Abbildung 2.2.

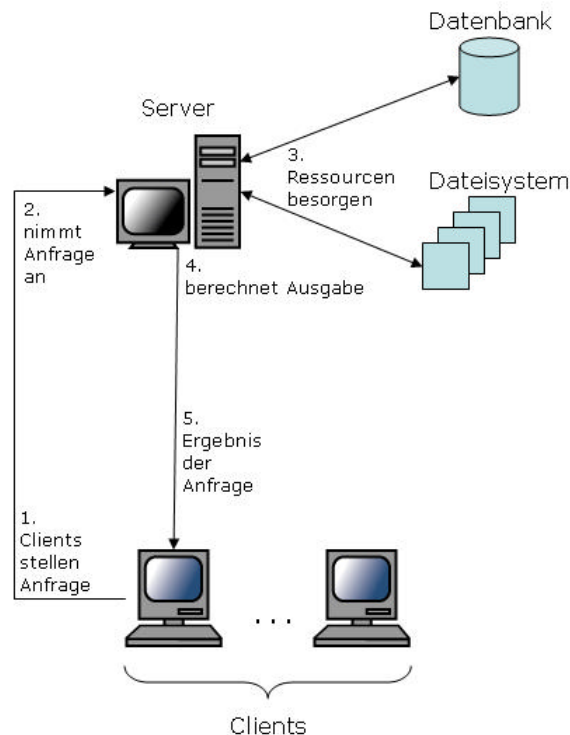


Abbildung 2.2: Ablauf zwischen Client und Server

2.4.2 Verteiltes System

Als verteiltes System bezeichnet man ein System, das aus mehreren, auf vernetzten Computern installierten Komponenten besteht, aber seinen Benutzern als einheitliches zusammenhängendes System gegenüber tritt. Ein Beispiel hierfür ist der OCS mit angebundener PostgreSQL Datenbank, einem Mail-Server, einem CVS-Server und einem News-Server. Zur Speicherung aller anfallenden Daten wird die Datenbank benötigt, die Benutzer können im System an einem News-Forum zur Diskussion teilnehmen und werden über wichtige Ereignisse per E-Mail informiert. Dies alles geschieht für den Benutzer völlig transparent, er benötigt lediglich einen Internet Browser, um auf das verteilte System zugreifen zu können. Ein großer Vorteil hierbei ist die allgemeine Verfügbarkeit des Systems. So kann der Benutzer von überall auf der Welt zugreifen und sieht stets seine vertraute Umgebung.

Kapitel 3

Entwicklung und Modifikation

Das OCS wurde mittels des ABCs entwickelt. Der zugehörige Workflow wird als Graph modelliert, dessen Hauptkomponenten die SIBs darstellen. Die SIBs werden untereinander mit Hilfe von Kanten verbunden. Ein SIB wird durch seine Spezifikation und durch die Implementierung in Java beschrieben. Um die Transparenz des Dienstes zu erhöhen werden Features in Makros zusammengefasst. Diese Vorgehensweise der Makrofizierung wird in dem nächsten Kapitel erläutert. Nach den Grundlagen, die die Teilnehmer der Projektgruppe erarbeiten mussten, wird die eigentliche Arbeit beschrieben. Zuerst wurden Einzelthemen, die Erweiterungen des Dienstes darstellten, erläutert. Durch das Bearbeiten der Einzelkurzthemen konnten sich die Projektgruppenteilnehmer mit dem OCS und ABC vertraut machen und Erfahrungen sammeln. Diese Erfahrungen wurden für die Bearbeitung der Gruppenkurzthemen, die im letzten Unterkapitel beschrieben wurden, benötigt.

3.1 Makrofizieren

3.1.1 Vorteile des Makrokonzepts

Makros kapseln Graphsektionen in wiederverwendbaren Paketen. Dies erleichtert die Wiederverwendung von SIB-Sequenzen erheblich. Zudem bekommen grosse Dienste mit sehr vielen SIBs durch die Verwendung von Makros mehr Übersicht. So hat es der Programmierer leichter den Dienst zu erweitern oder ihn nach Fehlern zu durchsuchen. Wenn ein Makro in einem SLG eingesetzt wird, so ist es in dem Graphen in zwei Formen sichtbar. Einmal abstrakt als Knoten (siehe Abbildung 3.1) und einmal als editierbarer Teilgraph.

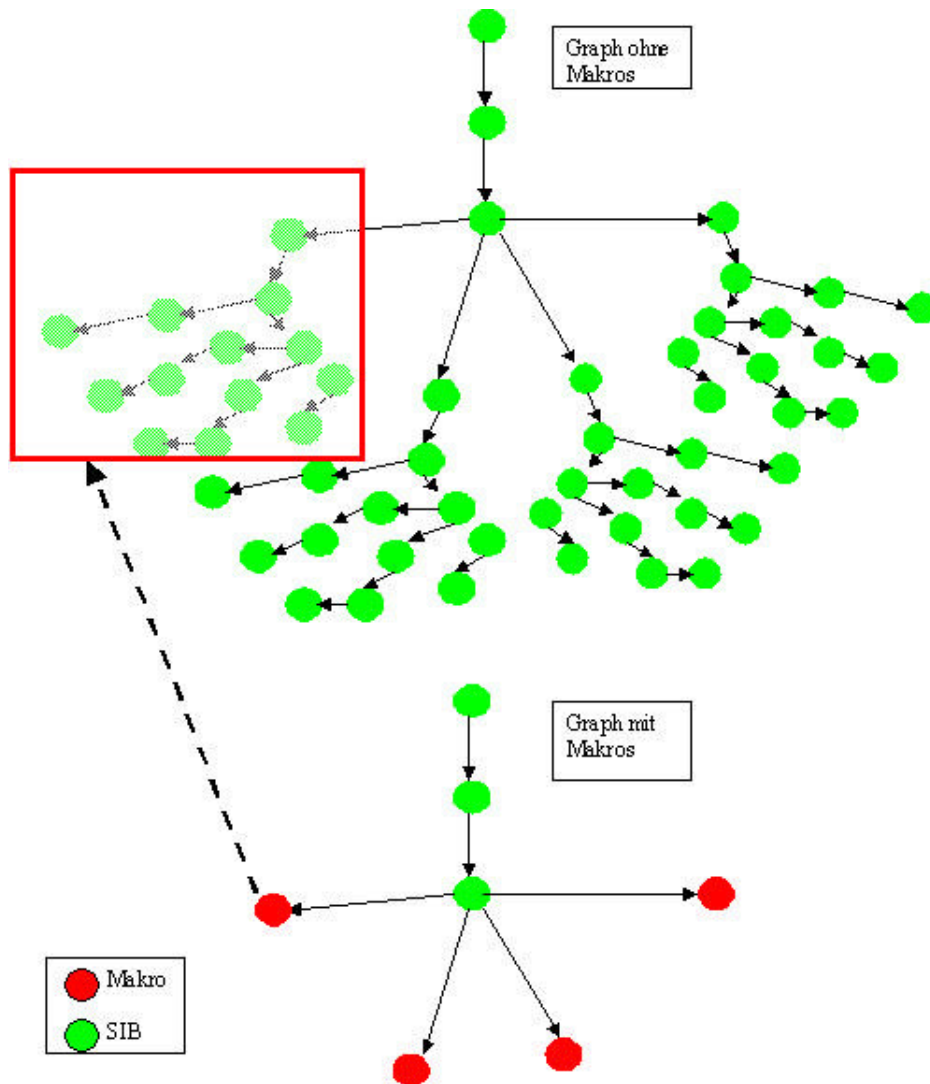


Abbildung 3.1: Makrofizierter Graph

3.1.2 Modifikation des Dienstes

3.1.2.1 Voraussetzung

Um den Dienst durch Makros vereinfachen zu können, müssen zunächst geeignete, zusammenhängende Graphsegmente gefunden werden. Der OCS hat durch sein Featurekonzept gut voneinander separierbare Workflows.

3.1.2.2 Vorgehen

Beim Makrofizieren wird der Teilgraph unter "makroname.smg" gespeichert. Da das Makro eingehende und ausgehende Branches hat, müssen diese exportiert werden. Der Dienst wurde in der Einzelkurzthemenphase "makrofiziert". Jedes Feature wurde von den Projektgruppenteilnehmern in ein eigenes Makro überführt. Dieses hatte für die weitere Arbeit am Graphen den Vorteil der verbesserten Übersichtlichkeit und erleichterte später die Zusammenführung der Teillösungen.

3.2 Einzelkurzthemen

3.2.1 Article Feature

Beschreibung der Aufgabe

Die im OCS angemeldeten Autoren wurden nicht mittels einer Email vom Dienst über den Erfolg oder Misserfolg ihrer Artikeleinreichung informiert. Das bedeutet, dass nur PC Chairs eine Bestätigungsemail im Fall einer erfolgreichen Artikeleinreichung bekommen. Deswegen wurde der Artikel Submission Workflow um zusätzliche Checks (ArticleStatus Check) und das Versenden der Bestätigungsemail an den Autor erweitert. Ist eine Artikeleinreichung nicht erfolgreich, sollen zusätzlich alle PC Chairs und der Autor selbst per Email informiert werden.

Modellierungsphase

Das Use Case (siehe Abbildung 3.2) stellt die Handlungsoptionen des Autors dar, die durch die Lösung der Aufgabe realisiert werden sollen. Im nachfolgenden Abschnitt wird erläutert, wie der konkrete Workflow aussieht. Das Aktivitätsdiagramm aus der Abbildung 3.3 beschreibt den Workflow der Artikel Submission im Erfolgsfall. Das bedeutet, nachdem der Dienst die Anfrage des Autors empfängt, prüft der Dienst anschließend die Zustände des Submissionsformulars und Artikels. Alle PC Chairs und der Autor selbst werden eine Bestätigungsemail erhalten, falls der Artikel erfolgreich eingereicht wurde (Zustand "submitted"). Andererseits ist es sehr schwierig, nach der Artikeleinreichung

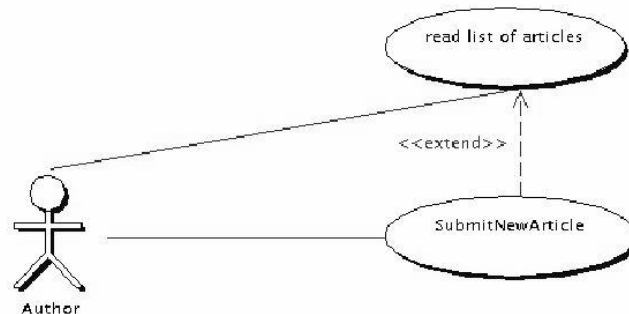


Abbildung 3.2: Use Case „Submit New Article“

den Zustand “created” im Dienst sofort zu überprüfen, falls der Artikel nicht erfolgreich eingereicht werden konnte. In diesem Fall wird ein anderer Workflow im Gegensatz zu oben verwendet (siehe Abbildung 3.4).

In der Konfigurationsdatei des Tomcat Services (“web.xml”) wird eine neue Eigenschaft “ArticlesStateTimeWatcherInterval” mit einem Ganzzahlenwert definiert. Dieser Wert stellt die Sekunden dar, nach denen ein serviceinterner Thread gestartet wird. Dieser Thread überprüft die Zustände aller Artikel in der Artikelliste. Wird ein Artikel mit dem Zustand “created” gefunden, wird der Autor dieses Artikels und alle PC Chairs mittels Email über die nicht erfolgreiche Einreichung dieses Artikels informiert.

Implementierungsphase

Diese Phase besteht aus zwei Teilen, der Modifikation des Graphen und der Implementierung der neuen SIBs. Der Erfolgs- und Misserfolgsfall wird in den nächsten zwei Abschnitten getrennt erläutert.

Verschickung einer Email bei erfolgreicher Artikeleinreichung

Zur Realisierung dieser Funktion soll zunächst das Makro “Article” modifiziert werden. Der Startknoten des Graphen ist “ClearTempSessionValues”.

Da die Inhalte der Email für die PC Chairs und für den Autor verschieden sind, unterscheiden sich auch die Workflows der Emailerstellung. Wie in Abbildung 3.5 gezeigt wird, hat das alte Makro die Email an die PC Chairs schon erstellen können. Die Emailadressen aller PC Chairs werden durch den SIB “LoopOverEmailStringArray” ausgelesen. Danach werden die Emails durch die Standard Email-SIBs: “CreateNewMimeMessage”, “AddEmailHeader”, “CreateBody” und “SendMimeMessage” zusammengestellt und verschickt.

In Abbildung 3.6 werden zwei neue SIBs für die Erstellung der Mail an den Autor in das Makro eingefügt. Die autor-spezifischen Emailinhalte werden durch den SIB “PrepareSubmitArticleAuthorMailParameter” geholt und in den Call Context geschrieben. Der

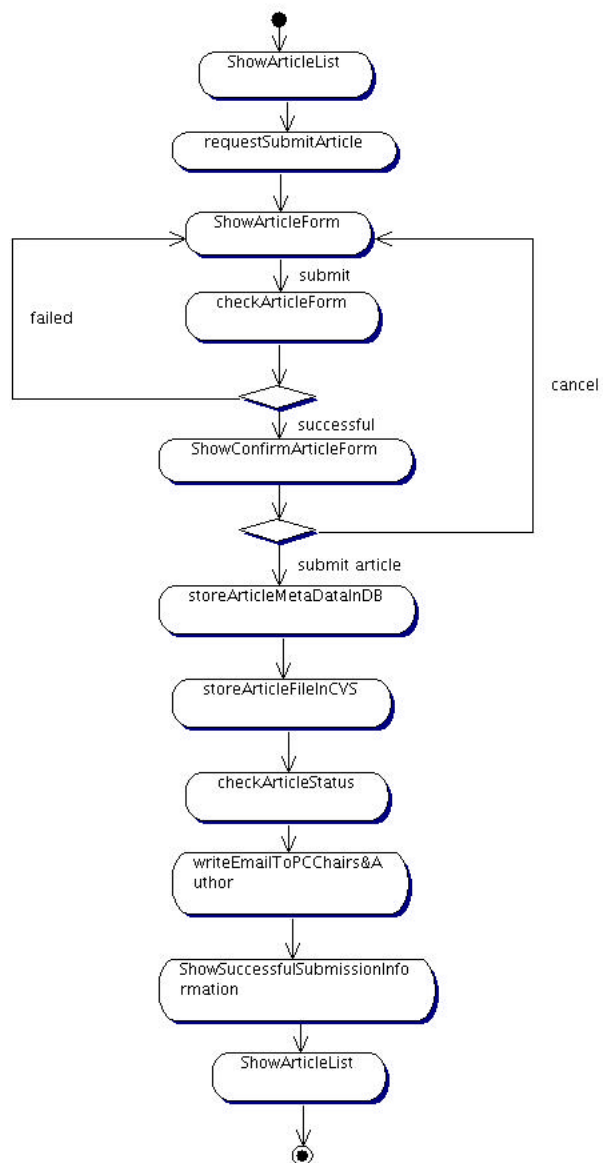


Abbildung 3.3: Aktivitätsdiagramm „Submit New Article 1“

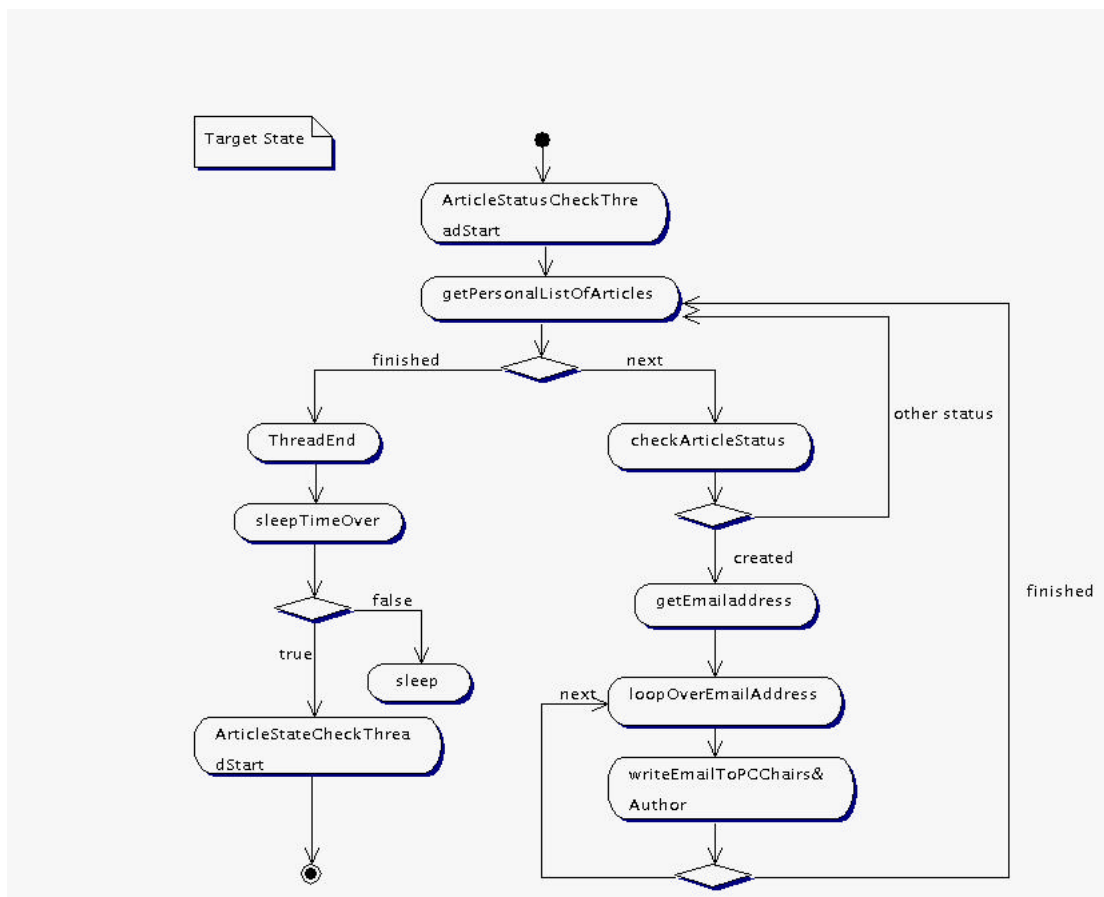


Abbildung 3.4: Activity Chart „Submit New Article 2“

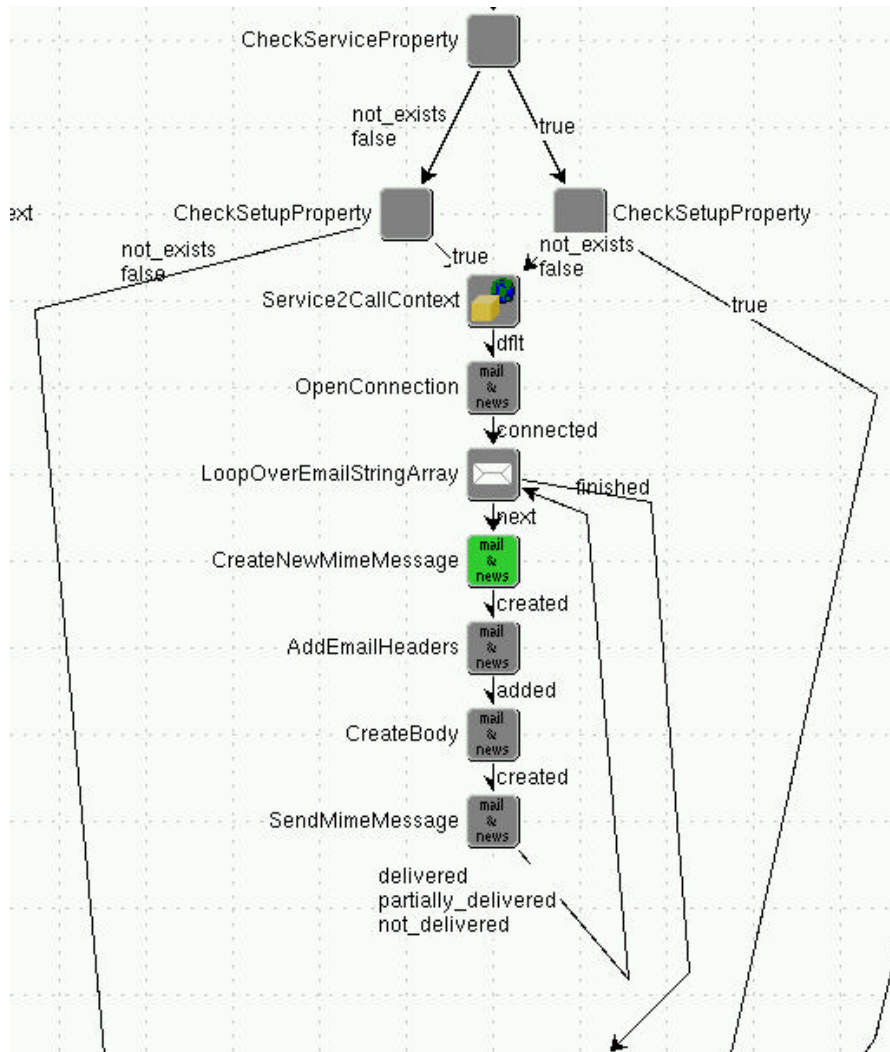


Abbildung 3.5: Makro „Old Article Macro“

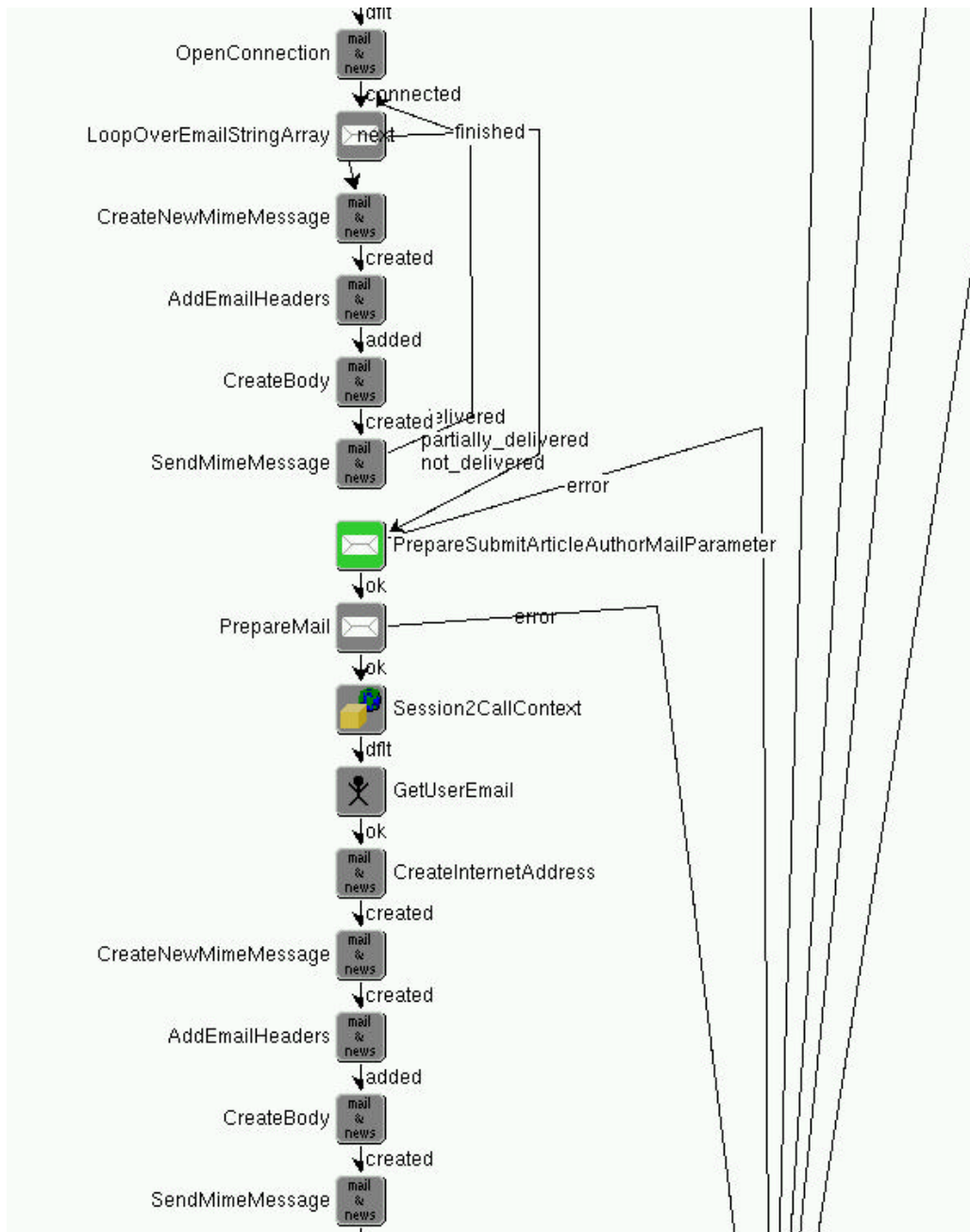


Abbildung 3.6: Makro „New Article Macro“

SIB “GetUserEmail” holt den String mit der Emailadresse des Autors, der wiederum im SIB “CreateInternetAddress” benutzt wird, damit eine wirkliche Emailadresse erzeugt werden kann. Anschließend wird die Email durch die vier oben genannten Standard Email-SIBs an den Autor gesendet.

Verschickung einer Email bei nicht erfolgreicher Artikeleinreichung

Hierzu muss eine Änderung außerhalb des Makros vorgenommen werden (siehe Abbildung 3.7 und 3.8).

Auf Grund der vorherigen, in der Datei web.xml definierten Eigenschaft “ArticlesStateTi-

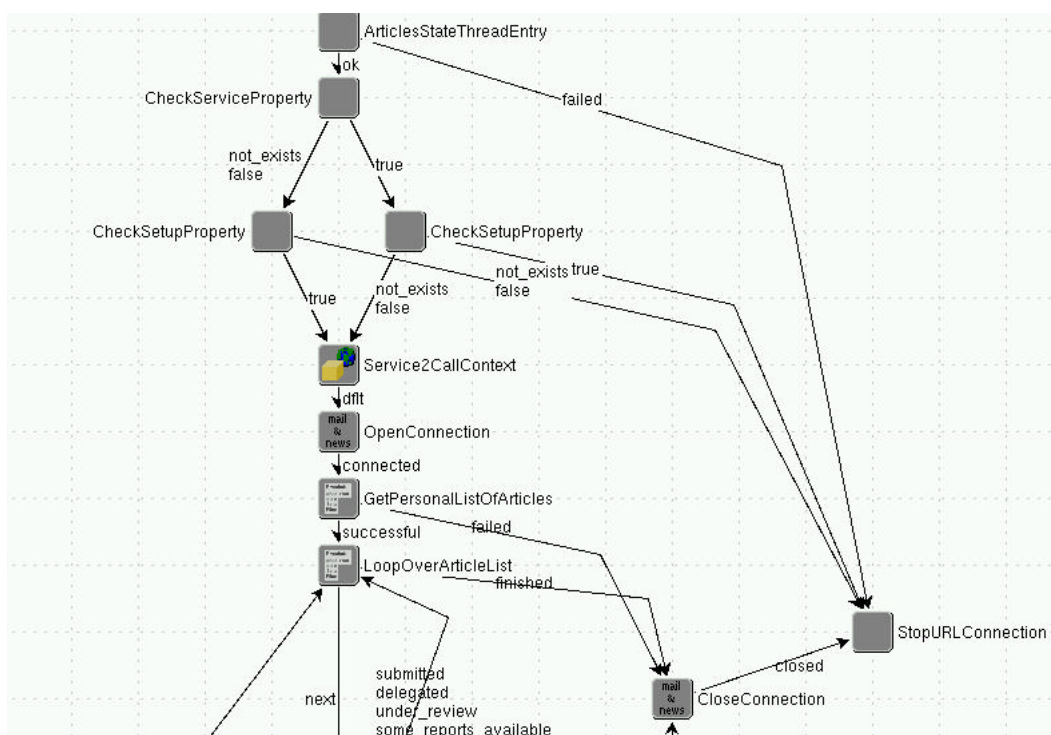


Abbildung 3.7: Conference Graph „successful submission 1“

meWatcherInterval” kann ein Thread “ExceededArticlesStateThread” zur Initialisierung des OCS gestartet werden. Der Entry SIB “ArticlesStateThreadEntry” wird mit dem Parameterwert “ArticlesStateThreadEntry” ausgeführt. Der SIB “GetPersonalListOfArticles” holt eine Liste mit Artikeln aus dem Call Context. Der SIB “LoopOverArticleList” prüft jeden Artikel. Ein neues SIB “PrepareArticlesStateDeadlineMailParameter” holt alle benötigten Komponenten der Email, wie z.B. “subject”, “from”, “to” und schreibt sie in den Call Context. Am Ende kann die Email wieder mittels der vier Standard Email-SIBs an den Autor verschickt werden.

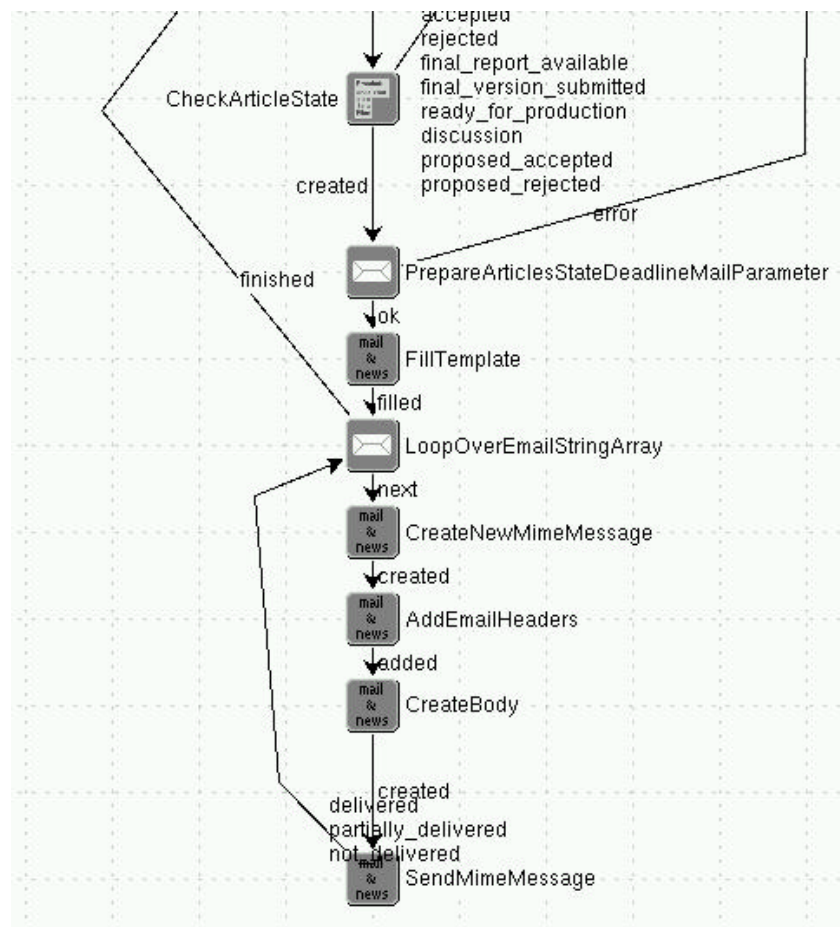


Abbildung 3.8: Conference Graph „successful submission 2“

SIB „PrepareSubmitArticleAuthorMailParameter“ : Dieser SIB liefert die benötigten Emailparameter und eine Template, die über den SIB Parameter "templatename" angegeben werden kann. Dieser SIB ist allerdings nur für Emails an Autoren gedacht, für Emails an PC Chairs ist das SIB "PrepareSubmitArticleTasklistEntryMailParameter" zu verwenden.

SIB „GetUserEmail“ : Dieser SIB liest einen String mit der Emailadresse des Benutzers aus dem Call Context, dessen Name über den Parameter "object_key" spezifiziert werden kann. Dieser String wird wiederum unter dem im Parameter "email_key" angegebenen Namen im Call Context abgelegt.

SIB „PrepareArticlesStateDeadlineMailParameter“: Wie das SIB "PrepareSubmitArticleAuthorMailParameter" realisiert dieser SIB die selbe Funktion, aber nur für den Fall einer nicht erfolgreichen Artikeleinreichung.

Testphase

Alle neuen Funktionen wurden erfolgreich implementiert und wurden auf korrekte Implementation getestet.

3.2.2 Delegation Feature

Beschreibung der Aufgabe

1. Aufgabe: Einbau einer Bestätigungsseite, welche den Benutzer auffordert seine Verteilung der Begutachtungsaufträge zu bestätigen. Er hat somit noch die Möglichkeit evtl. die Verteilung zu modifizieren, bevor die Begutachter durch den Dienst informiert werden.
2. Aufgabe: Jeder Eintrag in der Matrix soll einen zusätzlichen Link erhalten. Dieser Link öffnet ein neues Browserfenster, das alle Reports des jeweiligen Artikels anzeigt. Die Realisierung soll ähnlich der Evaluationmatrix sein.

Modellierungsphase

Lösung zur Aufgabe 1

- Activity Chart (siehe Abbildung 3.9): Aus der ShowArticleList wählt der Benutzer (PC Chair) einen Artikel zum Delegieren, wonach die Seite "ShowDelegateArticle" angezeigt wird. Der Benutzer kann hier einen oder mehrere Begutachter wählen. Mit dem Button "delegate" wird die Seite "confirm delegation" angezeigt, sie enthält Informationen über die Änderungen, die durch die Durchführung des Delegierens verursacht werden. Der Benutzer hat die Möglichkeit dies durch den Button "submit"

zu bestätigen. In diesem Fall werden die angeforderten Änderungen durchgeführt. Der Benutzer kann mit dem Button "cancel" das Delegieren abbrechen.

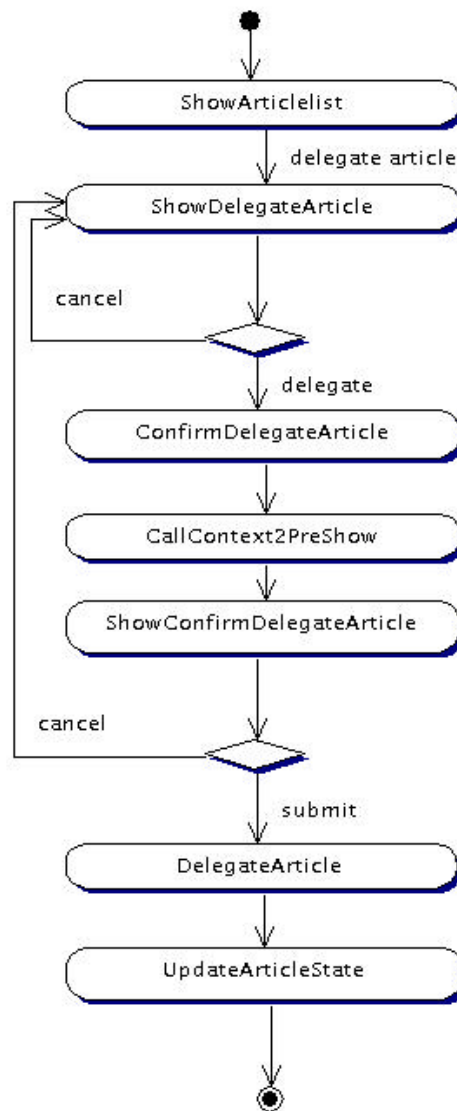


Abbildung 3.9: Activity Chart "Delegation Feature"

Implementierungsphase

- SIBs: Neue SIBs "ConfirmDelegateArticle" und "ShowConfirmDelegateArticle" wurden implementiert.
- wml-Dateien: Die Datei "confirmdelegatearticle-cont.wml" wurde erstellt.

- Makro: Um die Lösung zu realisieren, wurde das Makro "article" bearbeitet. In diesem Makro wurden folgende Änderungen durchgeführt: Der SIB "DelegateArticle" erreicht den SIB "ConfirmDelegateArticle" durch den Branch "delegate". Der SIB "ConfirmDelegateArticle" erreicht den SIB "CallContext2PreShow" durch den Branch "ok", dieser erreicht "ShowConfirmDelegateArticle" durch "dflt".
- Test: Um die notwendigen Tests durchzuführen, wurde zuerst ein Artikel von einem Autor eingereicht. Der PC Chair versuchte den Artikel an einen Begutachter zu delegieren. Dabei wurde er aufgefordert sein Delegieren zu bestätigen. Die Bestätigung erfolgte durch den PC Chair mit Hilfe des Buttons "submit", gefolgt von den entsprechenden Aktualisierungen durch den OCS. Anschliessend wurde der Begutachter vom Dienst über den neuen Artikelstand informiert. In einem zweiten Testversuch führte der PC Chair die geforderte Bestätigung nicht durch. So konnten durch den Button "cancel" die Eingaben des PC Chairs für das Delegieren nochmals aufgerufen und von ihm korrigiert werden. Dasselbe Testverfahren wurde erfolgreich mit weiteren Begutachtern durchgeführt.

Lösung zur Aufgabe 2

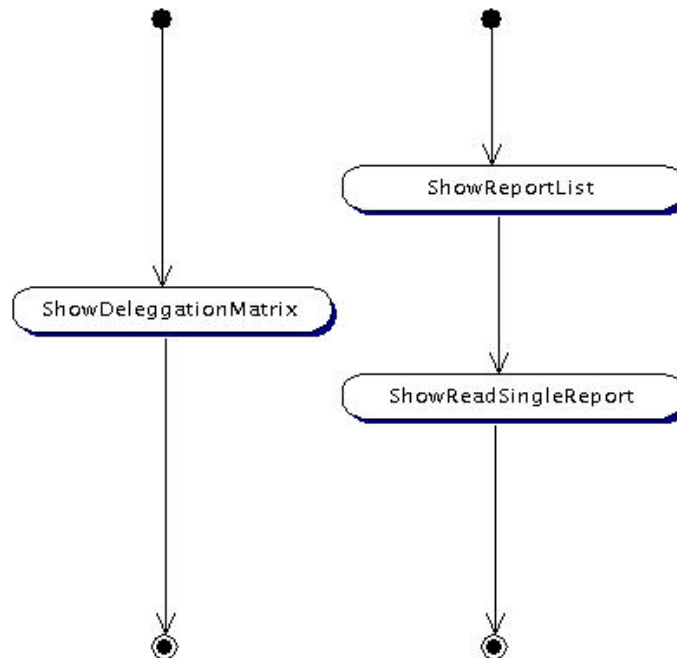


Abbildung 3.10: Activity Chart 2 "Delegation Feature"

- SIB: Der SIB "ShowDelegationMatrix" wurde angepasst. Er bekam einen neuen Branch "readarticle". Der SIB "ShowReadReportsOfArticle" wurde ohne Änderungen benutzt.
- wml-Dateien: Die Datei "delegationmatrix-cont.wml" wurde folgendermassen angepasst: Zu der ersten Spalte "ArticleID" in der Tabelle "Article" wurde zusätzlich eine weitere Spalte hinzugefügt. Diese enthält einen Eintrag "/R" als Link. Mit Hilfe des Links werden Reports zu dem entsprechenden Artikel angezeigt. Falls zu einem Artikel keine Reports verfügbar sind, wird kein entsprechender Eintrag angezeigt. (siehe Abbildung 3.10)
- Makro: Ein neues Makro "delegation" wurde erstellt. Es wird aus dem SIB "update-Nav" durch die Branches "delegation" und "subdelegation" erreicht. Dieses Makro wurde folgendermassen modifiziert: Der SIB "ShowDelegationMatrix" erreicht den SIB "ShowAllReportsOfArticle" durch den Branch "readreport".

Testphase

Nach der Durchführung diverser Testläufe wurde folgendes beobachtet: Ein Artikel, zu dem Reports existieren, erhält einen Link "/R" in der Tabelle "Article". Zu einem Artikel, zu dem keine Reports verfasst wurden, existiert kein Link in der Tabelle. Somit wurden keine Fehler in der Lösung der Aufgabe gefunden.

3.2.3 Email Feature

Beschreibung der Aufgabe

Die Aufgabe dieses Kurzthemas war das Email Feature zu erweitern. Auf der Seite "Write new Email" gab es bisher keine Möglichkeiten eine Email gezielt an Autoren, in Abhängigkeit des Artikelstatus, zu versenden. Deshalb war das Ziel der Aufgabe eine Möglichkeit bereitzustellen, um an Autoren, die Artikel eingereicht haben (Artikelstatus "submitted"), deren Artikel zurückgewiesen (Artikelstatus "rejected") oder deren Artikel akzeptiert worden sind (Artikelstatus "accepted"), eine Rund-Email versenden zu können. Dazu musste der bestehende Workflow "Write new Email" angepasst werden.

Modellierungsphase

In der Modellierungsphase wurde zuerst der bestehende Workflow analysiert (siehe Abbildung 3.11), um zu ergründen, wie er verändert werden musste. Um das Ziel der Aufgabe zu erreichen, mußte ein neuer Unterpunkt auf der Seite "Write new Email" eingefügt werden. Dieser sollte die drei oben beschriebenen Möglichkeiten zum Versenden von Emails enthalten. Vor der Durchführung der Aufgabe bestand die Möglichkeit Rund-Emails an die verschiedenen Rollen des Online Conference Service, oder an

die verschiedenen Rollen des eigenen Stabs zu versenden. Also existierte bereits eine Möglichkeit, eine Rund-Email an Autoren zu verschicken. Diese Einteilung reichte

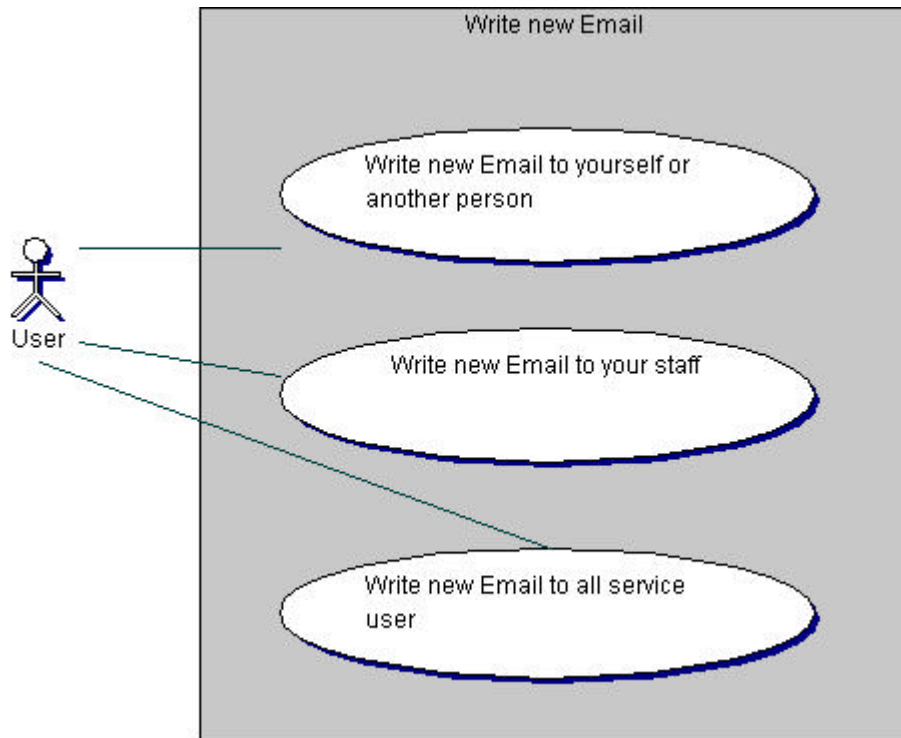


Abbildung 3.11: Use Case des Istzustands "Write new Email"

jedoch nicht aus. Die Rolle der Autoren sollte nach dem Status ihrer Artikel unterschieden werden. Deshalb musste der Unterpunkt "Email schreiben an Autoren" verfeinert werden. Die Autoren sollten nach den drei Artikelstati unterschieden werden, und zwar ob ihre Artikel sich im Status "accepted", "submitted" oder "rejected" befinden. So sollte der User mit einem Klick eine Rund-Email an bestimmte Autoren verschicken können (siehe Abbildung 3.12). Der bisherige Workflow funktionierte, wie in Abbildung 3.13 dargestellt. Der Dienst bereitete ein Email-Template für den Benutzer vor. Danach wurde das Email-Template vom Benutzer ausgefüllt und abgeschickt. Der Dienst überprüfte im Hintergrund, ob alle notwendigen Felder ausgefüllt worden waren. War dies nicht der Fall, so mußte die Emailform vom Benutzer nochmals ausgefüllt werden. Zur jeder Zeit bestand natürlich die Möglichkeit, das Schreiben der Email abubrechen. War das Template korrekt ausgefüllt, so wurde eine vorläufige Email zur Überprüfung ausgegeben. Dabei wurden die Textmakros und die Empfänger der Email eingefügt, um dem Benutzer die vollständige Email zu präsentieren. War die Email zur Zufriedenheit ausgefüllt, so konnte der User die Email abschicken. Dieser Workflow musste für die Erfüllung der Aufgabe geändert werden. Bevor die Email dem Benutzer zur Überprüfung gezeigt wurde, mussten die Emailadressen der Autoren, abhängig vom Zustand der eingereichten

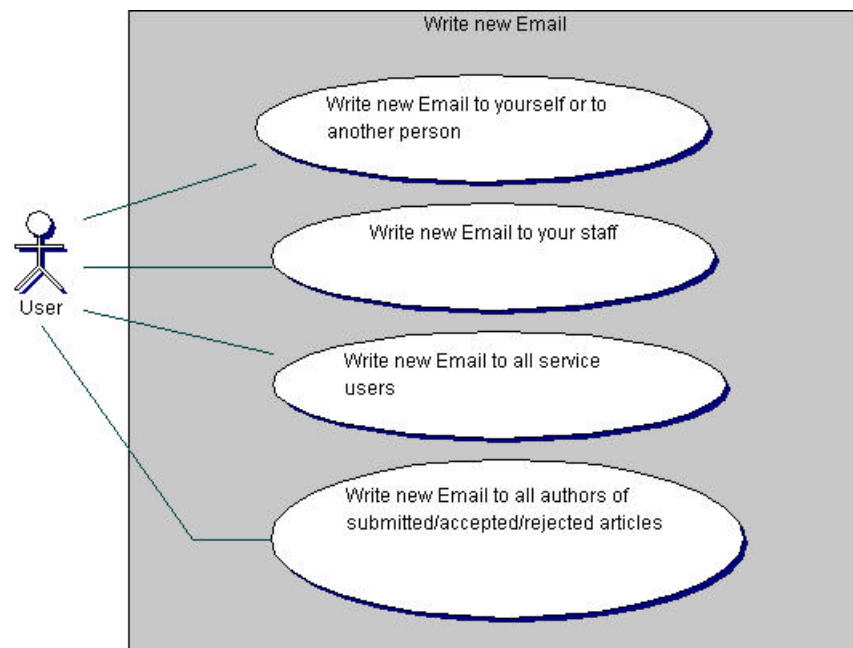


Abbildung 3.12: Use Case des Sollzustands "Write new Email"

Artikel, eingeholt werden. Dazu wurden zuerst alle Artikel aus der Datenbank ausgelesen. Es wurde der Status der Artikel überprüft und je nach Anforderung die Artikel in eine neue Artikelliste kopiert. Aus dieser Artikelliste, die mit Artikeln mit dem Zustand "accepted", "rejected" oder "submitted" gefüllt war, wurden die Emailadressen der Verfasser ausgelesen. Die Empfängerliste der Email wurde mit diesen Emailadressen zusätzlich gefüllt. Anschließend wurde der Emailentwurf dem Benutzer zur Überprüfung angezeigt (siehe Abbildung 3.14).

Implementierungsphase

Um das Ziel der Aufgabe zu erfüllen, mußten einige wml-Seiten und neue java-Klassen erstellt werden. Zuerst wurde die wml-Seite "composeemail-editor-cont.wml" bearbeitet. Die einzige Änderung war die Erweiterung um den Unterpunkt "Write Email to Authors". Es wurden die drei geforderten Möglichkeiten "submitted", "accepted" und "rejected" hinzugefügt. Wegen der Entscheidung Radiobuttons für die Auswahl einzusetzen, wurde eine vierte Wahlmöglichkeit "none", also keinem der Autoren eine Email zu schreiben, hinzugefügt. Jeder dieser Möglichkeiten wurde ein String zugeordnet, so konnte im Graphen mit Hilfe einfacher Compare-SIBs verglichen werden, welche Wahl der Benutzer getroffen hatte. Anschliessend wurde der gewählte Artikelzustand als Parameter an einen neu geschriebenen SIB "GetArticlesOf" übergeben. Dieser SIB filtert alle Artikel nach dem als Parameter eingestellten Artikelzustand. Als Parameter sind alle Artikelzustände des OCSs einstellbar. Als nächstes wird eine Artikelliste in den

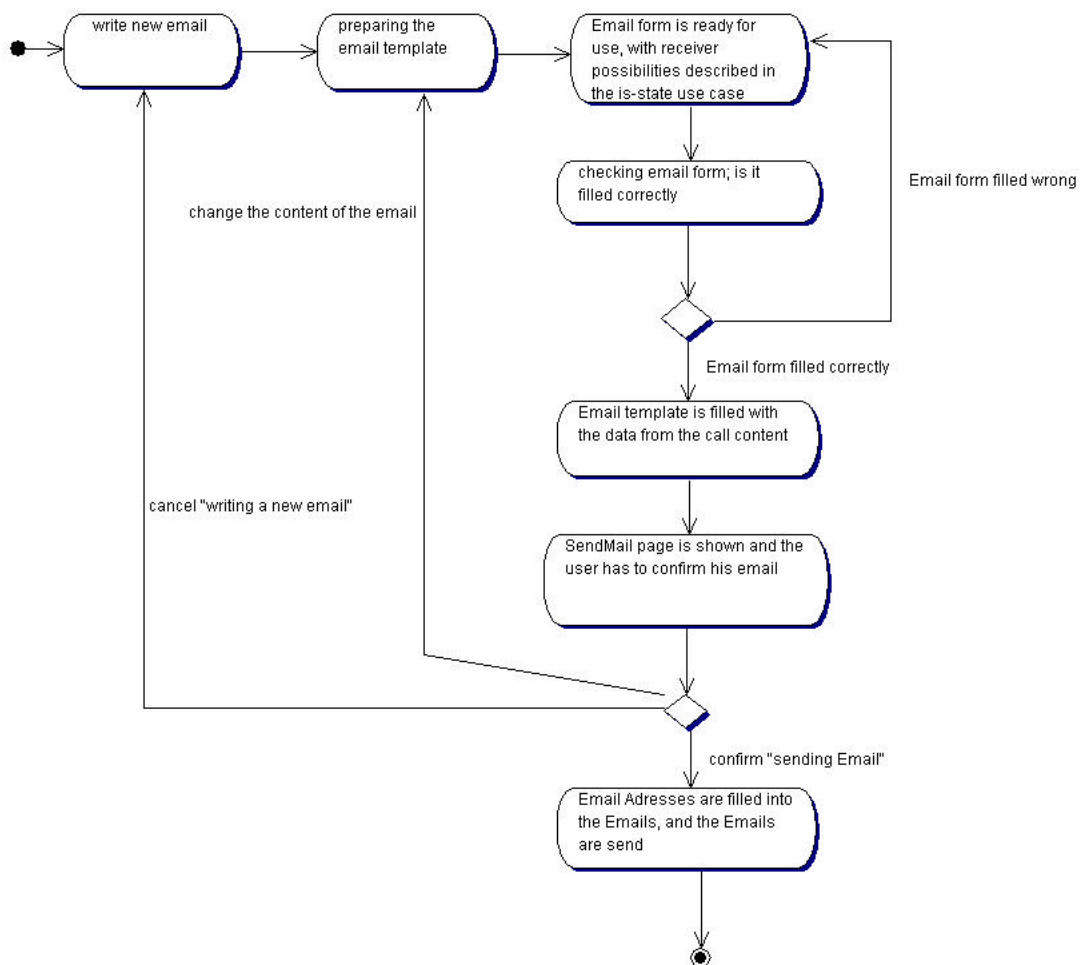


Abbildung 3.13: Activity Chart des Istzustands "Write new Email"

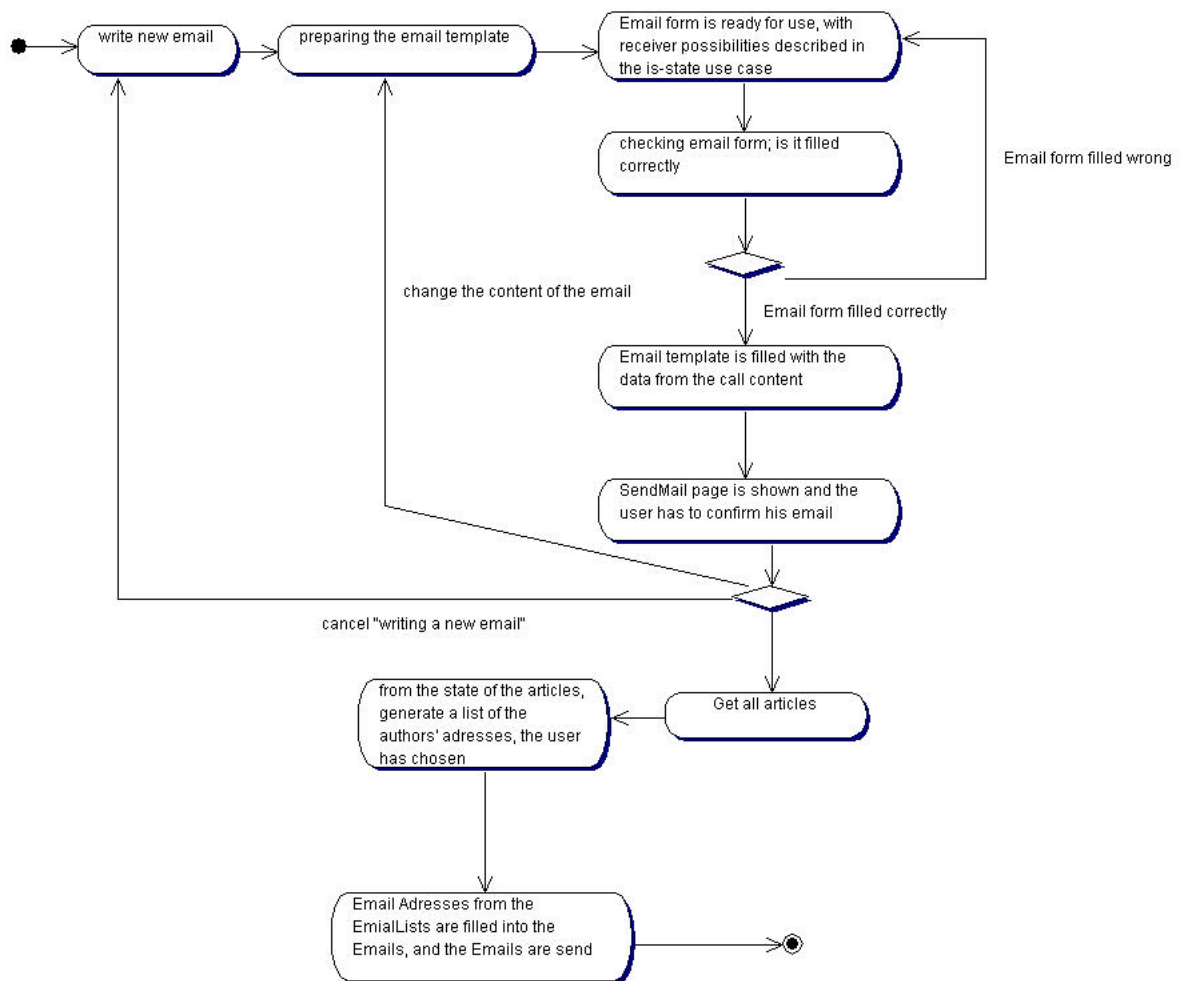


Abbildung 3.14: Activity Chart des Sollzustands "Write new Email"

Call Context geschrieben, deren Artikel dem eingestellten Artikelzustand entsprechen. Dieser SIB wird drei Mal in dem neuen Workflow eingesetzt, abhängig von den drei Wahlmöglichkeiten des Benutzers. Der nächste SIB "GetEmailFromArticles" generiert aus einer Artikelliste einen Vektor, der alle Emailadressen der Autoren aus der Artikelliste enthält. Somit wurden alle Informationen gesammelt, die für die Erfüllung der Aufgabe gebraucht wurden. In dem bereits existierenden, nachfolgenden SIB "CheckEmailForm" musste nur der neue Emailvektor hinzugefügt, sowie die Emailadressen aus dem Vektor in die Empfängerliste geschrieben werden. Da der Benutzer nach der Bestätigungsseite noch Änderungen an der Email vornehmen darf, mussten alle benötigten Variablen über den Workflow zwischengespeichert werden und bei Bedarf korrekt gesetzt werden.

Testphase

Für den Test wurden neue Benutzer, ein paar Autoren, ein PC-Member und ein PC Chair angelegt. Zuerst wurde getestet, ob der Workflow korrekt funktioniert, wenn noch keine Artikel eingereicht wurden. In der Empfängerliste befanden sich, wie erwartet, keine Autoren. Nebenbei wurde immer getestet, ob die gewählten Parameter des Benutzers auf der Webseite richtig gesetzt wurden. Anschliessend wurden Artikel eingereicht. Diese wurden zufällig auf verschiedene Zustände gesetzt. Bei den Tests befanden sich immer die richtigen Autoren in der Empfängerliste.

Mit der Erfüllung dieser Aufgabe wurde das Emailfeature des OCS komfortabler gestaltet. Die Erweiterung um den Unterpunkt "Write new Email to Authors", unterschieden nach den jeweiligen Artikelzuständen, erleichtert die Arbeit. Somit können alle Autoren mit einer Email erreicht werden, die beispielsweise zurückgewiesene Artikel eingereicht haben.

3.2.4 Evaluation Feature

Beschreibung der Aufgabe

Mit Hilfe der Evaluationmatrix, kann der PC Chair den Status eines Artikels setzen, ohne dass er vorher durch eine Infoseite seine Aktion bestätigen muss. Das Ziel der Aufgabe ist ein Workflow (siehe Abbildungen 3.15 und 3.16), mit einer zusätzlichen Bestätigungsseite zu entwerfen. Diese muss alle Informationen über den gewählten Status enthalten. Ausserdem soll die Möglichkeit bestehen, gleichzeitig mehrere Artikel auszuwählen und sie auf einen gewählten Status zu setzen.

Modellierungsphase

Auf der Evaluationmatrix-Seite, kann der PC Chair mehrere Artikel auswählen und diese auf den Status open Forum, close Forum...etc.. setzen. Nachdem der Status selektiert wurde, erscheint eine Bestätigungsseite. Betätigt der PC Chair den Button "OK", so wird

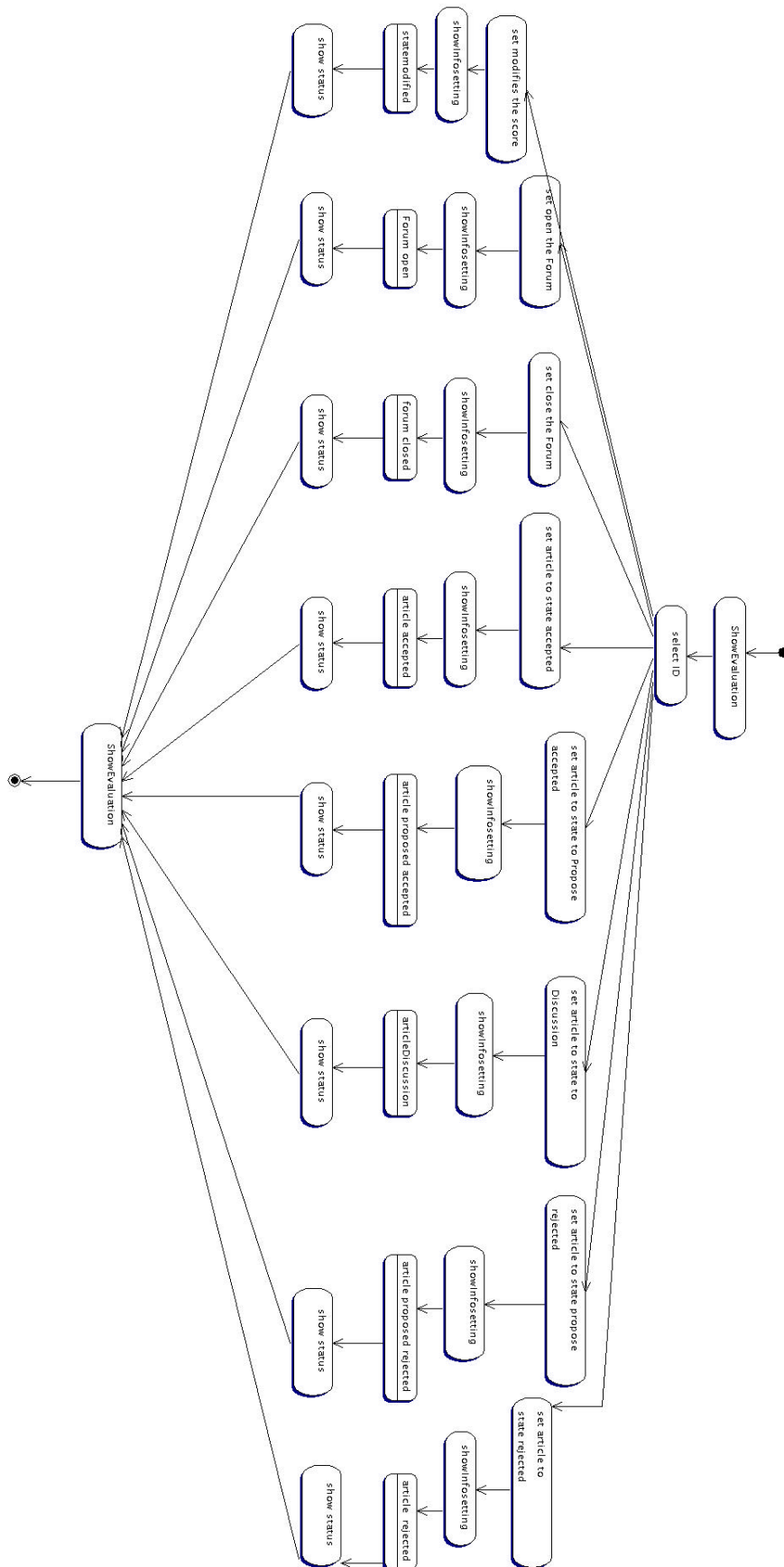


Abbildung 3.15: Activity Chart "Evaluation Feature"

der Status gesetzt und die Acknowledgment-Seite wird aufgerufen. Danach kann der PC Chair zurück auf die Evaluation-Matrix Seite kehren (siehe Abbildung 3.15).

Implementierungsphase

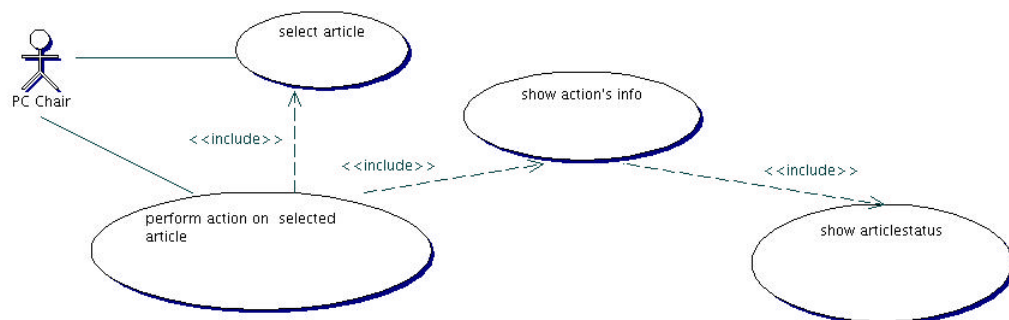


Abbildung 3.16: Use Case "Evaluation Feature"

Während der Arbeit wurden folgende SIBs entwickelt:

LoopOverarticleId, PrepareEvaluationMatrixEntryArray, CachEvalRequest.

SIB „LoopOverarticleId“ : Der SIB enthält als Parameter ein Array von ArticleIds. Er ermöglicht das Setzen eines Artikels, deren ArticleId im Array enthalten ist, auf den gewählten Status.

SIB „PrepareEvaluationMatrixEntryArray“ : Der SIB besitzt als Parameter ein Array von ArticleIds, ein Array von Articles und ein Array von EvaluationMatrixEntries. Die Funktion des SIBs ist ein Array von ArticleIds aus dem Call Context zu holen und ein EvaluationMatrixEntry-Objekt zu erzeugen. Bei dieser Aktion wird auch das Average eines Artikels berechnet.

SIB „CachEvalRequest“ : Mit Hilfe des SIBs kann das ScoreConfidence-Objekt aus dem Call Context gespeichert werden.

Folgende wml-Dateien wurden verändert:

evaluation_modify_matrix-cont.html, evaluation-acknow-cont.html. Die Seite showinfo_all.html wurde neu hinzugefügt.

Evaluation_modify-matrix-cont.html (siehe Abbildung 3.18) : Es wurden Checkboxes hinzugefügt, um mehr als nur ein Artikel auszuwählen. Eine zusätzliche Tabelle für die Article-Action wurde ebenfalls hinzugefügt. Im Vergleich zur alten Seite (siehe Abbildung 3.17) ist es nun möglich eine Aktion gleichzeitig auf mehreren Artikeln durchzuführen.

The screenshot shows a web browser window with the URL `http://sttt.cs.uni-dortmund.de/pg440ocs/servlet/Conference/NavbarEntry/evaluation`. The page content includes:

- A navigation menu on the left with buttons for Setup, Bidding, Articles, Delegation, Sub Delegation, Reports, Evaluation, My Staff, Conflicts, Email, Users, Roles, Profile, Preferences, Logout, and FAQ.
- A main content area with a 'PC Chair' button and a 'PC Forum' button.
- A table showing article scores and actions:

Art.ID	Score						Article Actions	Status	Type	Art.ID
	R.	Ø	TB	ON	IS	DS/NS/ss				
0041	1	4.5	4/3	5/3			mod. • close F • A • FA • D • PR • R	accepted	Tool Presentation	0041
0101	R	2	0.5	1/3	0/3		mod. • open F • A • FA • D • PR • R	all_reports_available	Technical Paper	0101
0061	R	3	0.0	0/3			mod. • open F • A • FA • D • PR • R	all_reports_available	Technical Paper	0061

Below the table is a 'Legends' section with a table of member information:

Initials	First Name	Last Name	Email Address
TB	Tamara	Budnik	pcmember1@metaframe.de
ON	Obalele	Ndango	pcmember2@metaframe.de
IS	Ines	Sanchez Pedroso	pcmember2@metaframe.de
DS	Dragan	Smiljanec	pcmember4@metaframe.de
NS	Natarajan	Srinivasan	pcmember5@metaframe.de
ss	sladjk	sladjka	willebra@metaframe.de

The 'Legends' section also includes explanations for 'Art.ID', 'R.', and 'Ø', and a list of possible actions: 'close F • A • FA • D • PR • R'.

Abbildung 3.17: Screenshot old page "Evaluation Feature"

Showinfo_all.html : Diese Website (siehe Abbildung 3.19) präsentiert alle Informationen über den vom PC Chair gewählten Status. Dort werden die gewählten ArticleIds angezeigt. Mit Hilfe des Ok-Buttons wird der Status gesetzt und durch das Betätigen des Back-Button die vorherige Seite der Evaluation-Matrix aufgerufen.

Evaluation-acknow-cont.html (siehe Abbildung 3.19): Diese Seite wurde neu in den Workflow eingebaut. Erst wenn diese bestätigt wird, wird die zuvor gewählte Aktion auf die selektierten Artikel ausgeführt.

Testphase

hans meier

PC Chair

- Setup
- Bidding
- Articles
- Delegation
- Reports
- Evaluation**
- My Staff
- Conflicts
- Email
- Users
- Roles
- Profile
- Preferences
- Logout
- FAQ

Evaluation Matrix

The following table can be sorted according to the following criteria: [ArticleID](#), [Average](#), [Type](#).
Click on the desired one.

The matrix entries in the columns of the PC Members show the corresponding **score/confidence** value of the report written by the particular member.

Note: Once a discussion forum is opened by the PC Chair, the status in the evaluation matrix is activated as a link to the forum for that specific paper.

Additionally the following two buttons lead you directly to the PC-wide discussion forum and to your individual overview (list) over the discussion forums currently accessible to you.

PC Forum **My Forum Overview**

Refresh (last refresh: *Mon Jan 26 11:44:37 CET 2004*)

S	Art.ID	R.	Score			Status	Type	Art.ID
			Ø	df	KL pm			
■	0021 R	1	1.5	2/1	1/1	accepted	Regular Paper	0021 R
■	0031 R	2	1.0		1/1	accepted	Regular Paper	0031 R
■	0011 R	2	1.0		1/1	accepted	Regular Paper	0011 R

Select Action

- close_forum
- open_forum
- set_a
- set_pa
- set_d
- set_pr
- set_r
- modify

action.

Legends

Abbildung 3.18: Screenshot new page "Evaluation Feature"

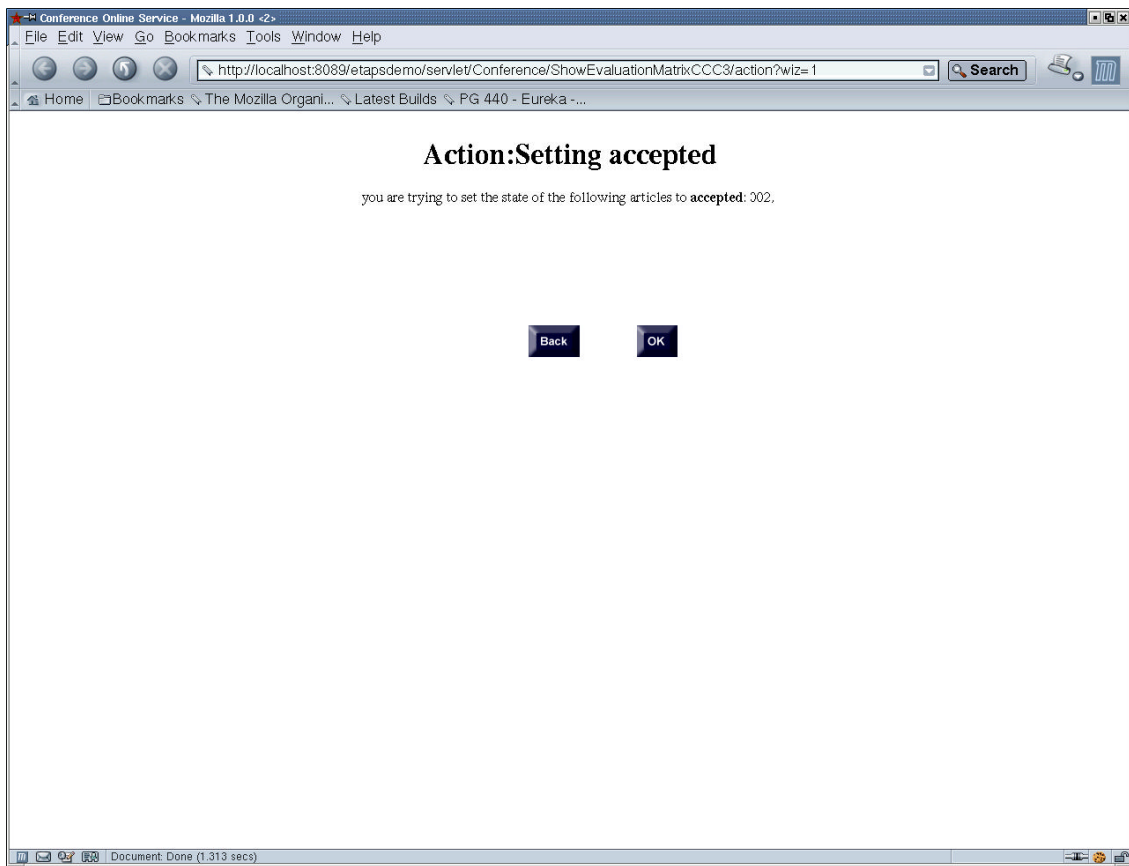


Abbildung 3.19: Screenshot "showinfo-all" beim "Evaluation Feature"

Es wurde die "Select all"-Schaltfläche getestet und festgestellt, dass alle ArticleIds ausgewählt wurden. Beim Setzen eines Status erscheint tatsächlich jeweils eine Bestätigungsseite, wonach die gewählten ArticleIds in der Acknowledgement-Seite angezeigt werden. Nach dem Bestätigen der Seite wird die gewählte Aktion ausgeführt.

3.2.5 News Feature 1

Beschreibung der Aufgabe

Das Lesen der Nachrichten ist zur Zeit zeitaufwendig und umständlich. Das Lesen einer Nachricht führt zu einem Seitenwechsel zwischen der Nachrichtenübersicht und der anzuzeigenden Nachricht. Dies bedeutet ein häufiges Klicken für den Benutzer. Die selektierte Nachricht soll unterhalb der Nachrichtenübersicht angezeigt werden, wie es auch in herkömmlichen Email-Anwendungsprogrammen realisiert ist.

Modellierungsphase

Nach der Auswahl der zu öffnenden Email, soll diese direkt unter der Mailingliste auftauchen, anstatt wie zuvor in einem neuen Fenster. Für den Service Logic Graphen bedeutet das, dass die Email aus dem entsprechenden Array gelesen wird, anschließend überprüft wird, ob ein Anhang in Form einer Datei vorhanden ist und dann in der gleichen Seite (all_folder_messages.html) angezeigt wird, statt wie zuvor in einer Neuen (read_message.html). Eine neue Html-Seite ist beim Anzeigen nicht mehr notwendig, daher entfällt sie. Abbildung 3.20 zeigt das Use Case des Ist-Zustands bei der Auswahl und Lesen einer Email im Online Conference Service. Nach der Auswahl der zu lesenden Email, wird diese in einem neuen Fenster geöffnet, von dem man wieder zurück zur Mailliste oder zur gesamten Forumübersicht navigieren kann.

Im Gegensatz zum Ist-Zustand wird nun (siehe Abbildung 3.21) nach der Auswahl der Email kein neues Fenster geöffnet. Die Email wird nach der Modifizierung des Workflows direkt unterhalb der Mailliste angezeigt. Weiteres Navigieren ("back" , "Forum Overview") ist hier nicht mehr notwendig. Das Activity Chart in Abbildung 3.22 zeigt den detaillierten Workflow beim Selektieren und Lesen einer Email im News Feature des Online Conference Services. Zunächst erscheint eine Übersicht der eingegangenen Emails. Nach dem die entsprechende Mail ausgewählt wurde, wird diese aus dem Newsserver geladen und in einem neuen Fenster angezeigt. Die Mails werden, abhängig vom Verzeichnis in dem sie sich befinden, in Arrays verwaltet.

Analog zur Abbildung 3.23 wird im Soll-Zustand die entsprechende Mail aus dem richtigen Array vom Newsserver geladen, jedoch nicht wie zuvor in "read_messages.html" sondern in "all_folder_messages.html" (Übersicht) angezeigt.

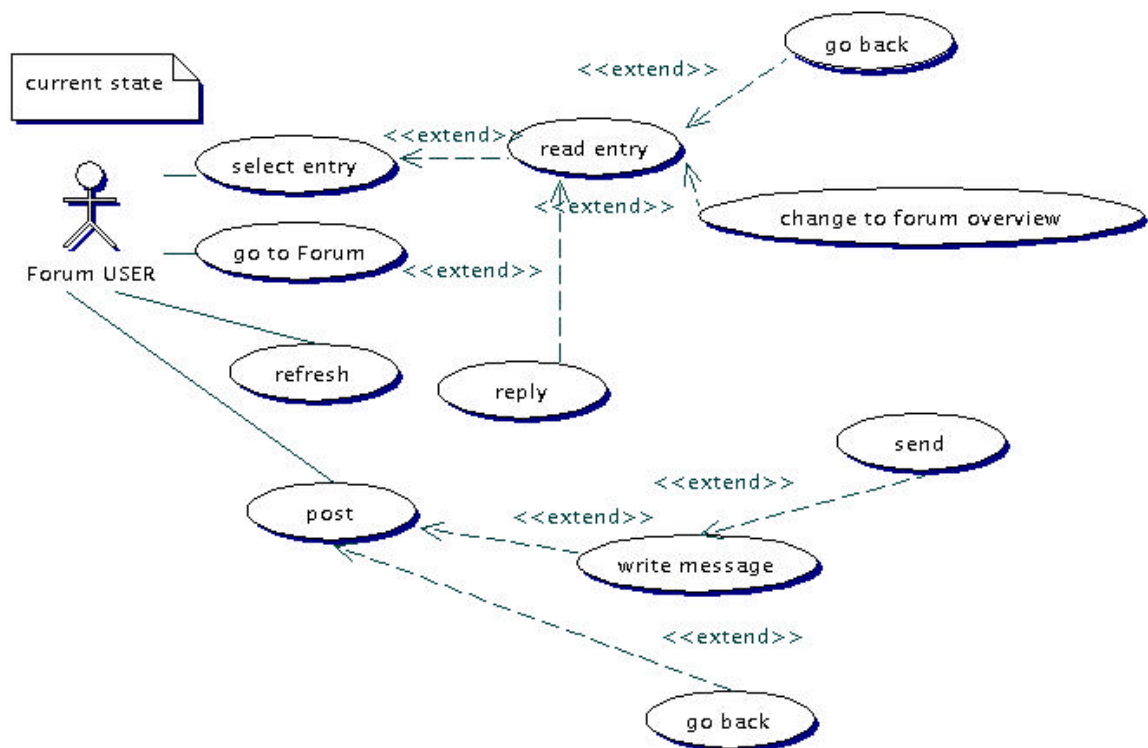


Abbildung 3.20: Use Case des Ist-Zustands "News Feature 1"

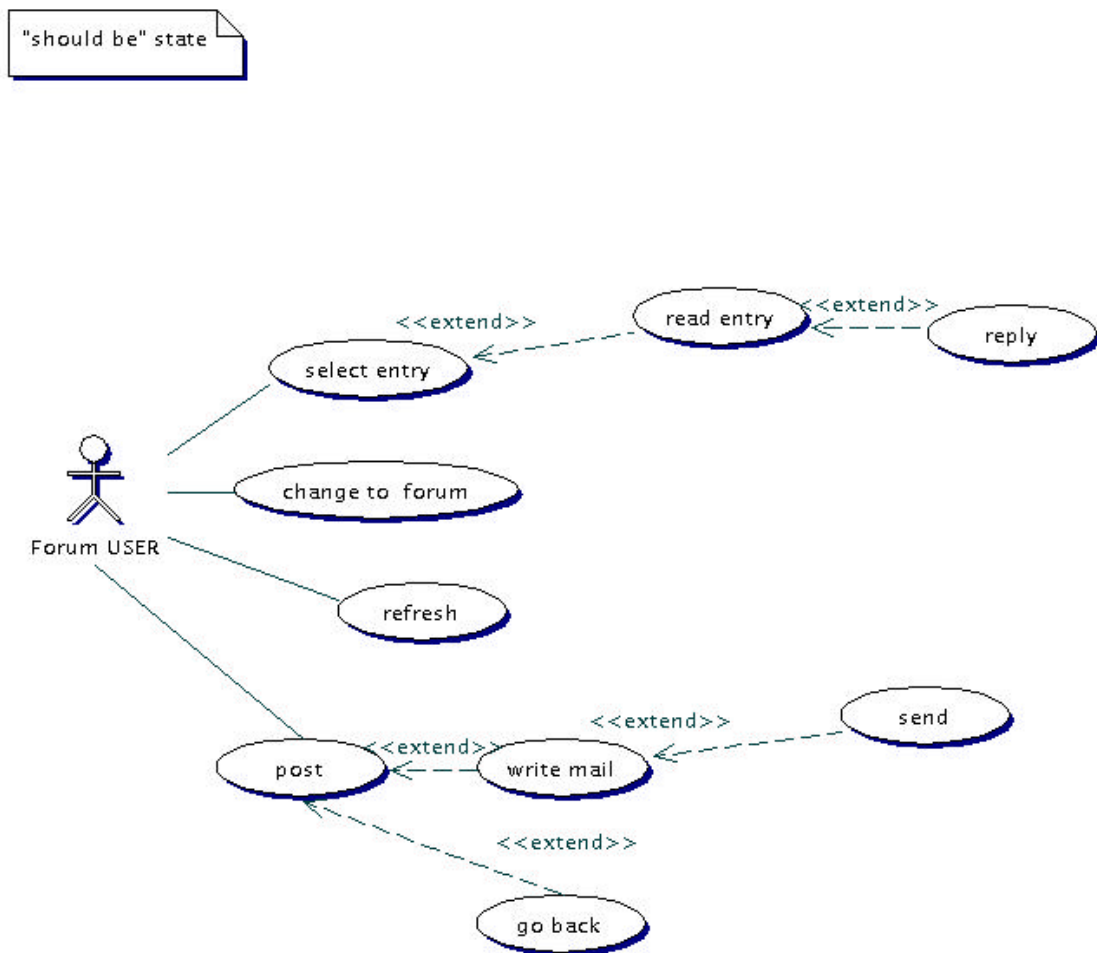


Abbildung 3.21: Use Case des Soll-Zustands "News Feature 1"

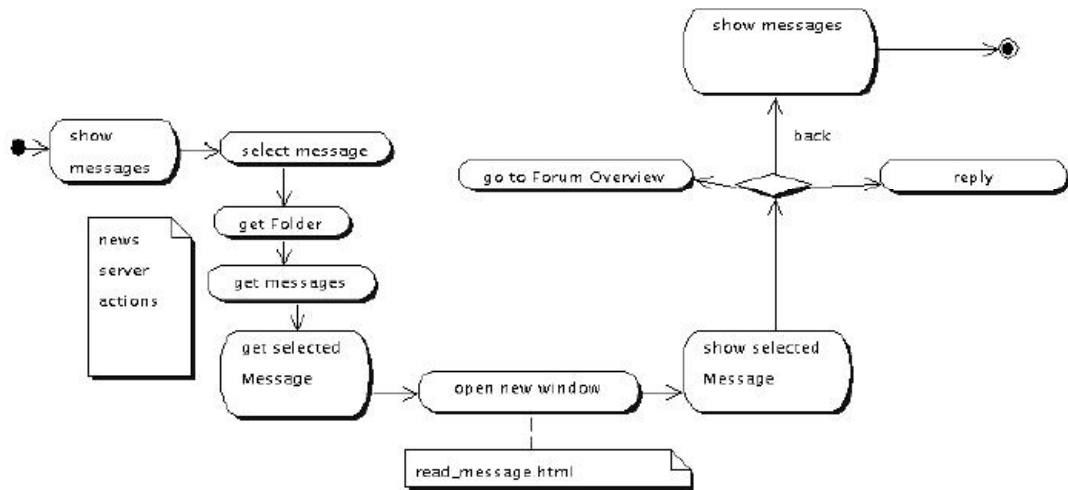


Abbildung 3.22: Activity Chart "Show messages" Ist-Zustand des "News Feature 1"

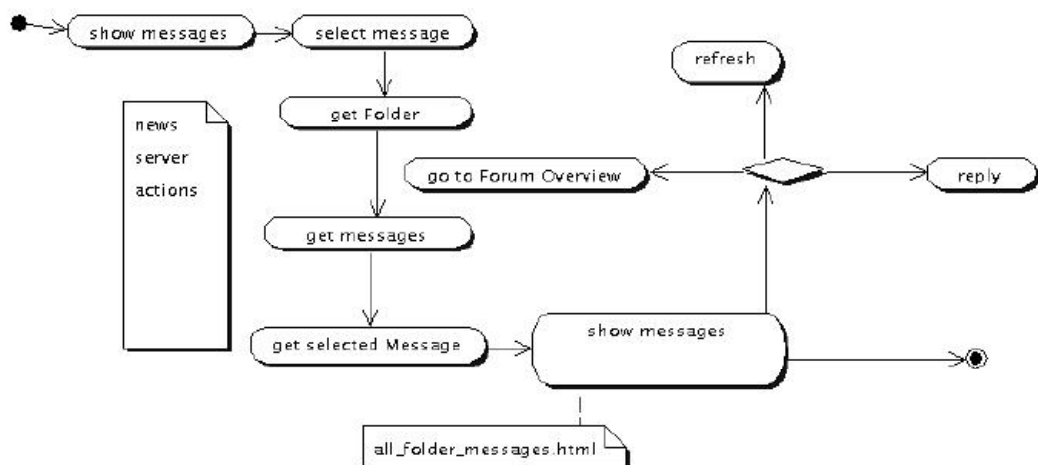


Abbildung 3.23: Activity Chart "Show messages" Soll-Zustand "News Feature 1"

Implementierungsphase

Service Logic Graph

Workflow

Der entsprechende Workflow der SIBs sah ursprünglich wie folgt aus: Forumoverview (Tabelle)...(1),(2),(3) ...(18)...Antwortfenster wird geöffnet ...ShowMessages(1), Service2Callcontext(2), OpenConnection(3), Request2CallContext(4), Session2CallContext(5), CallContext2PreShow(6), GetFolder(7), OpenFolder(8), GetFolderMessages(9), GetMessageFromArray(10), GetMessageContent(11), WriteIdtoCommUserProperties(12), CheckPartorMultipart(13), PreShowRecipientTypes(14), CallContext2PreShow(15), CloseConnection(16), showMessage(17), Request2CallContext(18)...

- (1): Nachrichtenübersicht wird angezeigt (all_folder_messages.html)
- (2)-(6) : Auswahl der Nachricht in der aktuellen Seite wird in den Call Context geladen
- (7)-(11): entsprechende Nachricht wird aus der entsprechenden Liste geladen
- (13)*: Überprüfung, ob die Email einen Anhang enthält
- (15)-(16): Daten werden ins PreShow geladen
- (17): Nachricht wird in einem neuen Fenster angezeigt (read_message.html)
- (18)... : Antwortfenster wird geladen

* Falls die Email einen Anhang enthält, folgt ein analoger Workflow, jedoch mit entsprechend anderer Html-Seite zum Anzeigen der Email (read_multipart_message.html statt read_message.html).

Änderungen

Offenbar muss das ShowMessage SIB entfallen, da hier keine neue Html-Seite geladen werden soll, sondern die Aktuelle erneut (mit ergänzter Email). Die entsprechende Kante muss also wieder nach oben zeigen: aus Kante (16)- (17) wird (16) - (1). Das Antworten muss natürlich der SIB (1) übernehmen: aus der Antwort Kante (17)-(18) wird also (1)-(18). Diese Änderung muss natürlich analog für den Teil erfolgen, der durchlaufen wird, falls die Email einen Anhang enthält. Der zusätzliche SIB, der eine Email mit Anhang anzeigt, muss ebenfalls entfallen. Anstatt soll wieder all_foldermessages.html geladen werden (SIB(1)). Anstelle der bisherigen Reply-Kante verfügt das SIB (1) nun über eine zweite Reply-Kante ("reply_attach"). Zusätzlich muss eine Download-Kante (1) - x existieren, wobei x den bisherigen Download-Workflow darstellt.

Fazit

- 2 Show SIBs wurden gelöscht (...)
- SIB ShowMessages wird um 3 Ausgänge erweitert
 - a) Reply-Kante für anhanglose Emails: gleiches Ziel wie Reply-Kante des 1. gelöschten Show SIBs
 - b) Reply-Kante für Emails mit Anhang: gleiches Ziel wie Reply-Kante des 2. gelöschten Show SIBs
 - c) Download-Kante für Emails mit Anhang: gleiches Ziel wie Download Kante des 2. gelöschtes Show SIBs
- eine Kante jeweils für anhanglose Emails und analog für Emails mit Anhang geändert: statt nach "closeConnection" ein neues Show SIB und eine neue Html-Seite anzuzeigen, wird die aktuelle Html-Seite wiedergeladen. (Workflow zum Anzeigen der aktuellen Seite wird noch einmal durchlaufen bis zum SIB (1))

WML/HTML

1. allfoldermessages.html /.wml
2. readmessage.html /.wml
3. read_multipartmessage.html/ .wml

Konzept

Seite (1) muss um (2) und (3) erweitert werden.

Umsetzung

(2) und (3) müssen vor dem Kopieren in (1) leicht verändert werden. Die beiden Funktionen und somit auch die Buttons "back" und "Forum Overview" entfallen und können gelöscht werden, da sie nicht mehr sinnvoll sind (es wird kein neues Fenster geöffnet). Zusätzlich muss die "action" Bezeichnung in (3) geändert werden (<.....action-"\$thisSIB.reply_attach>) da die Reply-Kante nun "reply_attach" statt "reply" heisst. In (1) muss nun eine Abfrage erfolgen, ob es sich bei der aktuellen Nachricht um eine Nachricht mit oder ohne Anhang handelt. Ist kein Anhang vorhanden, so wird (2) jedoch nicht als eigene Datei, sondern in der gleichen Html-Seite direkt unter der Mailingliste angezeigt. Andernfalls, falls ein Anhang vorhanden ist, wird analog (3) angezeigt.

Testphase

Besonders berücksichtigt werden musste die Tatsache, dass das Newsfeature unabhängig voneinander um zwei Komponenten erweitert wurde (3.2.5 und 3.2.6) und anschließend erst zu Einem makrofiziert wurde. Es mussten diverse Fälle getestet werden: Anhanglose Emails und Emails mit Anhang müssen in jedem Status (gelesen, ungelesen, eigene) und in jeder Tiefe des Baumes (Antwort der Antwort usw.) lesbar und weiter beantwortbar sein. Weiterhin muss im Anhang der Downloadlink bis zum fertigen Download getestet werden.

3.2.6 News Feature 2

Beschreibung der Aufgabe

Bisher waren die gesendeten Nachrichten für den Sender (Autor) in der Messagebox als ungelesen markiert. Dies führte zur Unübersichtlichkeit. Die Aufgabe besteht darin "Post"-Seiten so zu erweitern, dass es dem Sender möglich ist, selbst zu entscheiden, die Nachricht an sich selbst, als gelesen markiert, zu verschicken.

Modellierungsphase

Das Use Case Diagramm (siehe Abbildung 3.24) zeigt den jetzigen Zustand der

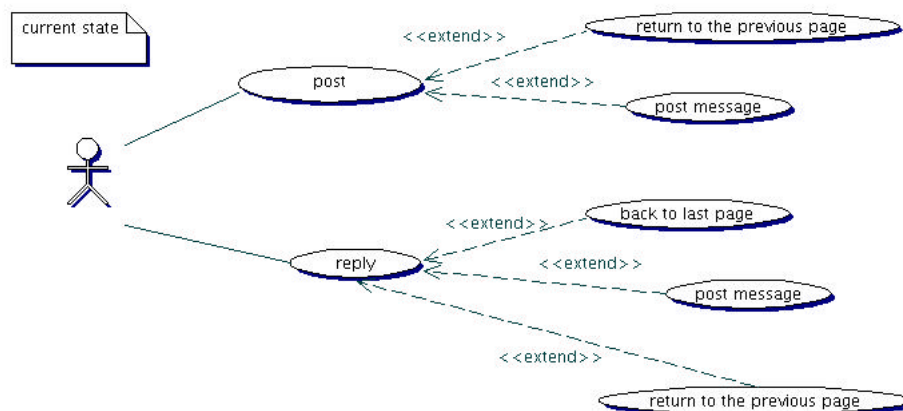


Abbildung 3.24: Use Case "News Feature 2" Ist-Zustand

zwei Geschäftsvorfälle des Online Conference Services.

Der erste Vorfall ist das Verschicken (post) einer neuen Nachricht. Der andere ist das Antworten (reply) auf eine Nachricht.

Das Use Case Diagramm (siehe Abbildung 3.25) zeigt den Soll-Zustand der zwei Geschäftsvorfälle des Online Conference Services, die gleich der oben beschriebenen

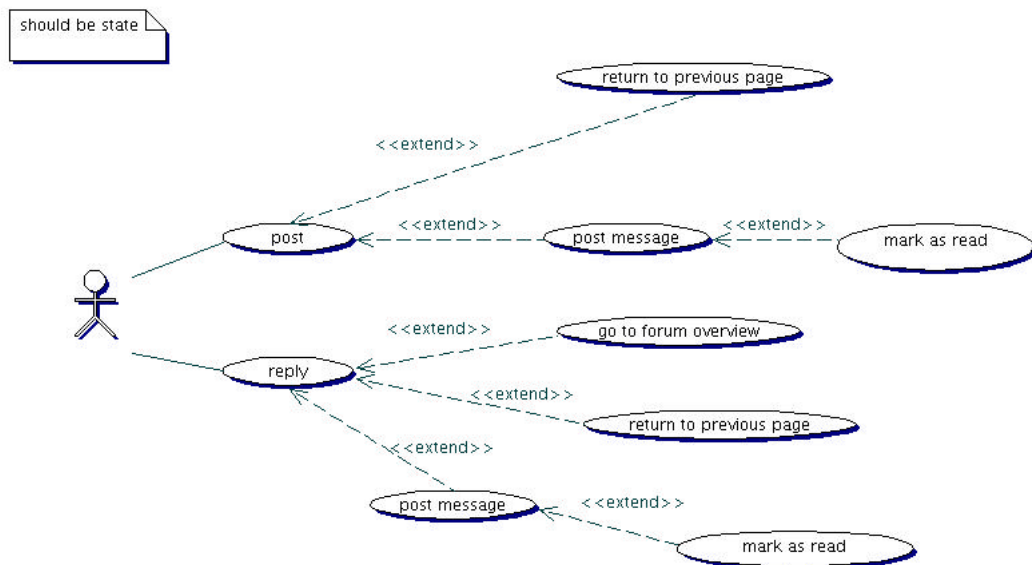


Abbildung 3.25: Use Case "News Feature 2" Soll-Zustand

Vorfälle sind. Mit der Erweiterung, optional zu wählen, ob die gesendete Nachricht für den Sender als gelesen/ungelesen markiert wird.

Die zwei Aktivitätsdiagramme in der Abbildung 3.26 präsentieren den jetzigen Zustand der zwei Workflows, die beim Verschicken einer neuen Nachricht, oder dem Antworten auf eine Nachricht abgearbeitet werden. Der linke Workflow zeigt die Sequenz, die beim Verschicken einer neuen Nachricht abgearbeitet wird und der rechte Workflow die Sequenz, die beim Antworten auf eine existierenden Nachricht abgearbeitet wird.

Die zwei Aktivitätsdiagramme in der Abbildung 3.27 präsentieren den Soll-Zustand der zwei Workflows, die beim Verschicken einer neuen Nachricht, oder dem Antworten auf eine Nachricht abgearbeitet werden mit der Erweiterung diese Nachrichten optional als gelesen/ungelesen zu markieren. Der linke Workflow zeigt die Sequenz, die beim Verschicken einer neuen Nachricht abgearbeitet wird und der rechte Workflow die Sequenz, die beim Antworten auf eine existierende Nachricht abgearbeitet wird.

Implementierungsphase

Veränderungen im OCS

Es wurden zwei unterschiedliche Veränderungen am OCS vorgenommen. Die erste Modifikation wurde an dem Graphen newsclient.msg, die zweite an den wml-Dateien durchgeführt.

Zwei wml-Seiten wurden verändert: write_message.wml und write_reply_message.wml. Diese Seiten wurden mit dem Element Checkbox erweitert. (siehe Abbildung 3.29)

Die Sequenz von SIBs in der Abbildung 3.28 wurde an zwei verschiedenen Stellen des Graphen eingefügt. Das sind die Stellen, wo die Reply Message oder die neu erstellte

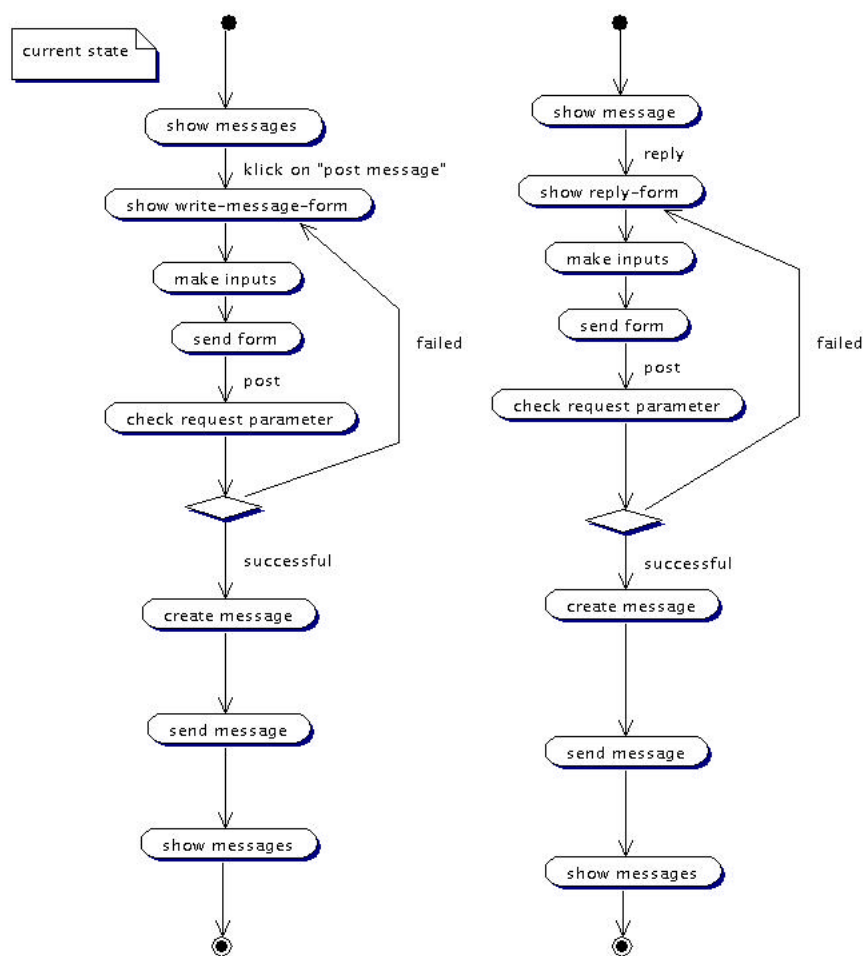


Abbildung 3.26: Activity Chart "News Feature 2"

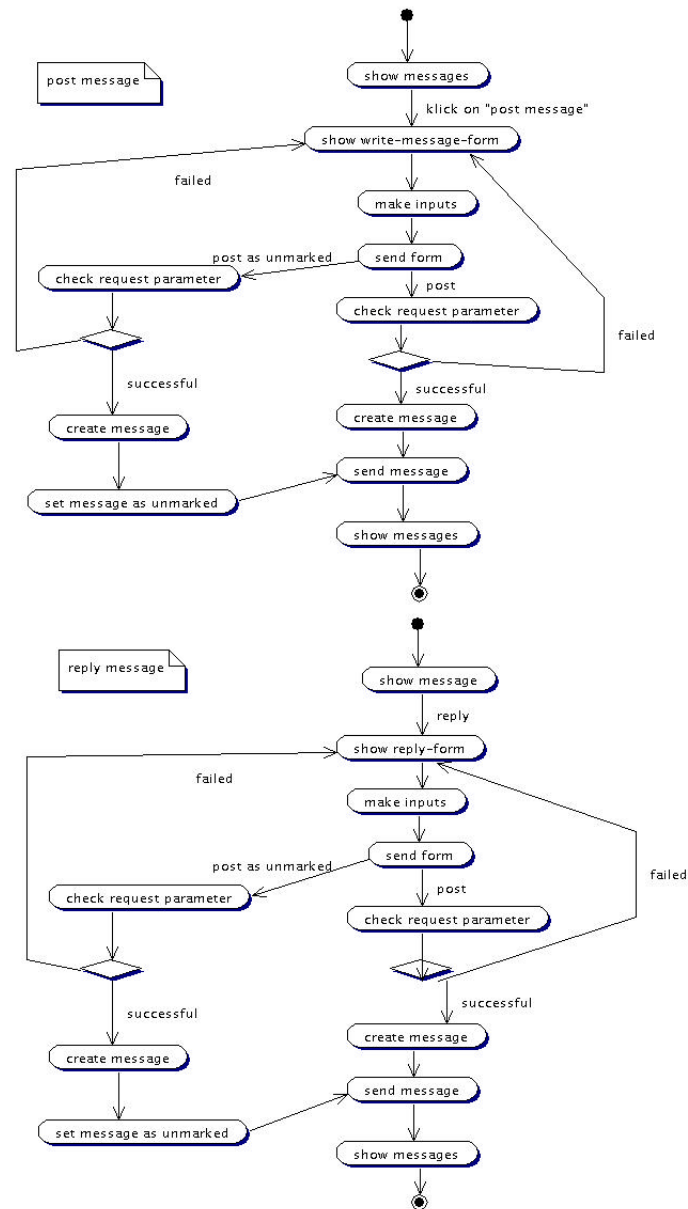


Abbildung 3.27: Activity Chart "reply message" und "post message"

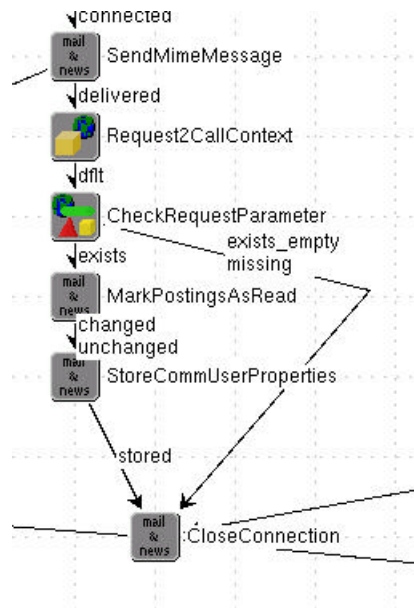


Abbildung 3.28: Abschnitt des Graphen, der im News Feature 2 eingesetzt wurde

Message an den Server verschickt wurden.

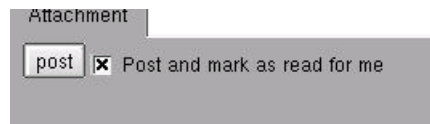


Abbildung 3.29: Checkbox im HTML-Dokument nach Modifizierung des News Feature Workflows

SIB "ReRequest2CallContext" schreibt den Verzeichnisnamen in den Call Context. Dies wird für den *MarkPostingsAsRead*-SIB benötigt.

SIB „CheckRequestParamter“ überprüft den Namen des Parameters der Checkbox. Ist die Checkbox selektiert, wird dem "exists" Branch gefolgt, ansonsten wird der andere Branch benutzt.

SIB „MarkPostingsAsRead“ Dieser SIB schreibt die Message-ID in die *CommUserProperties*, so dass der Service diese Nachricht als gelesen interpretiert.

SIB „StoreCommUserProperties“ Dieser SIB speichert die Änderungen in den CommUserProperties.

Testphase

Das Schreiben einer neuen Nachricht und das Antworten auf eine existierende Nachricht wurden getestet. Der Test berücksichtigte alle möglichen Fälle. Die Mitglieder meiner Gruppe haben die Erweiterungen getestet und als erfolgreich gelöst befunden.

3.2.7 Staff Feature

Beschreibung der Aufgabe

Ziel war es den Workflow zum Hinzufügen von Staff-Membren effizienter zu gestalten. Bisher erlaubte das Formular, in dem man einen neuen Staff-Member aus der User-Liste der Konferenz auswählt, lediglich eine einzige Person auszuwählen. Dies bedeutet, dass man, um mehrere neue Staff-Member hinzuzufügen, gezwungen war immer wieder die gleichen Schritte für jeden neuen Staff-Member durchzugehen. Das Formular und somit der Workflow sollten insofern verändert werden, dass man in der Liste der zur Verfügung stehenden User so viele User anwählen kann, wie man möchte, in dem man statt einer Options-Gruppe von Radio-Buttons Checkboxes verwendet, und für jeden User in der Liste ein Text-Eingabefeld für einen Kommentar zur Verfügung steht. Zuvor war nur ein Textfeld auf dem ganzen Formular vorhanden.

Modellierungsphase

Die Aufgabe ist nicht sehr umfangreich und eine ausgiebige Modellierung in UML nicht als notwendig anzusehen. Da die Einzelkurzthemen jedoch als kleiner Testlauf für die größeren Aufgaben zu betrachten sind, erfüllt die Modellierung durchaus ihren Zweck.

Diagramm "AddStaffMember" (siehe Abbildung 3.30): Das obere Diagramm illustriert, dass es sich hier nicht um eine tiefgreifende Änderung am OCS handelt. Staff-Member hinzuzufügen ist eine Option, die über die Staff-Member-List erreichbar ist. Statt wie bisher nur ein neues Mitglied hinzuzufügen, soll man mehrere Mitglieder auswählen können.

Akteur "PC Chair": Die typische Rolle zum Verwenden des AddStaffMember Features ist der PC Chair. Ist ein PC Chair eingeloggt und befindet sich im Bereich "My Staff", wo die gegenwärtige Liste seiner Staff-Member angezeigt wird, so kann er auf den Link "Add Member" klicken, um zum betroffenen Workflow zu gelangen.

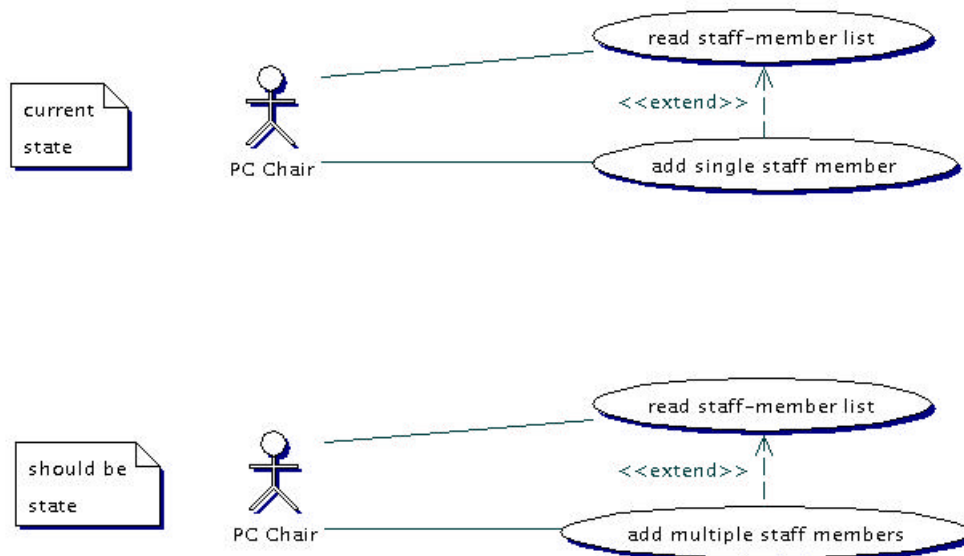


Abbildung 3.30: Use Case "AddStaffMember" des Staff Feature

Use Case "read staff-member list": Ein PC Chair kann sich vom OCS die Liste seiner Staff-Member anzeigen lassen (Hauptmenü "My Staff").

Use Case "add single staff-member": Wenn die Liste der Staff-Member angezeigt wird, steht die Möglichkeit zur Verfügung ein Mitglied hinzuzufügen - daher die "extend"-Beziehung.

Use Case "add multiple staff-members": Wenn die Liste der Staff-Member angezeigt wird, steht die Möglichkeit zur Verfügung, mehrere Mitglieder hinzuzufügen - daher die "extend"-Beziehung.

Activity Chart "AddStaffMember" vorheriger Zustand (siehe Abbildung 3.31): Dieses Diagramm beschreibt den Workflow zum Hinzufügen eines Staff-Members, wie er ursprünglich war - man kann jeweils nur ein neues Mitglied hinzufügen.

Activity Chart "AddStaffMember" soll Zustand (siehe Abbildung 3.32): Dieses Diagramm beschreibt den Workflow zum Hinzufügen mehrerer Staff-Member, wie er sein sollte - man kann jeweils eine beliebige Anzahl neuer Mitglieder hinzufügen.

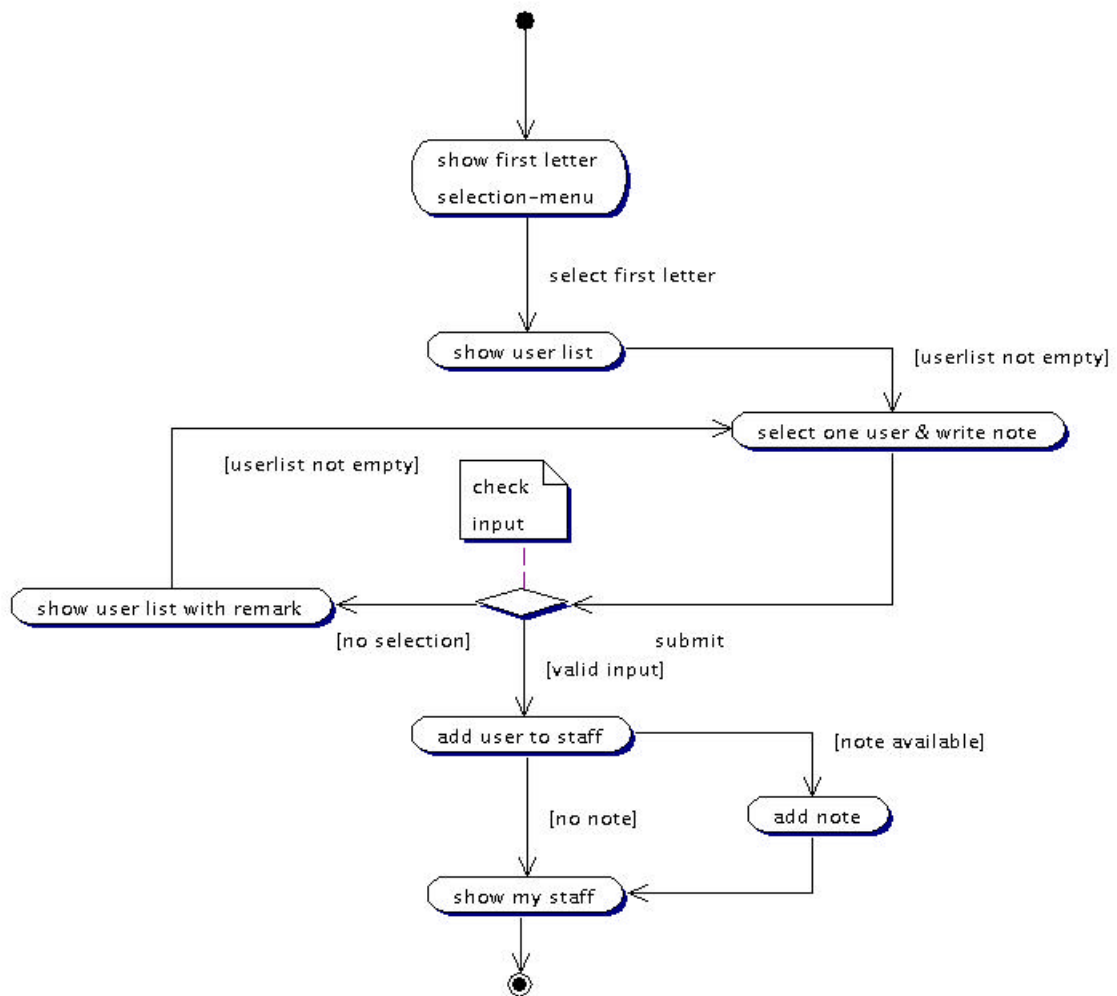


Abbildung 3.31: Activity Chart "AddStaffMember" vorheriger Zustand des Staff Feature

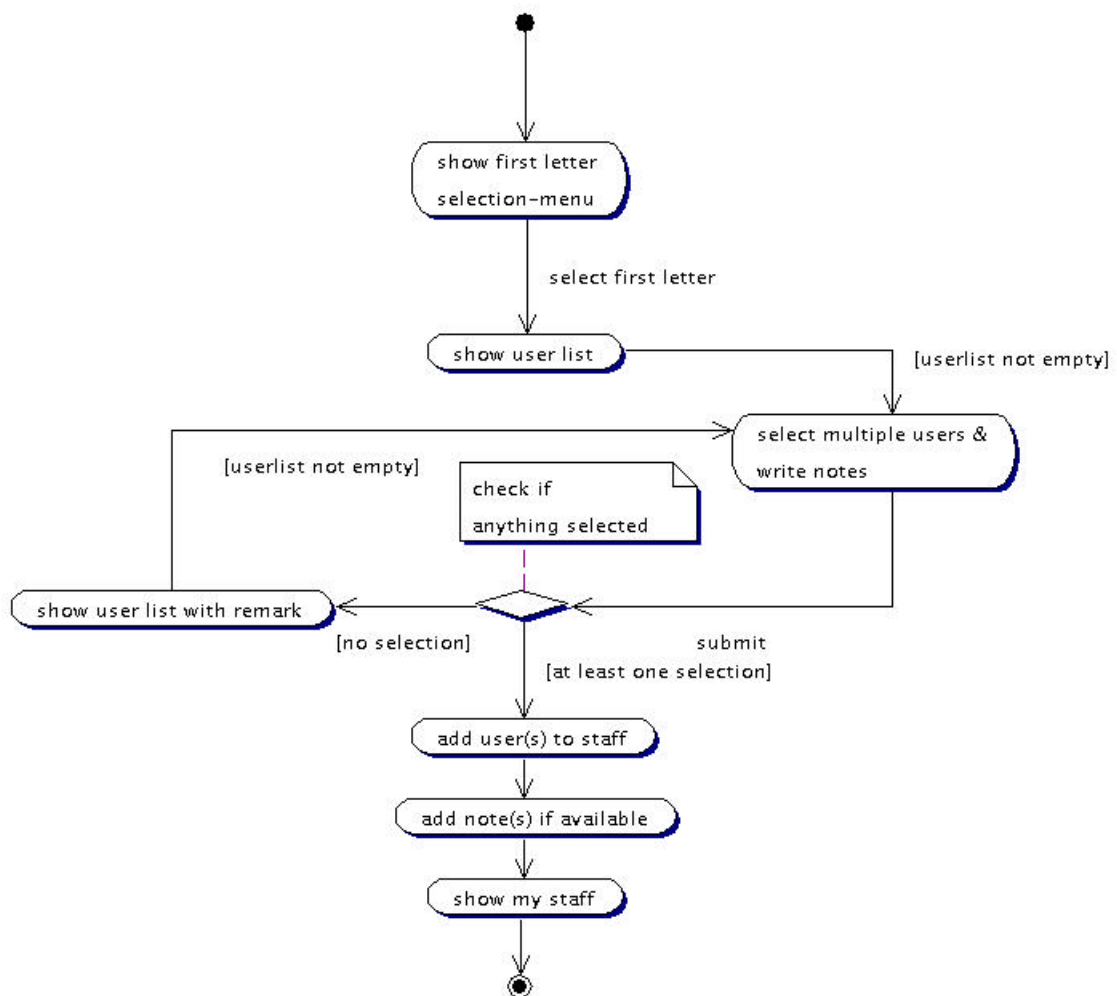


Abbildung 3.32: Activity Chart "AddStaffMember" Soll Zustand des Staff Feature

Implementierungsphase

Im Wesentlichen waren drei Änderungen vorzunehmen, um das gewünschte Resultat zu erzielen. Zunächst sollte aus dem Staff-Teilgraphen des OCS-Hauptgraphen ein Makro erstellt werden, was der Übersichtlichkeit und einer effizienteren Arbeit an dem System dienen sollte. Dann musste das alte SIB "AddStaffMember" mit einem neuen SIB "AddMultipleStaffMembers" ersetzt werden, das fähig war mehrere Staff-Member zu verarbeiten im Gegensatz zu einem Einzelnen. Nachdem die Änderung im Backend der Applikation gemacht worden war, musste noch das Frontend, die HTML-Templates angepasst werden.

Neue Dateien

Makro "staff"

staff.smg

staff.sib

SIB "AddMultipleStaffMembers"

AddMultipleStaffMembers.java

AddMultipleStaffMembers.sib

Veränderte Dateien

Graph "Conference"

Conference.slg

HTML-Template "AddStaffMember"

addstaffmember-cont.wml

(siehe Abbildung 3.33)

Testphase

Das Formular zum Hinzufügen von Staff-Mitgliedern erlaubt im Wesentlichen nur drei mögliche Formen von Eingaben.

Fall 1: Der User erstellt keinen Input.

Fall 2a: Der User wählt einen einzelnen User aus (ohne einen Kommentar daneben zu schreiben).

Fall 2b: Der User wählt einen einzelnen User aus (und schreibt einen Kommentar daneben).

Fall 3a: Der User wählt einige User aus (ohne einen Kommentar daneben zu schreiben).

Fall 3b: Der User wählt einige User aus (und schreibt bei manchen davon einen Kommentar daneben).

Fall 3c: Der User wählt einige User aus (und schreibt bei allen ein Kommentar daneben).

Ergebnisse:

Fall 1: Es wird eine Fehlermeldung angezeigt, die darauf aufmerksam macht, dass keine Angaben gemacht wurden.

Fall 2: Der ausgewählte User wird in der Member-Liste angezeigt (mit Kommentar, falls

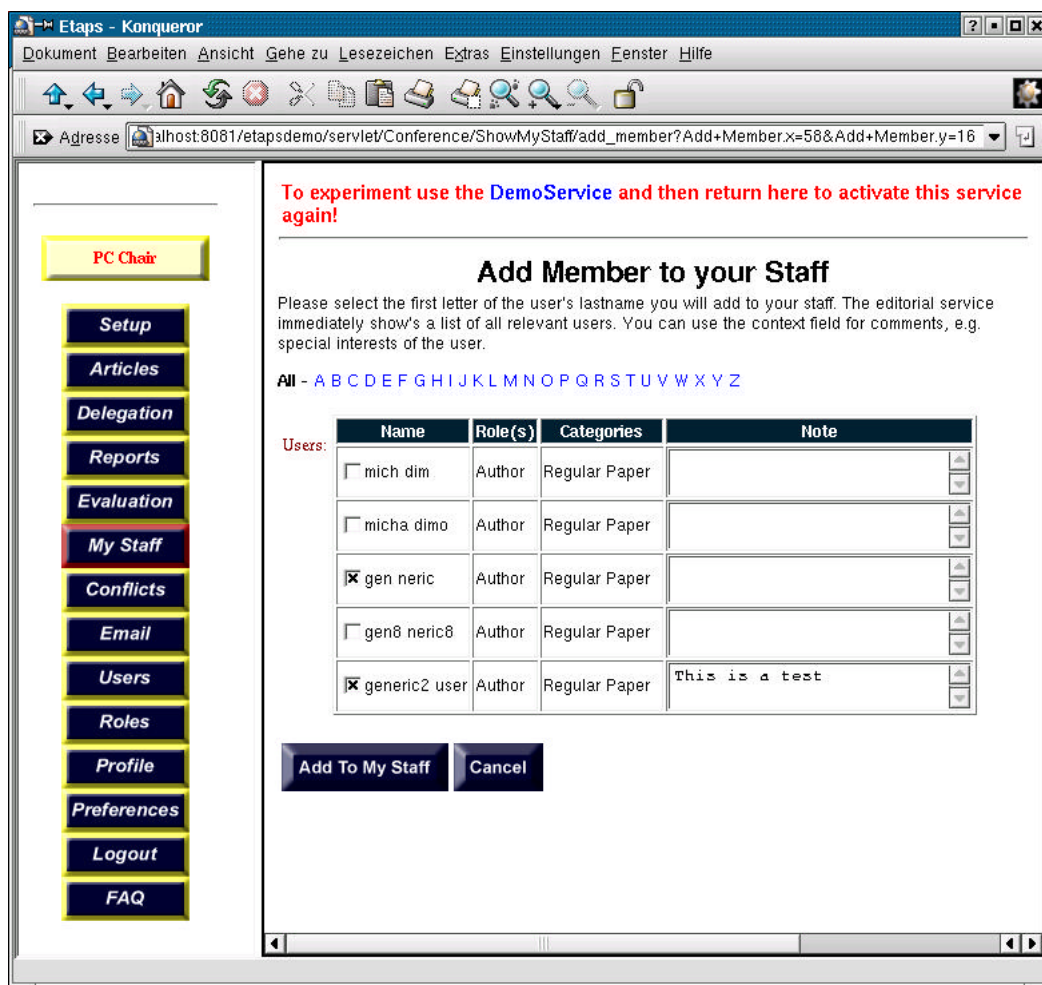


Abbildung 3.33: Ein Screenshot des modifizierten "AddStaffMember"-Formulars

einer gemacht worden ist)

Fall 3: Alle ausgewählten User werden in der Member-Liste angezeigt (mit jeweiligem Kommentar, falls einer gemacht worden ist)

Dieser Test wurde von der Gruppe verifiziert.

3.2.8 Setup Feature 1

Beschreibung der Aufgabe

Dieses Einzelkurzthema befasste sich mit dem Setup des Komitees. Der PC Chair hat dort die Möglichkeit, neue Komiteemitglieder über eine Eingabemaske hinzuzufügen.

Modellierungsphase

Das Problem, das bis zur Umsetzung der Einzelkurzthemen bestand, war dass neu eingegebene Mitglieder nicht übernommen wurden, wenn die Seite, ohne die Eingaben zu speichern, mit mehr Eingabefeldern neu geladen wurde. Des weiteren bestand keine Möglichkeit, die neuen Mitglieder automatisch zum Stab des PC Chairs oder zu den Stäben aller PC Chairs hinzuzufügen. Eingabefehler, etwa ein vergessener Vor- oder Nachname, eine ungültige Emailadresse oder das Anlegen eines Benutzers ohne Rollenzuweisung wurden durch eine unzureichende Fehlermeldung angezeigt (siehe Abbildung 3.34 und 3.35).

Nach der Bearbeitung sollten eingegebene Mitgliederdaten zwischengespeichert werden, wenn die Seite mit einer veränderten Anzahl Eingabefelder neu aufgebaut wurde. Eingabefehler sollen nachvollziehbar angezeigt werden, d.h. Grund und Ort des Fehlers sollen erkennbar sein. Darüberhinaus soll die Möglichkeit bestehen, die neuen Mitglieder direkt dem Stab des PC Chairs oder den Stäben aller PC Chairs zuzuordnen (siehe Abbildung 3.36 und 3.37).

Implementierungsphase

Das erste Ziel war, die ungespeicherten Daten bei einem Wiederaufbau der Seite mit einer veränderten Anzahl Eingabefelder nicht zu verlieren. Da es nicht einfach möglich war, die Daten im Call Context zwischenspeichern, was mit dem bestehenden Design des Komitee Setups zusammenhing, musste ein extra SIB geschrieben werden, der die Daten zwischenspeichert. Diese Aufgabe übernimmt der SIB CacheCommittee-Configuration, der auf CheckCommitteeConfiguration basiert.

Um mit den gespeicherten Daten umgehen zu können, musste auch ShowCommitteeConfiguration angepasst werden. Da anfänglich bereits gespeicherte und neu eingegebene Daten vermischt wurden, wurde ShowCommitteeConfiguration in zwei Teile geteilt. Ein Teil behandelt die neu eingegebenen Daten und der Andere wird ausgeführt, wenn keine neuen Daten eingegeben wurden, was dem alten Verhalten entspricht. Zusätzlich musste

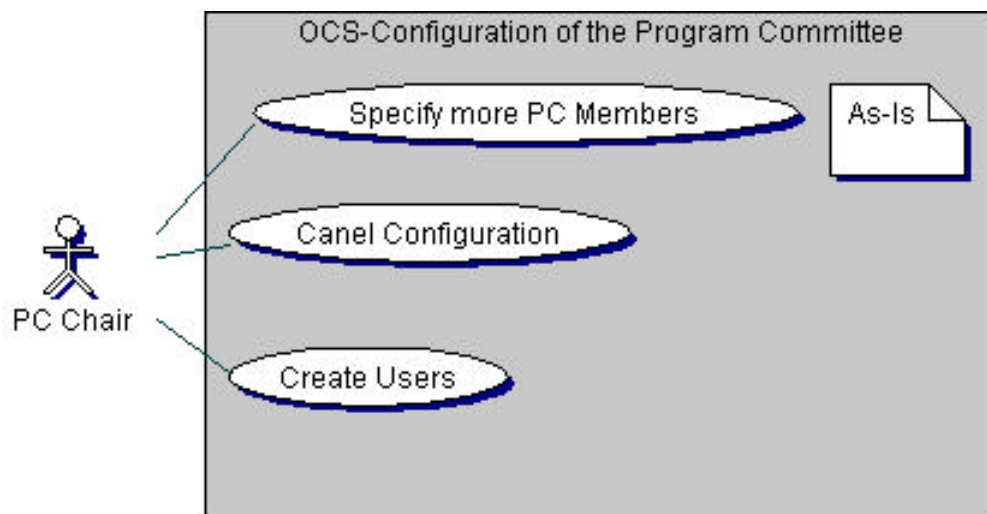


Abbildung 3.34: Zustand vor der Bearbeitung des Kurzthemas "Setup Feature 1"

die Datei `committee-setup-cont.wml` angepasst werden, um die Rollenassoziation, die von `CacheCommitteeConfiguration` angelegt wird, anzuzeigen.

Da eine Verschachtelung nötig war, um eine Schleife über alle neuen User sowie alle PC Chairs zu erreichen, mussten auf Basis von bestehenden SIBs neue SIBs geschrieben werden, die flexibler im Umgang mit den Usern sind, über die geloopt wird.

`LoopOverUserlist` las die Userliste direkt aus der Session. Es war zwar in der Lage Benutzer anhand eines zugewiesenen Features zu filtern, schrieb aber die Benutzerliste unter einem festen Namen in die Session. Dadurch war es unmöglich es zur Verschachtelung zweier Schleifen über User, hier PC Chairs und neuer Benutzer, zu benutzen. Um dieses Problem auch mit Blick auf die Zukunft zu lösen, wurden folgende neue SIBs geschrieben:

`GetUsers` besitzt zwei Parameter: Das (optionale) Feature, das die Benutzer besitzen müssen und der Call Context Key unter dem die Liste abgelegt werden soll.

`LoopOverUsers` besitzt drei Parameter: Den Schlüssel unter dem die Userliste im Call Context abgelegt wurde, einen Schlüssel unter dem der Iterationszähler gespeichert wird und der Schlüssel unter dem der aktuell zu bearbeitende Benutzer abgelegt wird.

Mit diesen beiden SIBs, war es schliesslich möglich, beliebig verschachtelte Schleifen über Benutzerlisten auszuführen. Um nun die neuen Benutzer zum Stab des PC Chairs hinzuzufügen, konnte das vorhandene SIB `AddStaffMember` benutzt werden. Allerdings war es nur in der Lage, einen Benutzer zum Stab des aktuell eingeloggten Benutzers hinzuzufügen. Es sollten aber auch Benutzer zu Stäben nicht eingeloggter PC Chairs hinzugefügt werden. Damit diese beiden Aufgaben von einem SIB erledigt werden konnten, wurde noch der SIB `AddMemberToStaffOfMember` geschrieben. Er liest den

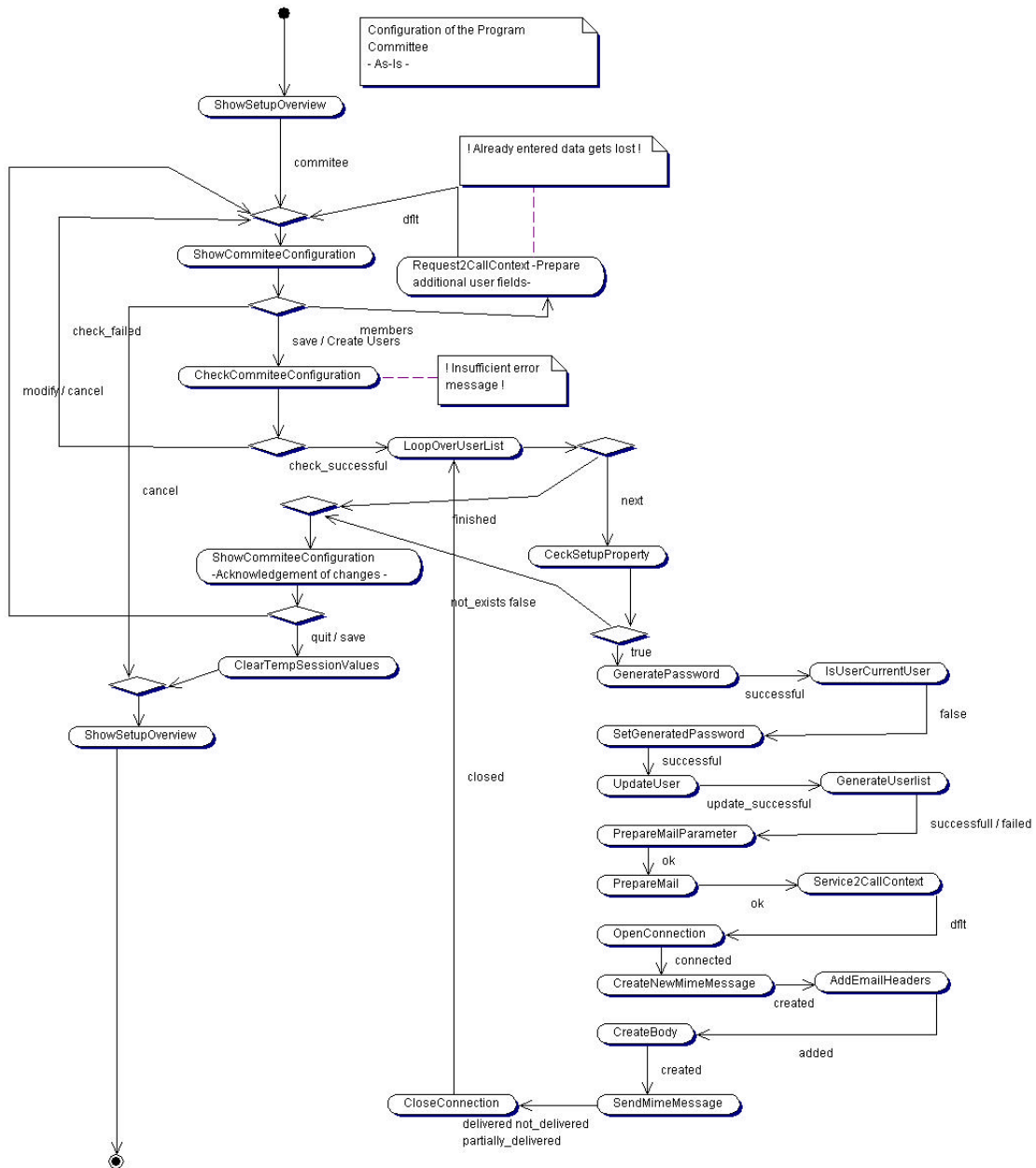


Abbildung 3.35: Zustand vor der Bearbeitung des Kurzthemas "Setup Feature 1"

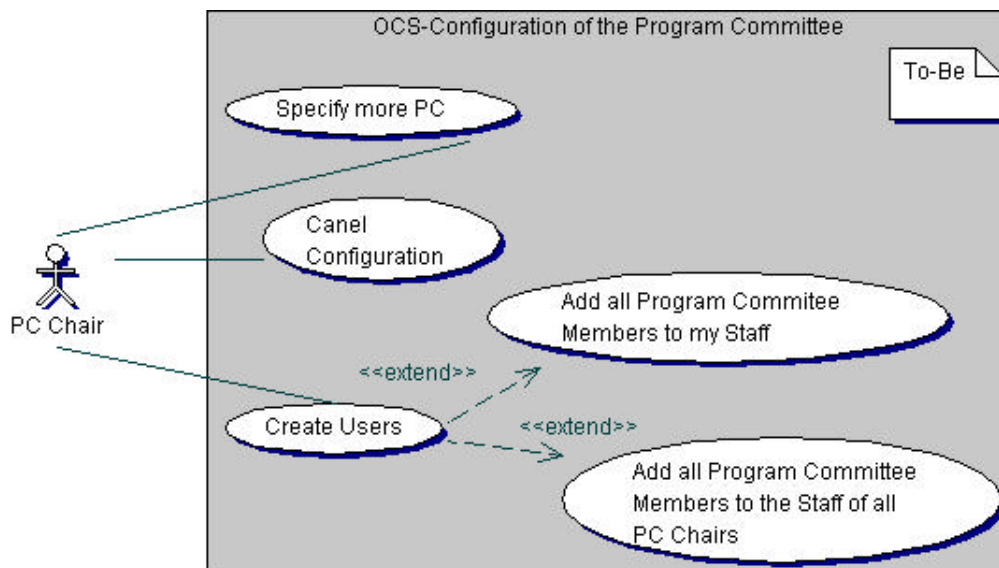


Abbildung 3.36: Zustand nach der Bearbeitung des Kurzthemas "Setup Feature 1"

Besitzer des Stabs und den hinzuzufügenden Benutzer aus dem Call Context.

Testphase

Zuerst wurde getestet, ob der veränderte Workflow ungespeicherte Benutzer nicht verlor, wenn die Seite mit mehr Eingabefeldern als zuvor vorhanden, neu geladen wurde. Nach anfänglichen Vermischungen von alten und neuen Benutzern, die schon in der Beschreibung der Implementierungsphase erwähnt wurden, arbeitete CacheCommittee-Configuration zuverlässig und behielt die eingegebenen Benutzer bei. Wurde die Anzahl der Eingabefelder kleiner als die Anzahl der bereits gespeicherten Benutzer gewählt, wurde immer noch die Mindestanzahl an Eingabefeldern angezeigt.

Der nächste Schritt war das Testen des Hinzufügens der neuen Mitglieder zu dem Stab des PC Chairs bzw. zu den Stäben aller PC Chairs. Das Hinzufügen zum Stab des aktuellen Users (eines PC Chairs) verlief ohne Probleme, ebenso wie das Hinzufügen zu einem beliebigen Stab, da das neue SIB AddMemberToStaffOfMember flexibel genug war, einen beliebigen User und einen PC Chair aus dem Call Context zu lesen.

3.2.9 Setup Feature 2

Beschreibung der Aufgabe

Die Aufgabe bei diesem Kurzthema bestand darin, den Unterpunkt 4 (Configuration of Roles) des Setup Feature zum einen zu erweitern und zum anderen bereits angefangene Implementierungen zu vervollständigen.

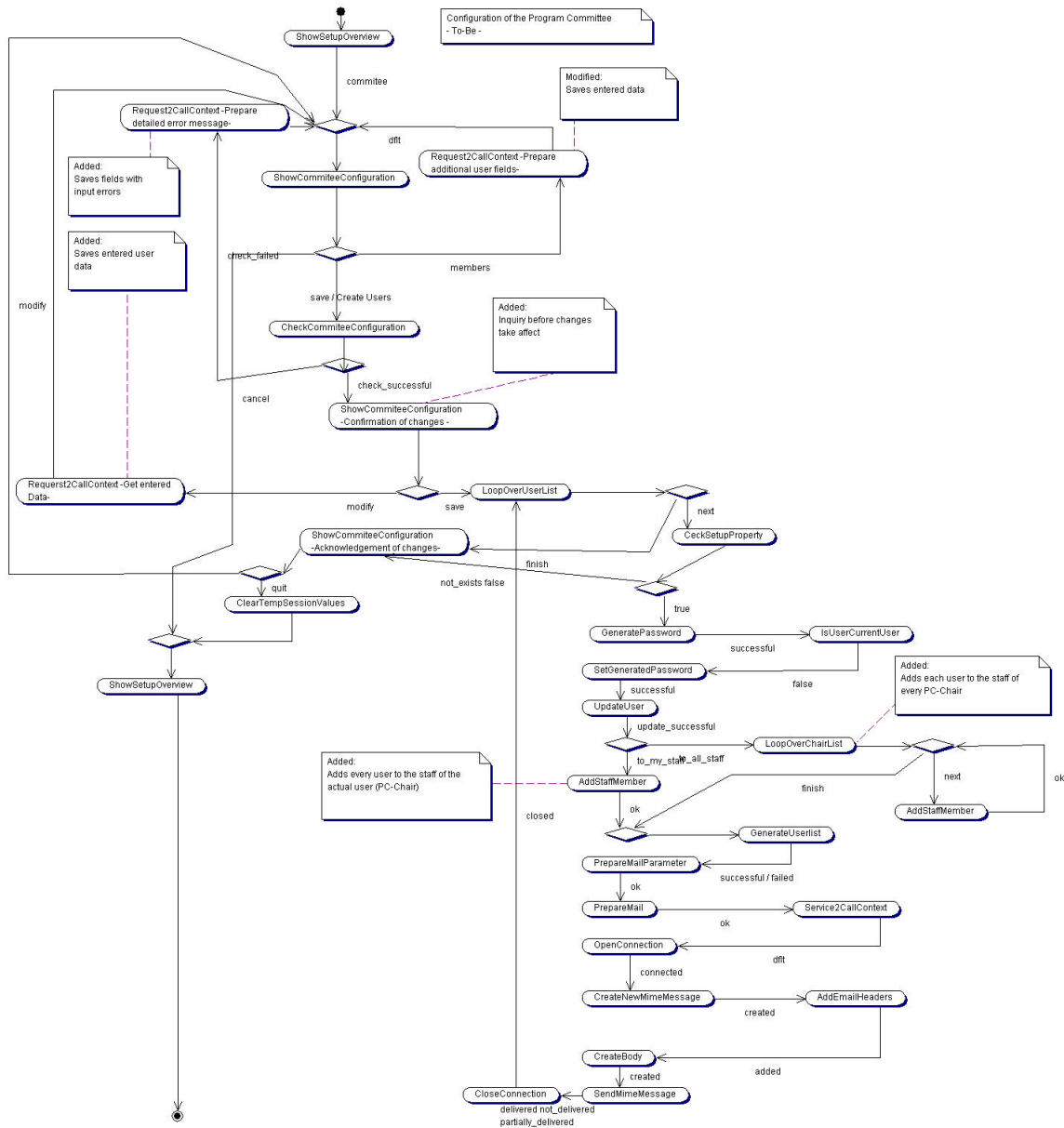


Abbildung 3.37: Zustand nach der Bearbeitung des Kurzthemas "Setup Feature 1"

Author Configuration: Bei diesem neuen Unterpunkt der Rollenkonfiguration soll es dem PC Chair ermöglicht werden zwei wesentliche Permissions von Autoren zu modifizieren. Es soll die Möglichkeit bestehen die Registrierung von neuen Autoren zu unterbinden und außerdem das Einreichen neuer Artikel zu sperren, ohne die Rolle mit Hilfe der Rollenkonfigurationsmatrix zu modifizieren.

PC Chair Configuration: Diese Option bestand zwar schon als Unterpunkt, war aber noch nicht vollständig implementiert. Auf dieser Seite soll es dem PC Chair ermöglicht werden, allen bereits registrierten PC Chairs auch die PC Member Rolle zuzuweisen, damit diese auch selbst eigene Artikel einreichen und fremde Artikel rezensieren können.

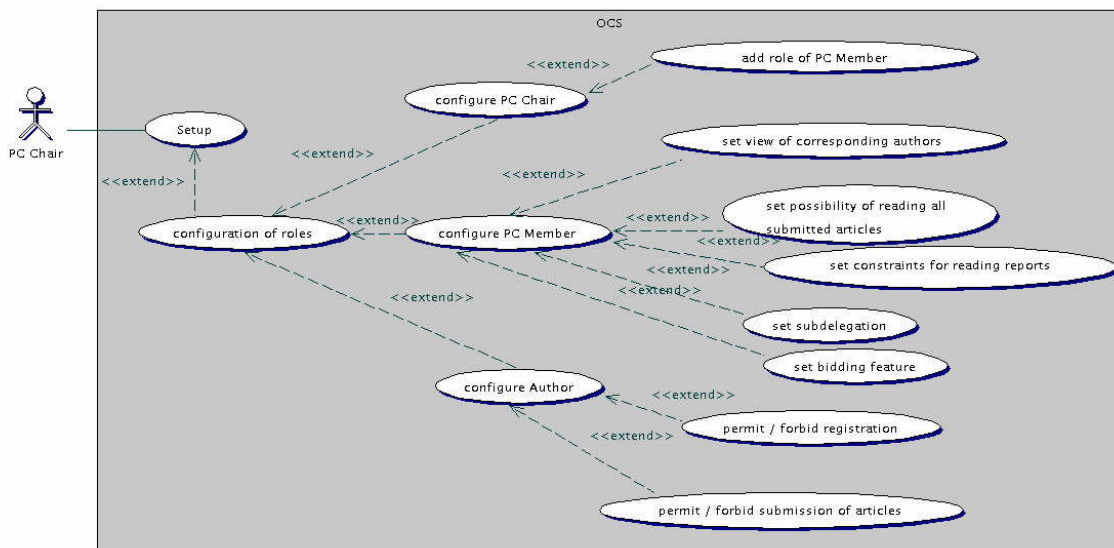


Abbildung 3.38: Use Case Diagramm des Zielzustands "Setup Feature 2"

Modellierungsphase

In der Modellierungsphase ging es bei diesem Kurzthema darum, die bereits vorhandenen Arbeitsabläufe des Dienstes zu analysieren und eine Struktur für die Erweiterung des Graphen zu entwerfen.

Die Benutzerführung für den Konfigurationspunkt "PC Chair Setup" ist schon vollständig vorhanden, hat aber keine Auswirkung auf die eigentlichen Einstellungen. Zunächst gelangt der Benutzer zu einer Seite, auf der er die Einstellungen vornehmen kann, danach auf eine Bestätigungsseite um die Angaben abschließend zu überprüfen. Die neu zu erstellende Benutzerführung für das "Author Setup" (siehe Abbildung 3.38) soll sich an diesem Ablauf orientieren und wird über einen zusätzlichen Link unter "Configuration of

Roles" zu erreichen sein.

Für den Unterpunkt "PC Chair Configuration" ist im Graphen nur die ausgehende Kante des SIBs "ShowPcChairRoleConfiguration" zum nachfolgenden SIB "ClearTempSessionValues" zu entfernen und durch zwei ausgehende Kanten "save" und "modify" zu ersetzen. Die Kante "save" wird gegangen, nachdem der Benutzer auf der Bestätigungsseite die Einstellungen speichert, die Kante "modify" wird gegangen, wenn der Benutzer die Einstellungen noch einmal modifizieren möchte. In der Implementationsphase ist nur die Java Klasse zum SIB "CheckPcChairRoleConfiguration" fertigzustellen, durch das User-Rollen-Rechte Management des Dienstes werden alle Seiten des Webdienstes entsprechend der neuen Rechtevergabe automatisch angepasst.

Der Graph für das Author Setup ist vollständig neu zu entwerfen und orientiert sich am neuen Verfahren für die Verwendung von Show-SIBs. Für die Aufbereitung der Daten für die Template zum Editieren der Einstellungen wird ein Pre-Show SIB eingesetzt, gefolgt von einem SIB der Klasse "ShowFramesetFiles". Dieses allgemeine SIB sorgt im Dienst für die Darstellung der Webseiten. Nach der Übernahme der Werte aus dem Request in den Call Context werden die Angaben durch den ebenfalls neuen SIB "CheckAuthorRoleConfiguration" ausgewertet und die entsprechenden Werte gespeichert. Anschließend wird die nun gültige Konfiguration auf einer Bestätigungsseite angezeigt, hierfür wird wiederum ein SIB des Typs "ShowFramesetFiles" verwendet. Auf der Bestätigungsseite kann der Benutzer wie beim PC Chair Setup die Änderungen übernehmen oder zur vorherigen Seite zurückkehren, um die Einstellungen zu modifizieren. (siehe Abbildung 3.39). Da die Möglichkeit, sich als neuer Benutzer beim Dienst zu registrieren, entzogen werden soll, ist eine Modifikation außerhalb des Setup Makro im Graphen notwendig. Um die Schließung der Registration im System zu berücksichtigen wird in einer Konfigurationsdatei ein neuer Wert angelegt und dieser auf "true" gesetzt. Dieser Wert muss nun im Teilgraphen des Registration Feature ausgewertet und bei geschlossener Registration eine entsprechende Seite angezeigt werden (siehe Abbildung 3.40).

Implementierungsphase

Um die "PC Chair Configuration" zu vervollständigen musste die Implementation des SIBs "CheckPcChairRoleConfiguration" vervollständigt werden. In der dazugehörigen Java Klasse werden alle zur Zeit im Dienst registrierten Benutzer mit PC Chair Rolle ausgelesen und die PC Member Rolle entzogen, beziehungsweise zugeordnet.

Der komplett neu entwickelte Teilgraph für das Author Setup benutzt bereits existierende SIBs zur Anzeige der WML Templates und zwei neue SIBs: "PreShowAuthorRoleConfiguration" und "CheckAuthorRoleConfiguration". Die Java Klasse zum SIB "PreShowAuthorRoleConfiguration" liest die derzeit geltenden Einstellungen aus der Datenbank und der Konfigurationsdatei um diese dann entsprechend in den Call Context zu kopieren. Die Klasse zum SIB "CheckAuthorRoleConfiguration" wertet die Benutzereingaben aus und führt die daraufhin notwendigen Konfigurationen durch. Der Rolle "Author" wird das Recht neue Artikel einzureichen (F-ART-03) mit Hilfe des Rollenadministrators zugeordnet oder entzogen. Der Status der Registration wird im

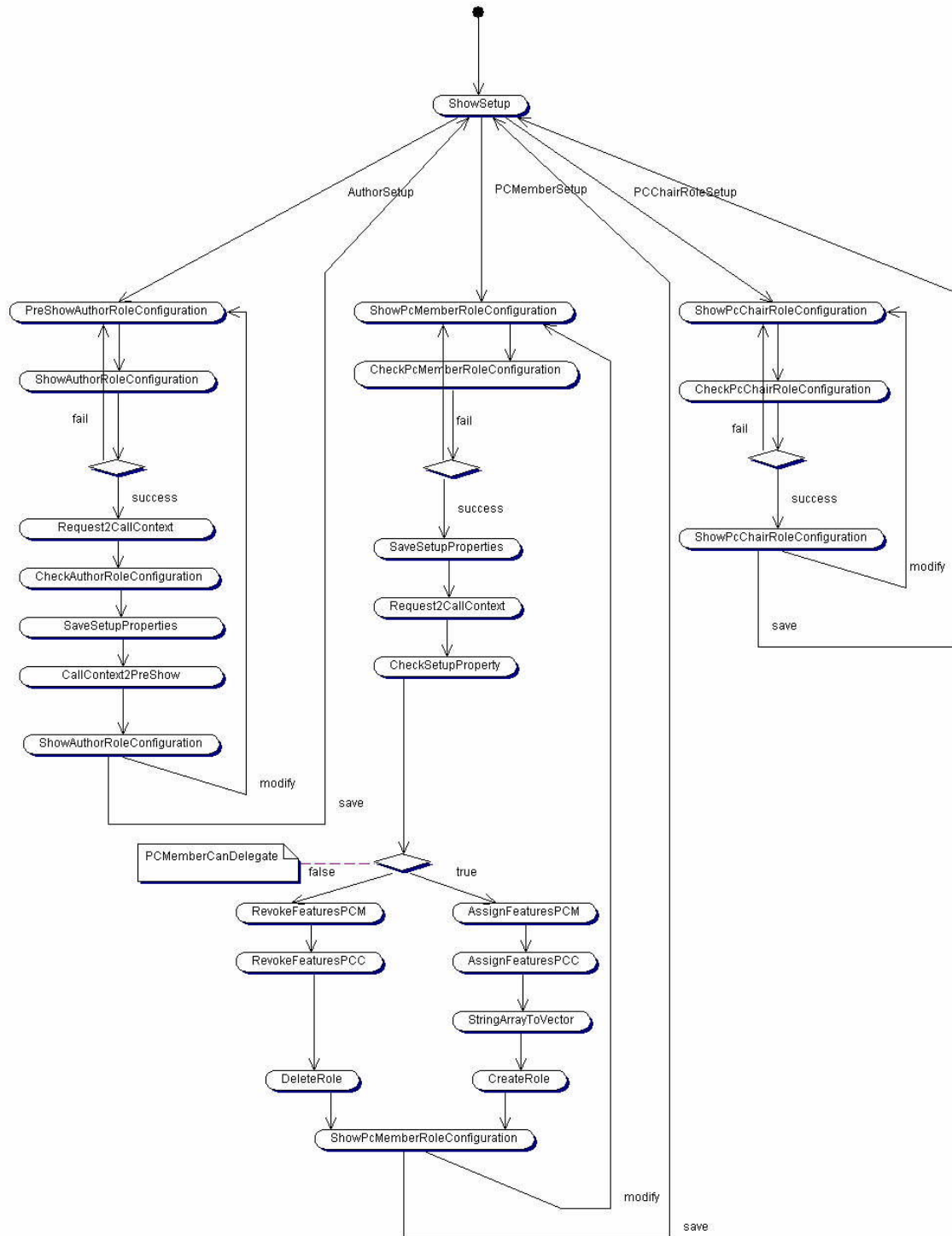


Abbildung 3.39: Ablauf des Unterpunktes "Role Configuration" im Setup, Zielzustand

Wert "RegistrationClosed" in der Datei "Setup.properties" des Dienstes abgebildet. Diese Datei stellt eine zentrale Konfigurationsdatei für den Dienst dar und ihr Inhalt wird immer in der Session gehalten. Daher muss zum einen die Kopie in der Session modifiziert werden und zusätzlich die Datei auf der Festplatte durch das nachfolgende SIB "SaveSetupProperties" geändert werden. Um den Status der Registration zu berücksichtigen ist

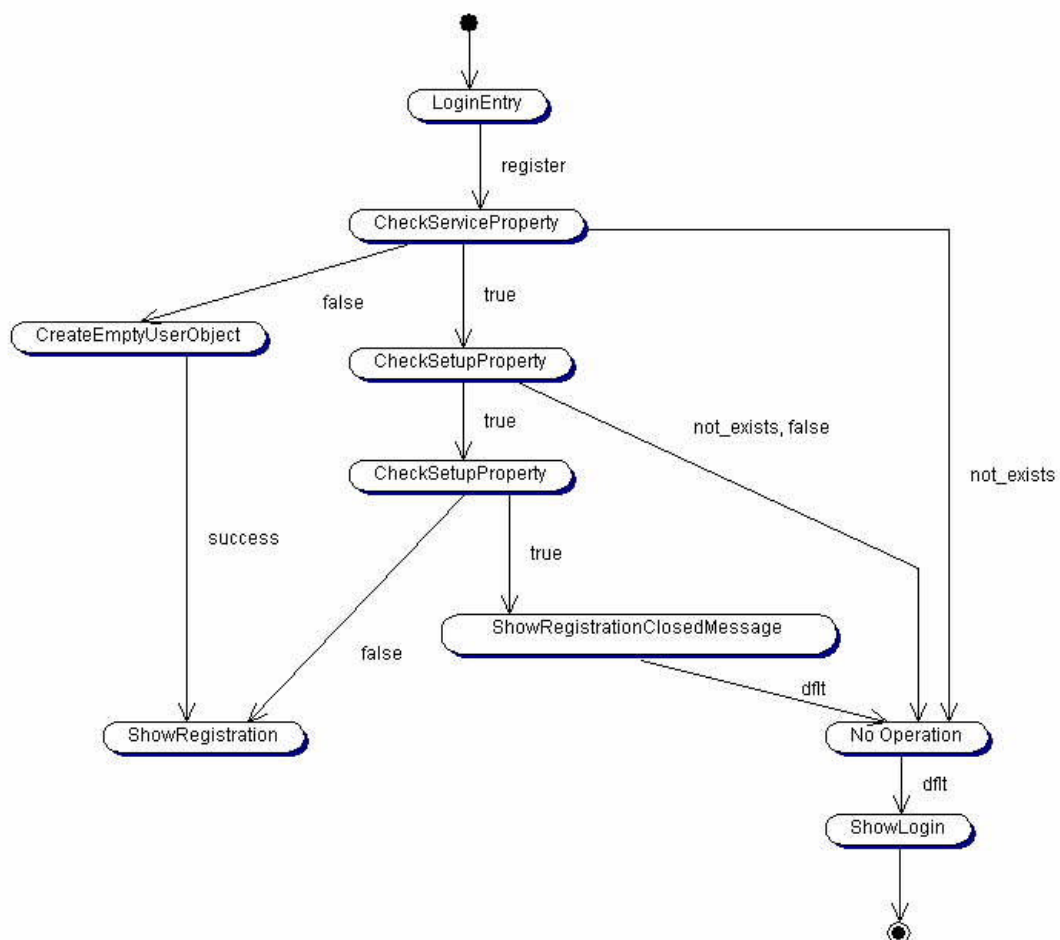


Abbildung 3.40: Activity Chart der Registration, Zielzustand

ein "CheckSetupProperties" SIB in den Graphen vor der Registration eingefügt worden. Dieses SIB überprüft den Wert "RegistrationClosed" auf "true". Trifft dieses zu, wird eine Seite angezeigt, die den Benutzer darüber informiert, dass die Registration für den Online Conference Service nicht möglich ist. Andernfalls wird der bisherige Graph verfolgt und die Seite der Registration angezeigt.

Testphase

In der Testphase wurde die Funktionalität der neu hinzugefügten Seiten im Setup und der Registration, sowie die Berücksichtigung der getroffenen Einstellungen im gesamten Dienst getestet. Da der OCS stark Rechte- und Rollenorientiert ist, wirkt sich insbesondere die Zuweisung der PC Member Rolle zu bestehenden PC Chairs in weiten Teilen des Dienstes automatisch auf die zur Verfügung stehenden Funktionen aus. Die Zuweisung der zusätzlichen Rolle wurde mit mehreren Logins getestet.

Beim Testen der Sperrung der Registration wurde insbesondere darauf geachtet, dass bei allen auf die Registrationsseite führenden Kanten des Graphen auch der neue Wert in der "Setup.properties" Datei überprüft wird. Die Sperrung der Artikeleinreichung wurde mit mehreren Logins getestet, wobei auch Logins mit mehreren zugewiesenen Rollen getestet wurden.

Die Entwürfe aus der Modellierungsphase mussten während der Implementation mehrmals überarbeitet werden, da sich der ursprünglich geplante Ablauf des Author Setup zu stark an den schon vorhandenen Ablauf des PC Chair Setup anlehnte. Dieser stellte sich als unnötig komplex für die Implementation des einfach gehaltenen Author Setup heraus. Die geplanten Modifikationen am PC Chair Setup konnten in der Implementation weitgehend wie geplant umgesetzt werden.

3.2.10 Sub-Delegation Feature

Beschreibung der Aufgabe

Das Feature „Report List“ soll um die Möglichkeit erweitert werden, sich alle Sub-Reviewer¹ des Systems übersichtlich anzeigen zu lassen. Es gibt zwei Gruppen von Sub-Reviewern: Zum einen diejenigen, die im System unter der Rolle „Sub-Reviewer“ registriert sind und zum anderen diejenigen, die manuell vom PC Member bei einem Review eines Artikels in die Eingabemaske eingetragen wurden.

Modellierungsphase

Da zu dieser Aufgabe kein Ist-Workflow existiert, wird in der Modellierungsphase nur auf den Soll-Workflow eingegangen.

Ein Benutzer, dem es erlaubt ist, sich die Report List und die Liste aller Sub-Reviewer anzuschauen, gelangt von der Report List zur Übersicht aller Sub-Reviewer. Von der Liste aller Sub-Reviewer kann er nun eine CSV²-Ansicht aufrufen (siehe Abbildung 3.41). Durch Anschalten des Features „Get list of all sub-reviewers“ kann der PC Chair einzelnen Rollen erlauben, sich die Liste aller Sub-Reviewer anzeigen zu lassen (siehe Abbildung 3.42). Zusätzlich zu den bereits vorhandenen Features in der Report List ist „GetSubReviewers“ das einzig neue. Das System muss dort die Sammlung und Aufberei-

¹Sub-Reviewer sind spezielle PC Members. Sie beurteilen einen Artikel, der ihnen von einem PC Member delegiert worden ist

²comma separated value = durch Kommata getrennte Werte

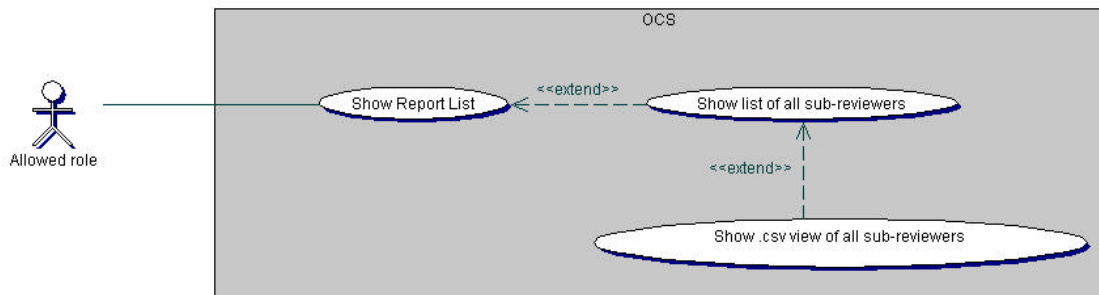


Abbildung 3.41: Use Case Diagramm "Sub-Reviewer anzeigen"

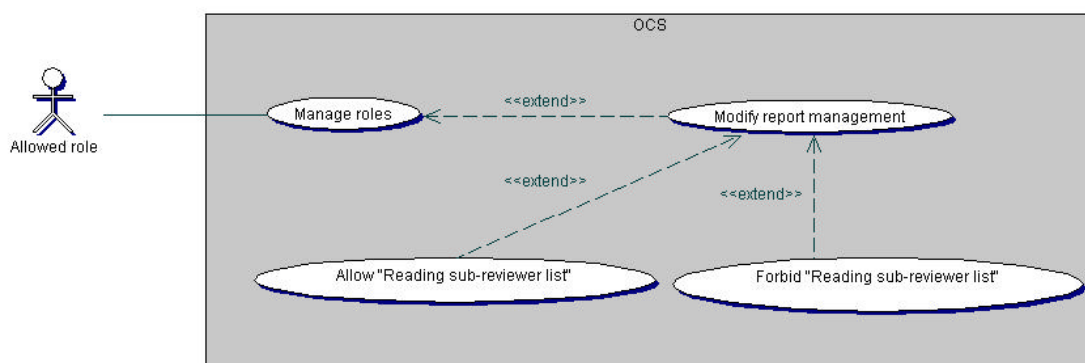


Abbildung 3.42: Use Case Diagramm "Rolleneigenschaften ändern"

tung der Daten für die Anzeige von „ShowSubReviewerList“ vornehmen. Der Benutzer kann dort wählen, ob er eine CSV-Ansicht bekommen möchte, die er dann in seinem Browser speichern kann (siehe Abbildung 3.43). Der PC Chair hat die Möglichkeit,

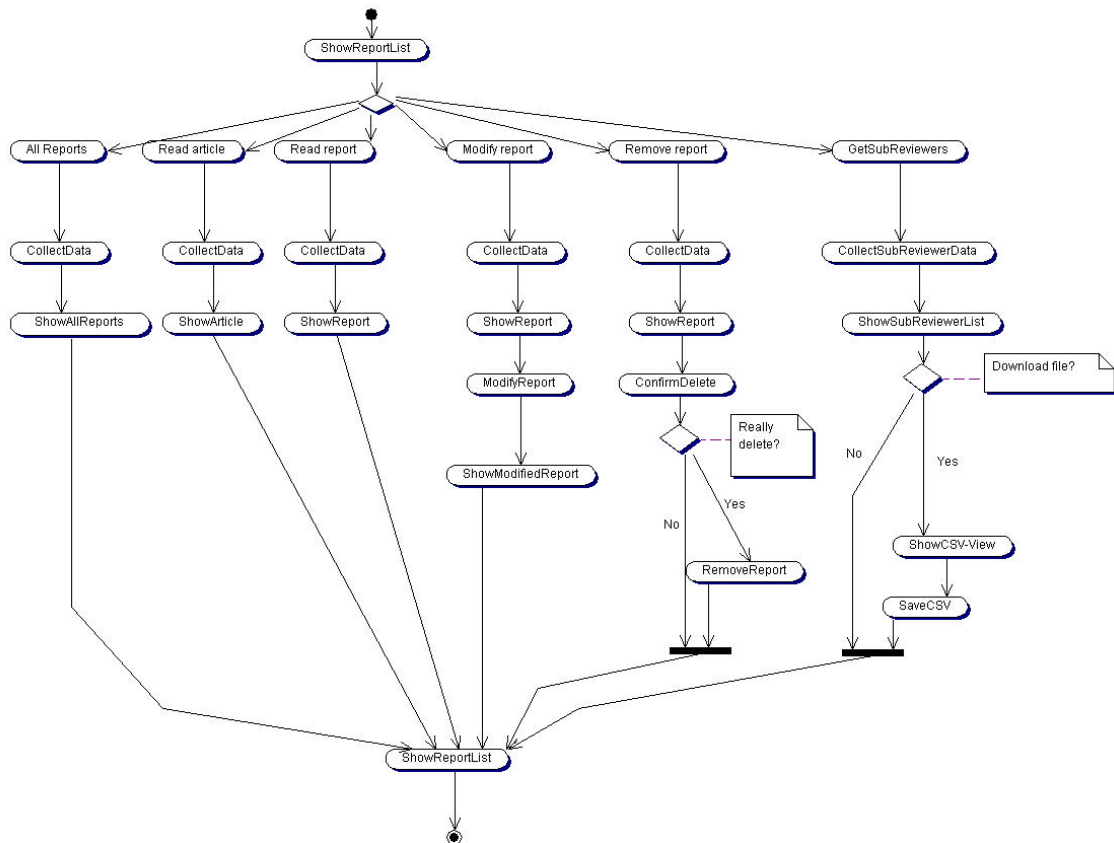


Abbildung 3.43: Activity Chart "erweiterte Report List" des "Sub-Delegation Feature"

einzelnen Rollen die Anzeige aller Sub-Reviewer zu erlauben oder zu verbieten (siehe Abbildung 3.44).

Implementierungsphase

Folgende Dateien (SIBs und WML) wurden verändert bzw. neu erzeugt:

- rolepermissions.inc (verändert)
- reportlist-cont.wml (verändert)
- CollectSubReviewerData (neu)
- subreviewer-cont.wml (neu)

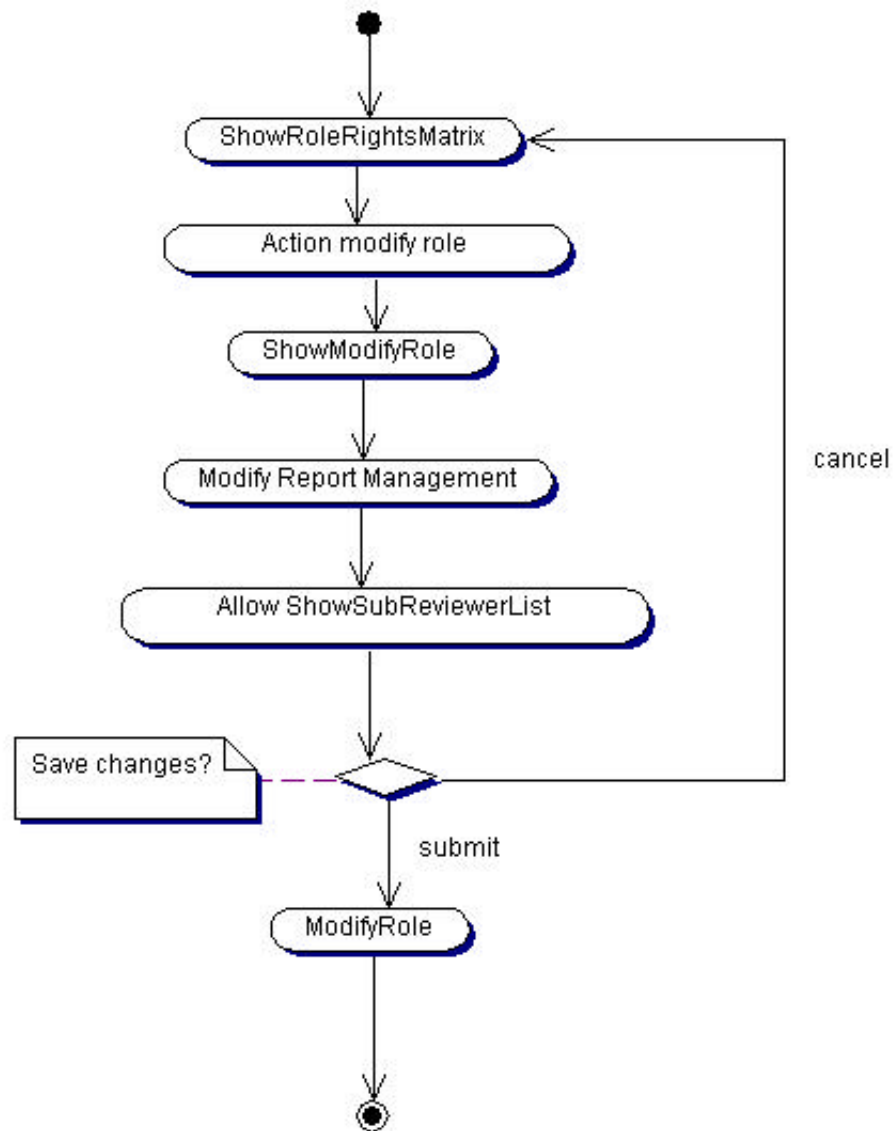


Abbildung 3.44: Activity Chart "Rolle modifizieren" des "Sub-Delegation Feature"

- subreviewer-download.wml (neu)

Um an alle Sub-Reviewer des Systems zu gelangen, müssen zwei Schritte unternommen werden. Zum einen müssen die registrierten Benutzer der Rolle „Sub-Reviewer“ bestimmt werden. Zum anderen existieren Sub-Reviewer, die manuell bei einem Review eines Artikels durch einen PC Member eingetragen wurden. Um an die nicht registrierten zu gelangen, muss über alle Reports iteriert werden, um so alle dort eingetragenen Sub-Reviewer zu bekommen. Es kann natürlich dadurch zu doppelten Einträgen kommen, so dass man diese Daten zusammenführen muss.

Die aufbereiteten Daten werden in drei Feldern angezeigt: Vorname, Nachname und E-Mail. Hier existiert aber noch ein Problem, welches nicht einfach gelöst werden kann: Wenn ein PC Member manuell Sub-Reviewer einträgt, kann er sie nur in ein Textfeld eingeben. Wenn er mehrere eintragen möchte, trennt er sie nach Konvention mit einem Komma. Es kann nun aber sein, dass eine Person nicht nur einen Vor- und einen Nachnamen hat, so dass das System nicht wissen kann, wo hier der Vor- und Nachname getrennt werden soll. Nach Konvention wird hier immer am ersten vorkommenden Leerzeichen getrennt.

Die Daten werden in folgender Reihenfolge gesammelt: Zuerst werden alle registrierten Sub-Reviewer ausgelesen, danach wird über alle Reports iteriert und verglichen, ob für diesen manuell eingetragenen Benutzer schon ein registrierter vorhanden ist. Falls nicht, wird er der Liste hinzugefügt. So ist gesichert, dass keine Daten der registrierten Benutzer verloren gehen (z.B. hat ein manuell eingetragener Benutzer keine E-Mail-Adresse).

Die gesammelten Daten werden als ein Array von EditorialUser Objekten gespeichert, so dass nach Wunsch Eigenschaften des Benutzers zur angezeigten Seite hinzugefügt werden können.

Wenn der Benutzer eine CSV-Ansicht bekommen möchte, kann er auf den entsprechenden Button klicken. Das Array wird an diese Seite weitergereicht, so dass es nur von dem WML-Template anders angezeigt werden muss. Um die Programmierung nicht zu kompliziert werden zu lassen, muss der Benutzer diese Ansicht als .csv-Datei speichern. Sie wird ihm nicht direkt als Download angeboten.

Um dieses Feature einer bestimmten Rolle zuzuweisen, muss der PC Chair in der Rollenkonfiguration das Report-Feature „F-REV-12“ auswählen.

Um die Anzeige zu testen, wurden zwei Reports mit drei manuell eingetragenen Sub-Reviewern angelegt. Zusätzlich wurden zwei Sub-Reviewer registriert, die auch einen Artikel beurteilten. Es existieren doppelte Einträge, die herausgefiltert werden müssen. Wie man in Abbildung 3.45 sehen kann, zeigt die Übersicht wie vorgesehen alle verschiedenen Sub-Reviewer, sortiert nach den Nachnamen. Doppelte Einträge wurden herausgefiltert. Die CSV-Ansicht in Abbildung 3.46 präsentiert die gleichen Daten wie in Abbildung 3.45, nur werden sie hier anders angezeigt.

Testphase

Die Aufgabe wurde laut den Anforderungen erfüllt, es bleibt lediglich das Problem,

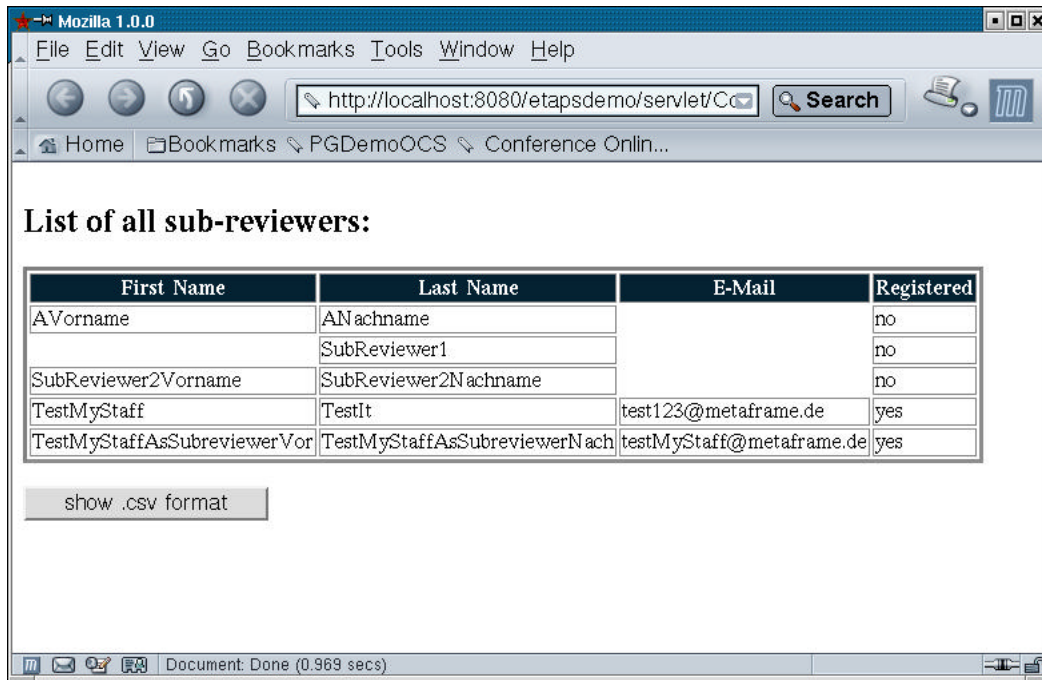


Abbildung 3.45: Screenshot "Sub-Reviewer anzeigen"

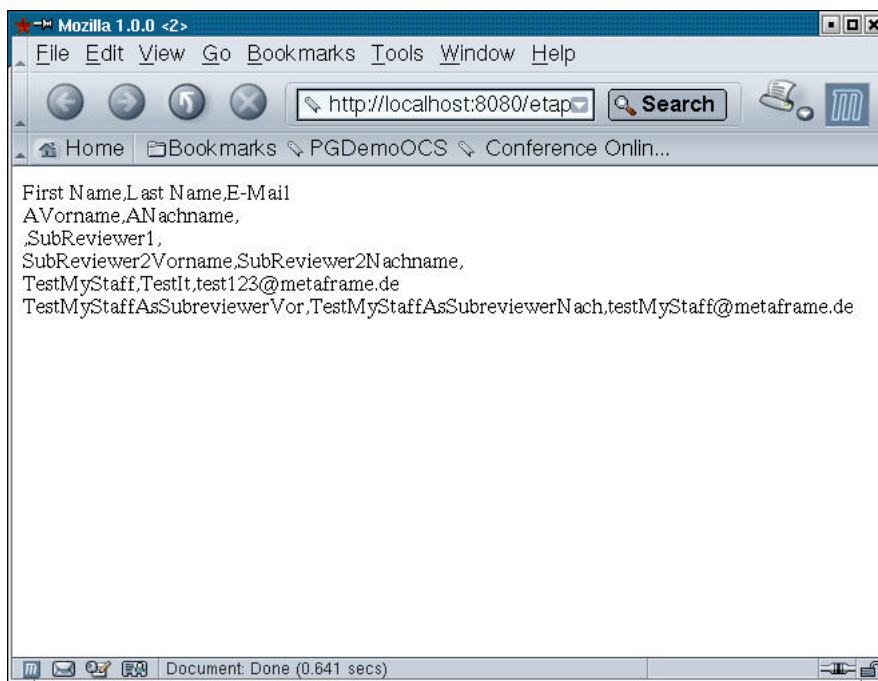


Abbildung 3.46: Screenshot "CSV-Ansicht der Sub-Reviewer anzeigen"

wie man Vor- und Nachname sinnvoll auftrennt. Beim Testen ließen sich keine Fehler erkennen.

3.2.11 User Feature

Beschreibung der Aufgabe

Die Aktion „remove user“ löscht einen Benutzer ohne vorher eine Bestätigungsseite anzuzeigen. Das führte bereits dazu, dass versehentlich Benutzer gelöscht wurden. Das Löschen von Benutzern, welche im Konferenzprozess involviert sind, wird bereits überprüft und verhindert.

Es soll eine Erweiterung des Workflows durch eine zusätzliche Bestätigungsseite bereitgestellt werden, welche das Profil des zu löschenden Benutzers anzeigt. Zusätzlich sollen kleine Modifikationen des bestehenden Workflows durchgeführt werden.

Modellierungsphase

Dieses Use Case Diagramm (Abbildung 3.47) zeigt die Aktion „Remove User“ an,

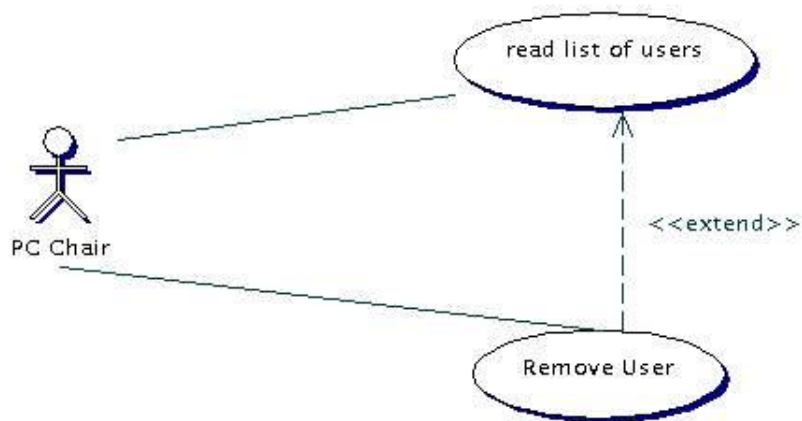


Abbildung 3.47: Use Case „Remove User“

welche nur durch den PC Chair ausgeführt werden kann.

Nach dem Anklicken des „Users“ Buttons im Hauptmenü des Online Conference Services kann sich der PC Chair eine Liste von allen Benutzern ansehen. Die Aktion „Remove User“ erlaubt es dem PC Chair, die Benutzer aus der Datenbank zu löschen.

Das Activity Chart (Abbildung 3.48) beschreibt, wie ein Benutzer aus der Datenbank

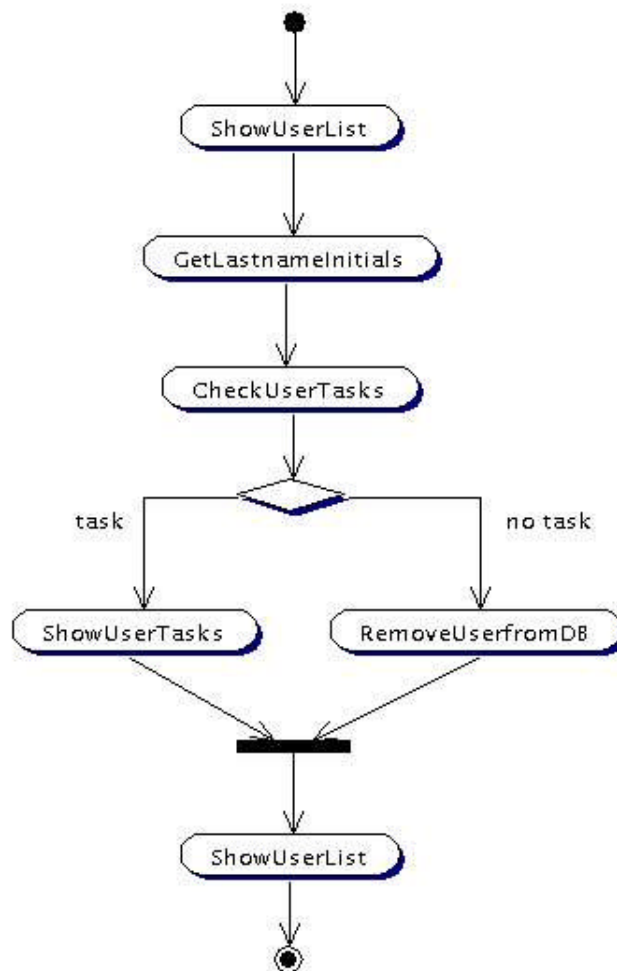


Abbildung 3.48: Activity Chart "Remove User" Ist-Zustand

entfernt wird. Nach dem Auswählen eines Buchstaben werden diejenigen Benutzer, deren Nachnamen mit diesem Buchstaben anfangen, in einer HTML Seite angezeigt. Dieser Buchstabe wird dann in einer Map abgespeichert. Falls ein Benutzer ausgewählt ist, wird überprüft, ob ihm eine Aufgabe zugewiesen worden ist. Wenn ja, darf er nicht gelöscht werden. Danach springt der Service wieder an den Überblick aller Benutzer zurück. Wenn das nicht der Fall ist, wird der Benutzer direkt aus der Datenbank entfernt. Dieser Vorgang sollte verhindert werden, falls der PC Chair versehentlich auf "remove user" klickt und nicht wirklich den Benutzer löschen will.

Außerdem wird am Ende des Workflows nochmal der Überblick aller Benutzer angezeigt,

aber nicht der zuvor ausgewählte Teilüberblick. Dies soll auch modifiziert werden. Im Vergleich zu dem obigen Diagramm in Abbildung 3.48 wurden ein paar Änderungen in dem Aktivitätsdiagramm in Abbildung 3.49 vorgenommen. Wenn der gewählte Benutzer keine Aufgaben hat, kann der PC Chair seinen Wunsch zum Entfernen des Benutzers in einer HTML Seite bestätigen. Am Ende des Workflows muß eine Liste der Benutzer angezeigt werden, deren Nachnamen mit dem vorher gespeicherten Buchstaben anfangen.

Implementierungsphase

Das Macro "user" (siehe Abbildung 3.50) wird durch die Kante „user“ vom Service-Entry SIB erreicht. Das Macro gehört zur Klasse „OCS“ und der Startpunkt des Graphs ist „ClearTempSessionValues“. Im Graph „Conference.slg“ wurde der User-Baum jetzt durch das „user“ Macro ersetzt. Die neuen SIBs in dem Macro (siehe Abbildung 3.51) sind „CheckUserTasks“, „ShowConfirmUserRemove“ und „RemoveUserFromDB“.

SIB „CheckUserTasks“ : Dieser SIB überprüft ob einem Benutzer irgendeine Aufgabe zugewiesen worden ist. Ist dies der Fall, wird das Entfernen des Benutzers verhindert.

SIB „ShowConfirmUserRemove“ : Dieser SIB zeigt das Profil des zu löschenden Benutzers in einer HTML Seite an. Der PC Chair kann hier den Wunsch zum Entfernen dieses Benutzers bestätigen. „User_ID“ wird als ein versteckter Parameter in der HTML Seite festgehalten und kann nachher für das Entfernen des Benutzers wieder verwendet werden.

SIB „RemoveUserFromDB“ : Dieser SIB entfernt einen Benutzer aus der Datenbank. Dabei ist die „User_ID“ im Call Context unentbehrlich.

HTML Seite „confirmuserremove-cont.html“ : Diese neue HTML Seite zeigt das Profil des zu löschenden Benutzers an, wie z.B. Name, Rollen, User_ID, Kategorie und Email Adresse. Der PC Chair kann den „ok“ Button anklicken, wenn er diesen Benutzer wirklich entfernen möchte.

Testphase

Alle erwünschten Funktionalitäten sind realisiert und der Test wurde erfolgreich durchgeführt.

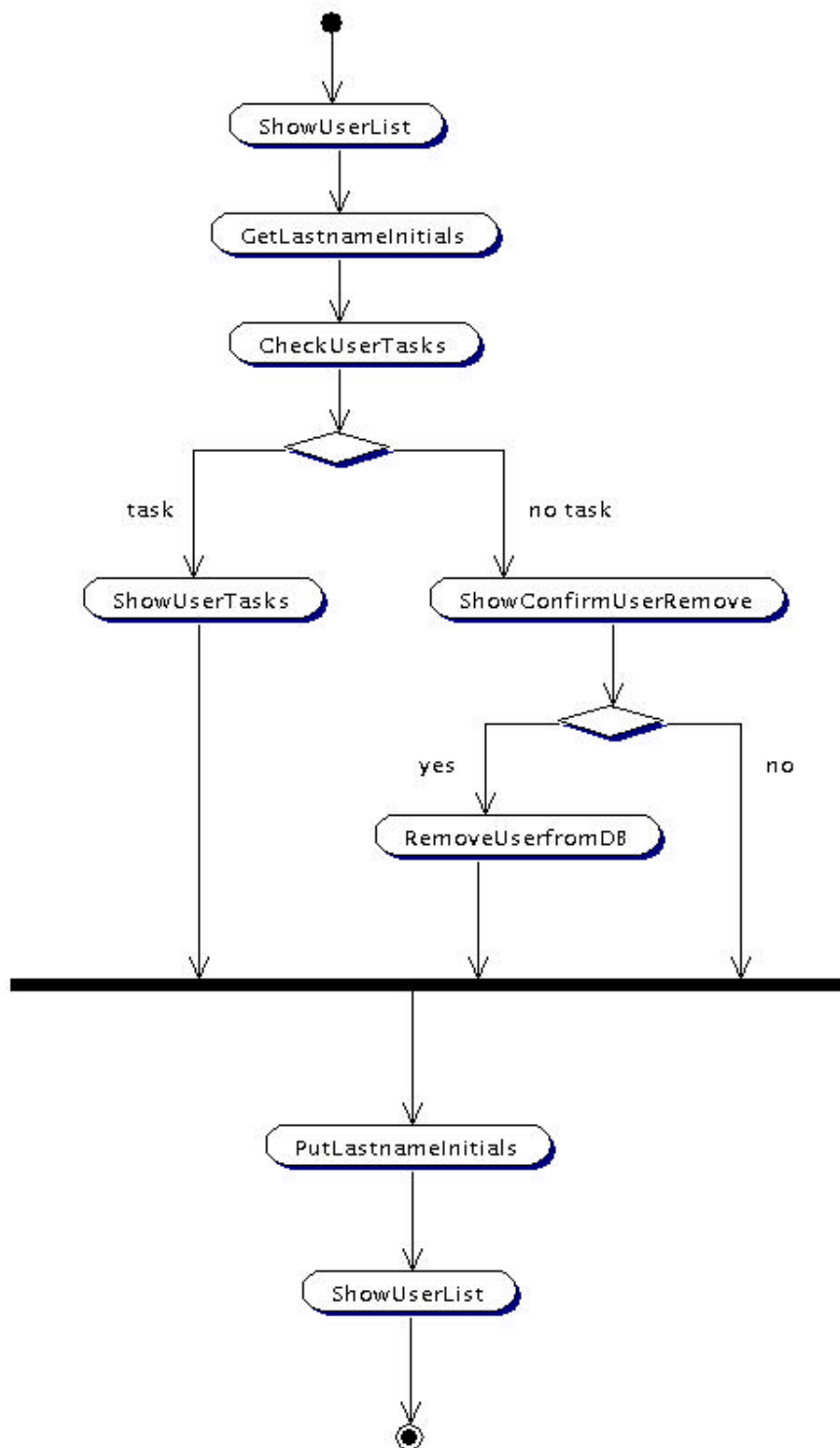


Abbildung 3.49: Activity Chart "Remove User" Soll-Zustand

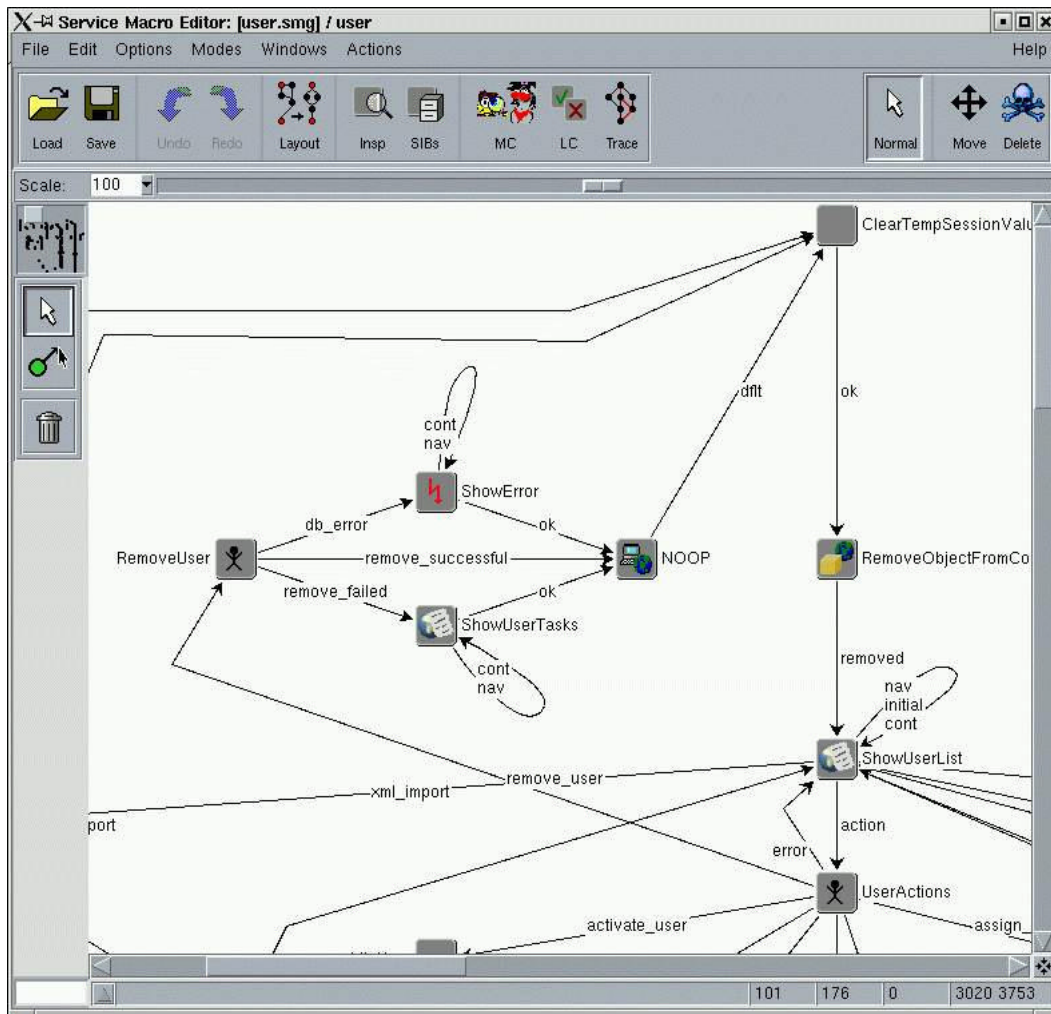


Abbildung 3.50: Macro „user“_alt ”User Feature“

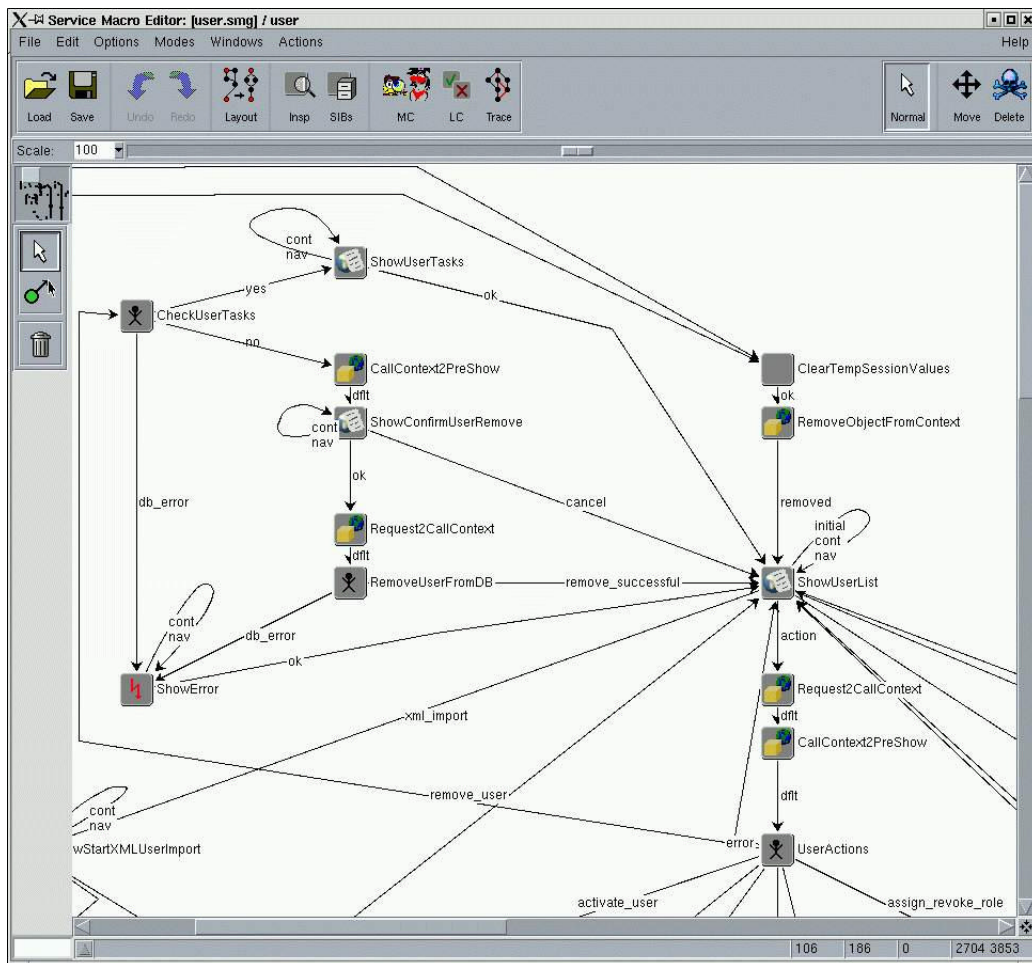


Abbildung 3.51: Macro „user“_neu ”User Feature“

3.3 Gruppenkurzthemen

3.3.1 Abstract submission workflow

Beschreibung der Aufgabe

Es soll einem Autor möglich sein zu dem bestehenden Article-Submission Workflow einen Abstract-Submission Workflow zu verwenden. In dem neuen Workflow soll der Autor zunächst nur den Abstract des Artikels (zusammen mit Meta-Informationen) einreichen und diesen zu einem späteren Zeitpunkt mit der Artikel-Datei vervollständigen, um ihn in eine vollwertige Article-Submission umzuwandeln. Dieser Workflow soll als zusätzliche Option, neben der alten Weise Artikel einzureichen, in das System integriert werden.

Hierzu sind einige Änderungen am OCS nötig:

1. Ein neuer Artikel-Status "abstract" muss in das System aufgenommen werden
2. Eine neue Deadline für das Einreichen der Abstracts soll eingerichtet werden und dadurch auch ein neuer Conference-Status "AbstractSubmissionDeadline"
3. Ein neuer Workflow für das Einreichen von Abstracts muss hinzugefügt werden (submit new abstract)
4. Ein neuer Workflow für das Modifizieren von Abstracts muss hinzugefügt werden (modify abstract)
5. Ein neuer Workflow für das Umwandeln eines Abstracts in einen vollwertigen Artikel muss hinzugefügt werden (submit full paper)
6. Die Role-/Rights-Matrix soll die neuen Features F-ART-16 zum Einreichen von neuen Abstracts und F-ART-17 zum Modifizieren von Abstracts enthalten
7. Das neue Feature soll auch über das Setup ein- und ausschaltbar sein (im Autor-Setup)

Modellierungsphase

Wie bei den meisten Software-Entwicklungen weicht das tatsächliche Resultat der Arbeiten von dem in der Modellierungs-Phase geplanten Bild stark ab. In diesem Fall zeigt sich dies besonders. Es soll hiermit darauf hingewiesen werden, dass dieser Abschnitt die Aufgabe, wie sie in der Modellierungs-Phase geplant war zeigt. Gravierende Änderungen am Design sind in diesem Kapitel an den jeweiligen Stellen beschrieben, aber auch zusätzlich im Kapitel **Implementierungsphase**.

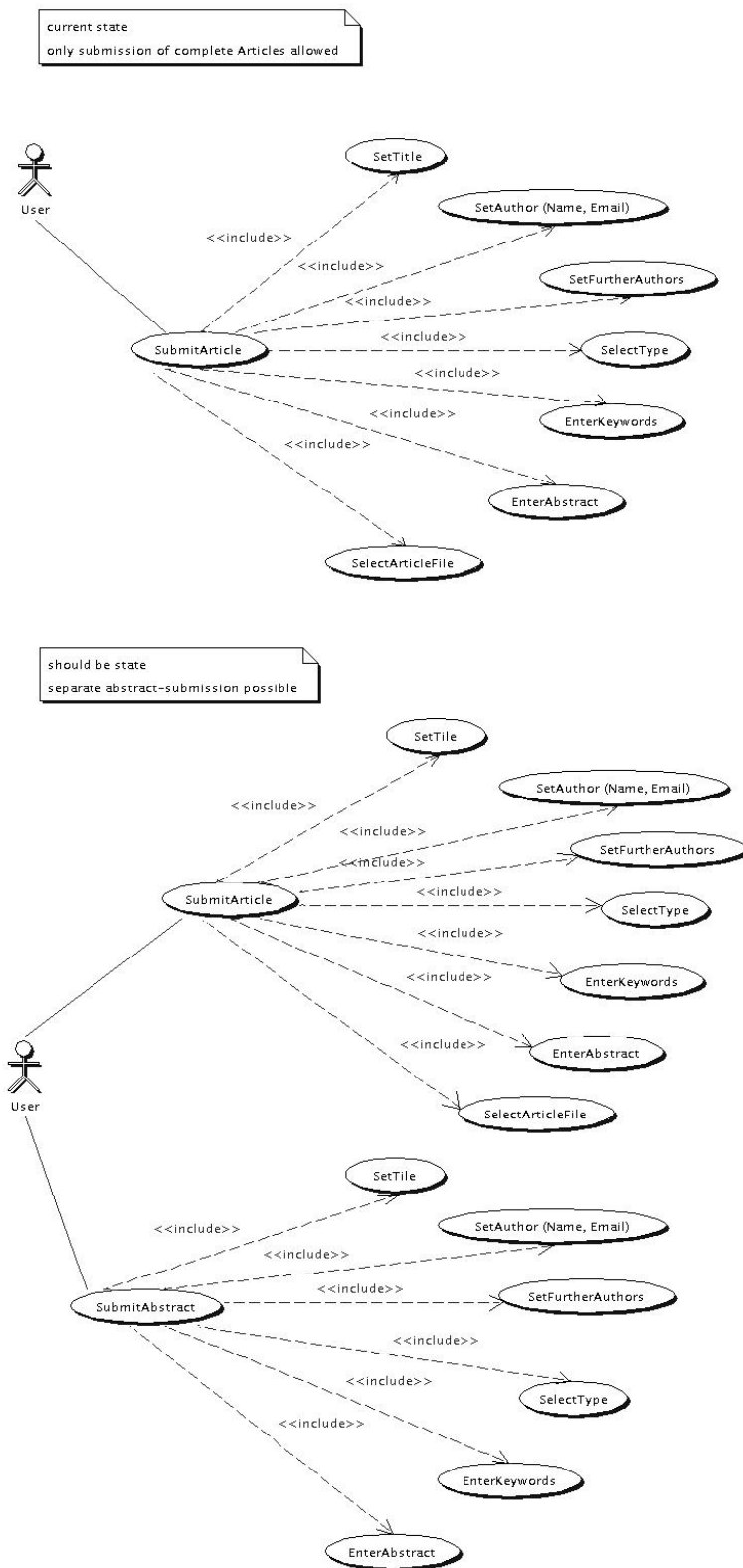


Abbildung 3.52: Use Case SubmitArticle "Abstract submission workflow"

Use Case Diagramme

Im Use Case "SubmitArticle" (Abbildung 3.52) ist in dem jetzigen Zustand nur der "SubmitArticle" Workflow zum Einreichen eines eigenen Artikels verfügbar, der eine Reihe von Eingaben erfordert (Abbildung 3.52 oben). Wie im darunterliegenden Diagramm (Abbildung 3.52 unten) gezeigt, soll der neue "SubmitAbstract" Workflow parallel dazu existieren und lediglich über kein Datei-Upload-Feld verfügen.

Hinweis: Während der Implementierungs-Phase wurde entschieden, dass das Abstract-Submission Formular ein Datei-Upload Feld enthält, in dem eine optionale Datei mit dem Abstract verschickt werden kann.

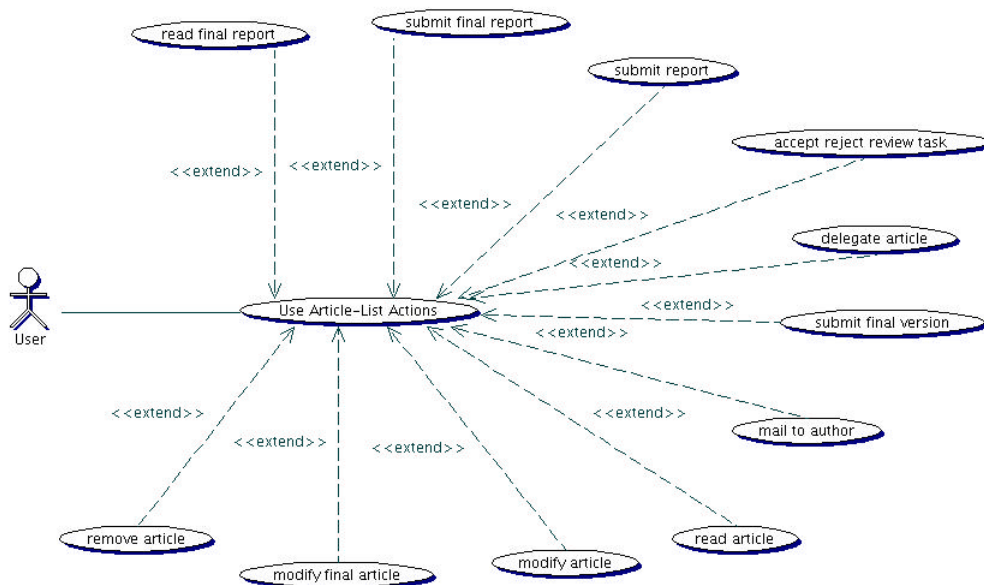


Abbildung 3.53: Use Cases: Mögliche Aktionen die für einen Artikel ausgeführt werden können (in der Actions-Spalte der Artikel-Tabelle)

Hinweis: Während der Design-Phase wurde geplant, die Aktionen wie in Abbildung 3.53 gezeigt zu belassen. D.h. keine neuen Aktionen hinzuzufügen und keine zu entfernen. Letztendlich wurde in der Implementierungsphase die Aktionsliste um zwei Punkte

erweitert: "modify abstract" (zum Modifizieren von Artikeln im Abstract-Status F-ART-17) und "submit full paper" (zum Komplettieren von Artikeln im "Abstract-Status", wofür das Feature F-ART-03 "submit own new article" benötigt wird).

Nachdem das neue Feature (F-ART-16) zu der Role-/Rights-Matrix hinzugefügt worden ist (Abbildung 3.54 unten) wird es möglich sein dieses über zwei Wege ein- und auszuschalten. Zum einen über das Setup (Punkt 4 Autor) "configure the role author", zum anderen über die "modify role" und "create role" Workflows.

Hinweis: Während der Implementierungs-Phase wurde entschieden ein weiteres Feature F-ART-17 hinzuzufügen, über das das Modifizieren von Abstracts erlaubt oder verboten werden kann.

Activity Chart

SubmitAbstract (Abbildung 3.55): Dieser Workflow ist neu, obwohl er stark auf dem bestehenden SubmitArticle Workflow basiert. Der wichtigste Unterschied ist, dass in dem SubmitAbstract Workflow keine Artikel-Datei versandt wird. Das SubmitAbstract-Formular fragt lediglich nach Meta-Informationen (Autoren, Schlagwörtern, Kurz-Zusammenfassung). Der zweite Unterschied ist, dass der resultierende Artikel den Status "Abstract" zugewiesen bekommt, im Gegensatz zu dem Status "Submitted". Das neue Feature soll über die Artikel-Seite erreichbar sein.

Constraints:

1. F-ART-16 ("Allow submission of abstracts in advance") ist für die Rolle des jeweiligen Users erlaubt.
2. Die Deadline zum Einreichen von Abstracts ist nicht abgelaufen.

Hinweis: Während der Implementierungs-Phase wurde entschieden ein Datei-Upload Feld in dem Abstract-Submission Formular zu belassen, in dem eine optionale Datei mit dem Abstract verschickt werden kann. Das Textfeld für den Abstract ist jedoch trotzdem ein Pflichtfeld. Ausserdem werden Benachrichtigungs-E-mails an den PC Chair und den Autor während des Workflows abgeschickt (dazu wurden neue Email-Vorlagen eingerichtet).

ReadArticleShouldBeState (Abbildung 3.56): Der ursprüngliche "ReadArticle" Workflow muss kaum verändert werden. Er muss lediglich mit Abstract-Artikeln zurecht kommen, die keine Dateien besitzen.

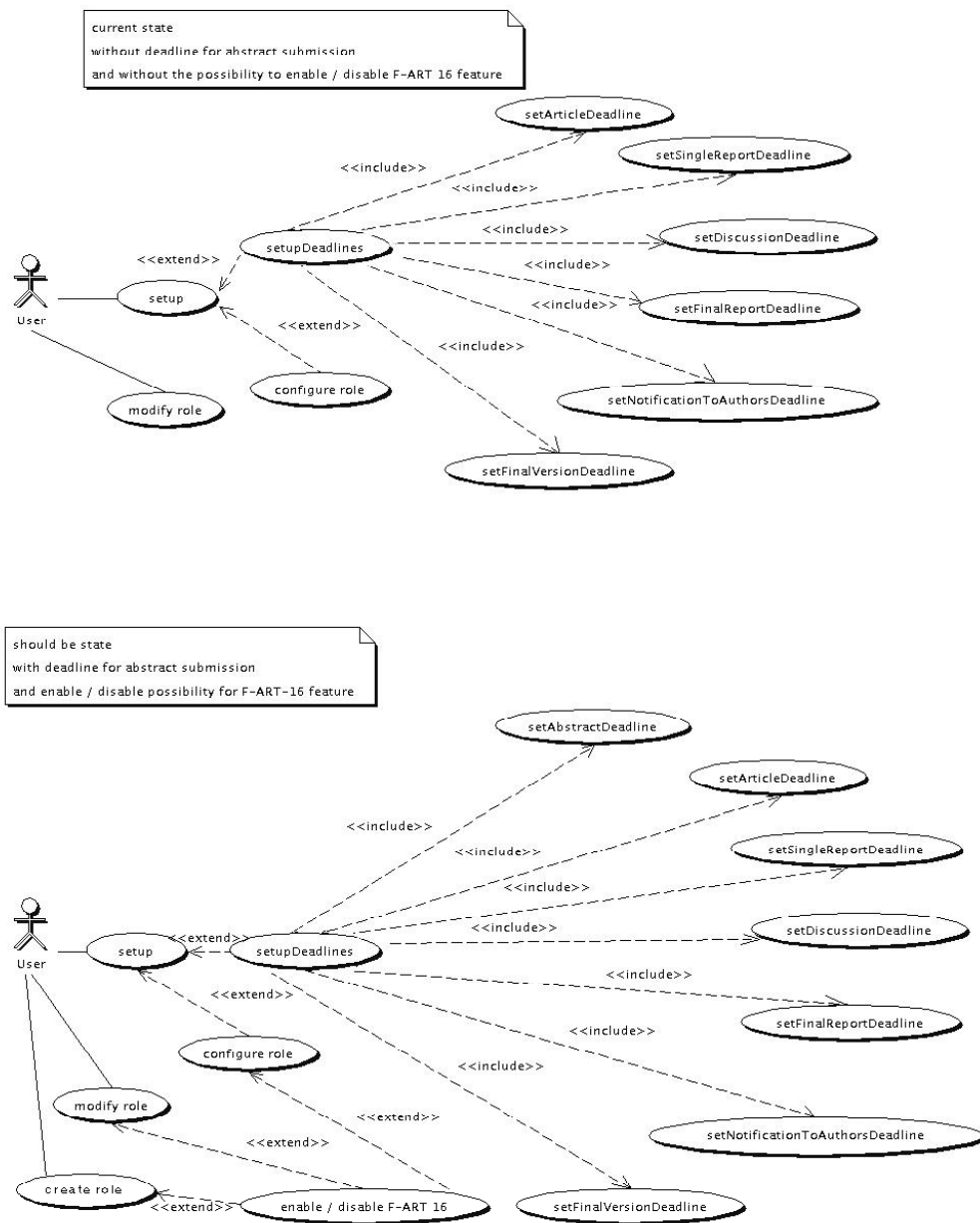


Abbildung 3.54: Abstract-Submission im Setup

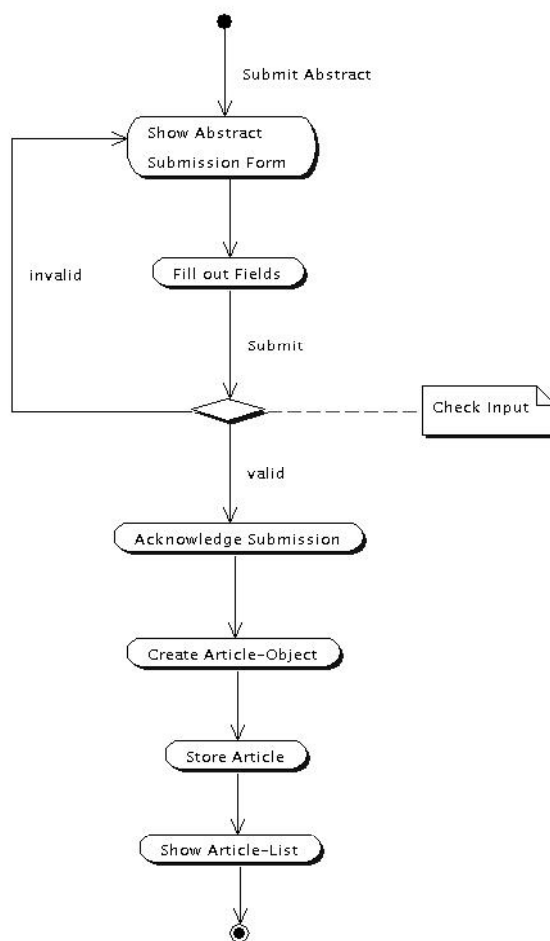


Abbildung 3.55: Activity Chart "SubmitAbstract"

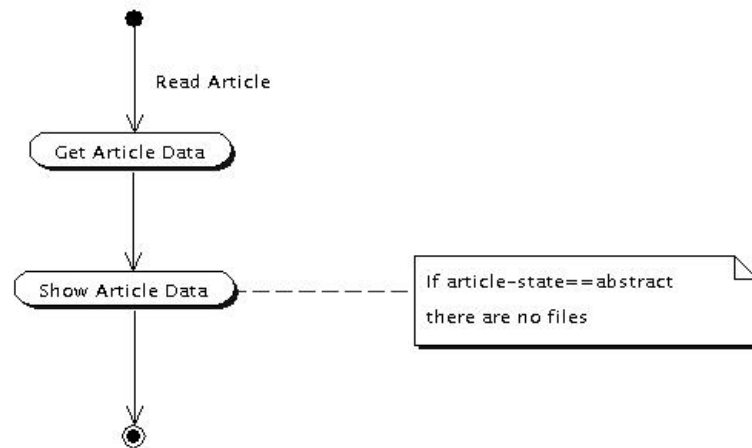


Abbildung 3.56: Activity Chart "ReadArticleShouldBeState"

Constraints:

1. F-ART-05 ("Read article") ist erlaubt für die Rolle des jeweiligen Users und der Artikel gehört in eine der erlaubten Kategorien ("own", "delegated" oder "all").
2. F-ART-06 ("Get list of articles") ist erlaubt für die Rolle des jeweiligen Users (andernfalls gibt es keine Links zum lesen der Artikel).

ModifyDeadlines (Abbildung 3.57): Der bereits bestehende Workflow zum Einrichten der Deadlines muss um die neue Deadline "Abstract submission deadline" erweitert werden (diese muss in der Liste an oberster Stelle stehen). Zur Zeit enthält das "ModifyDeadlines"-Formular sechs Deadlines, die modifiziert werden können (die "article submission", die "single report", die "discussion", die "final report", die "notification to authors" und die "final version deadline").

Constraints:

1. F-Setup ("ModifyDeadlines" ist ein Setup-Feature) ist erlaubt für die Rolle des jeweiligen Users.

ModifyRole (Abbildung 3.58): In der Rubrik "Article Management" der Roll-/Rights-Matrix muss das Feature F-ART-16 "submit new abstract in advance" zum Einreichen von eigenen Abstracts hinzugefügt werden.

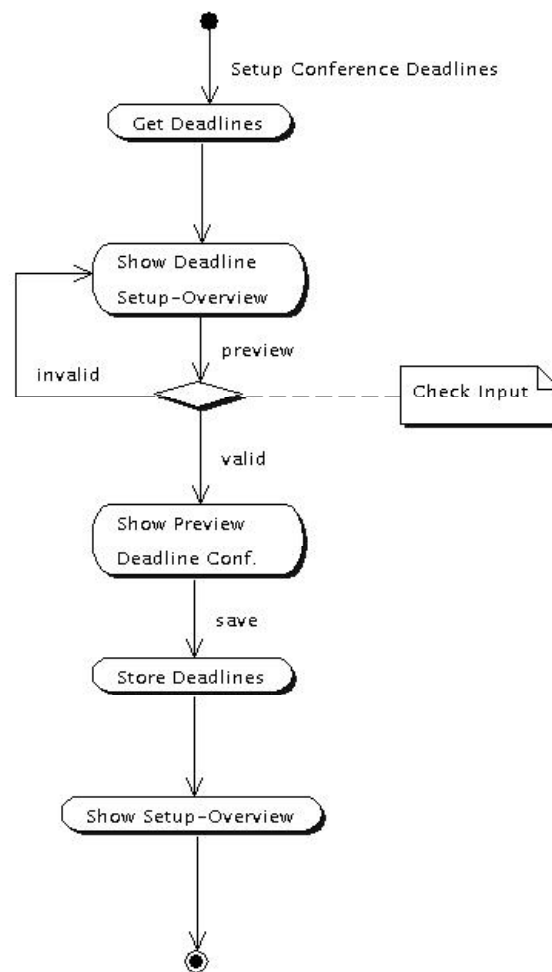


Abbildung 3.57: Activity Chart "ModifyDeadlines"

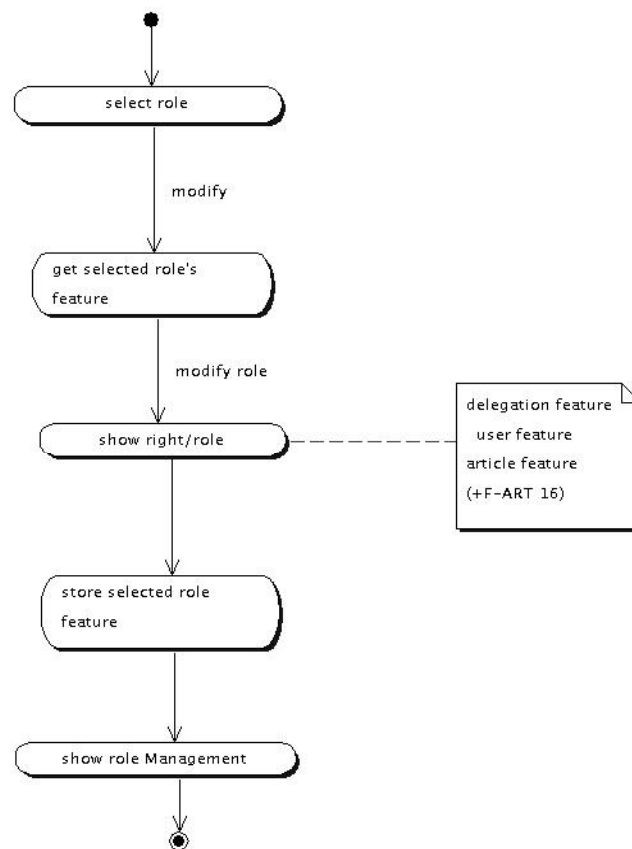


Abbildung 3.58: Activity Chart "ModifyRole"

Constraints:

1. "F-Role-02" ("modify an existing role") ist erlaubt für die Rolle des jeweiligen Users.
2. "F-Role-07" ("get list of all existing roles") ist erlaubt für die Rolle des jeweiligen Users. Andernfalls ist es nicht möglich eine Rolle zum Modifizieren auszuwählen.

Hinweis: Während der Implementierungs-Phase wurde entschieden ein weiteres Feature F-ART-17 hinzuzufügen, über das das Modifizieren von Abstracts erlaubt oder verboten werden kann.

ModifyArticle (Abbildung 3.59): Das ModifyArticle-Formular enthält alle Meta-Informationen zu dem Artikel (die Eingabe Felder sind vorgefüllt). Nachdem das Formular ausgefüllt und abgeschickt worden ist, muss bei Abstract-Status Artikeln geprüft werden, ob eine Datei mitgeschickt wurde. Falls ja, soll der Status des Artikels in "Submitted" übergehen.

Constraints:

1. F-ART-03 ("submit own new article") ist erlaubt für die Rolle des jeweiligen Users.
2. F-ART-06 ("get list of articles") ist erlaubt für die Rolle des jeweiligen Users.
3. Die Article-Submission Deadline muss noch gültig sein.

Hinweis: Ursprünglich war es geplant den ModifyArticle Workflow dazu zu verwenden, um einen Abstract-Artikel zu komplettieren (mit dem eigentlichen Artikel-Dokument). Während der Implementierungsphase wurde jedoch entschieden den normalen ModifyArticle Workflow komplett zu sperren für Abstract-Artikel. Ein komplett neuer Workflow "SubmitFullPaper" zum Komplettieren eines Abstracts wurde erstellt, der ebenfalls aus der Aktionsliste eines Artikels wählbar ist (nur für Artikel mit dem Status "Abstract").

Zum Modifizieren von Abstracts wurde ein neuer Workflow "ModifyAbstract" (F-ART-17) hinzugefügt, der keine Statusänderungen vornimmt. Nur der SubmitFullPaper Workflow kann dazu verwendet werden den Status von "Abstract" auf "Submitted" zu ändern.

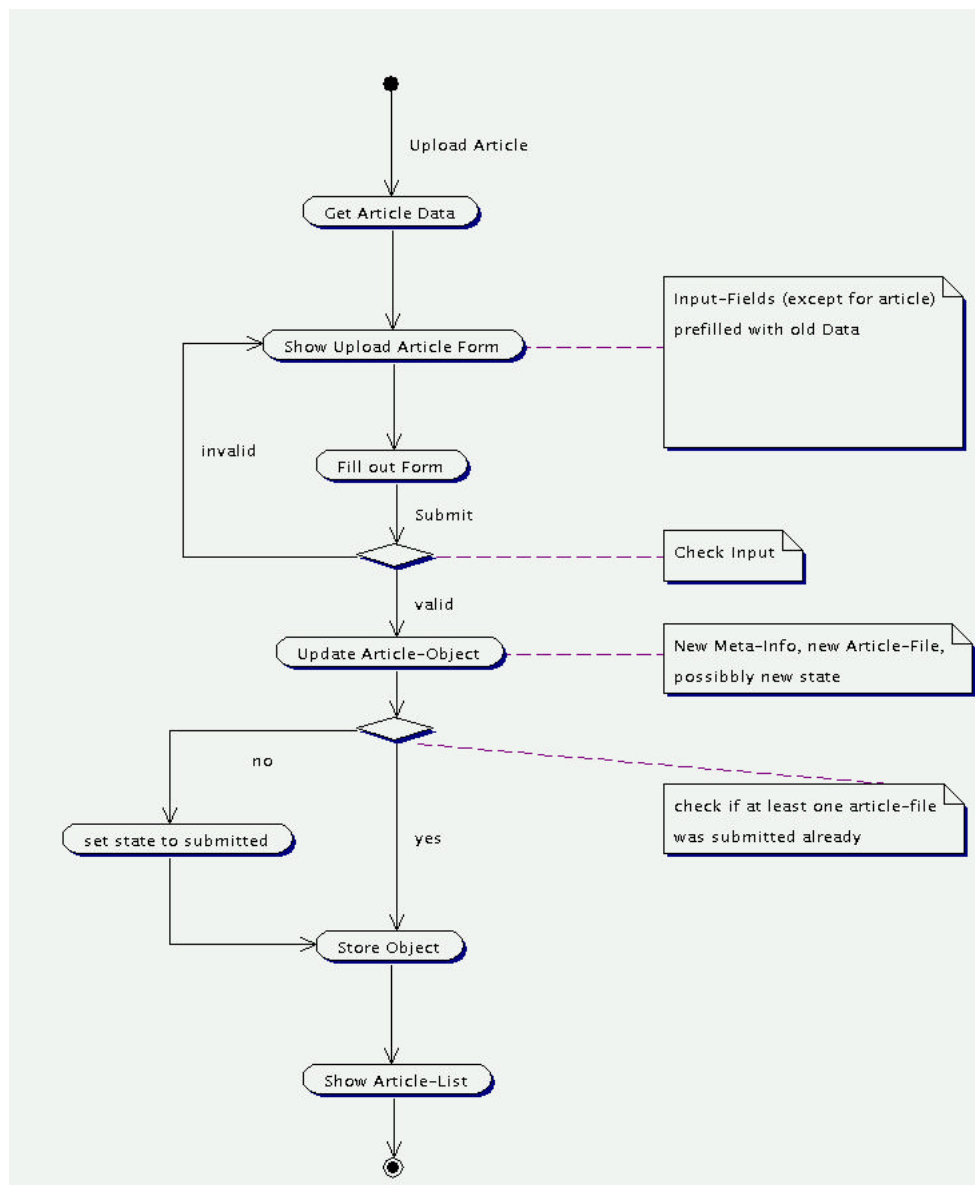


Abbildung 3.59: Activity Chart "ModifyArticle"

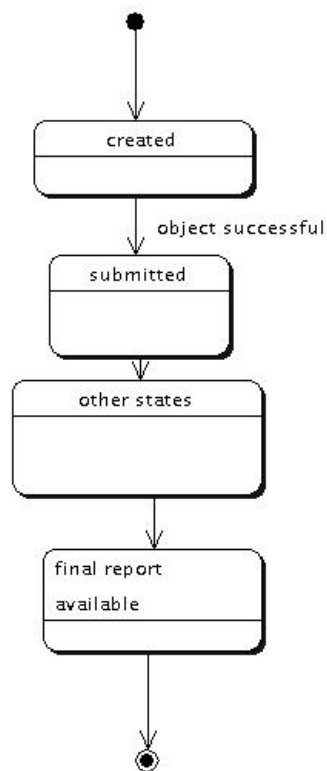


Abbildung 3.60: Activity Chart "ArticleStateChanges", Ist-Zustand

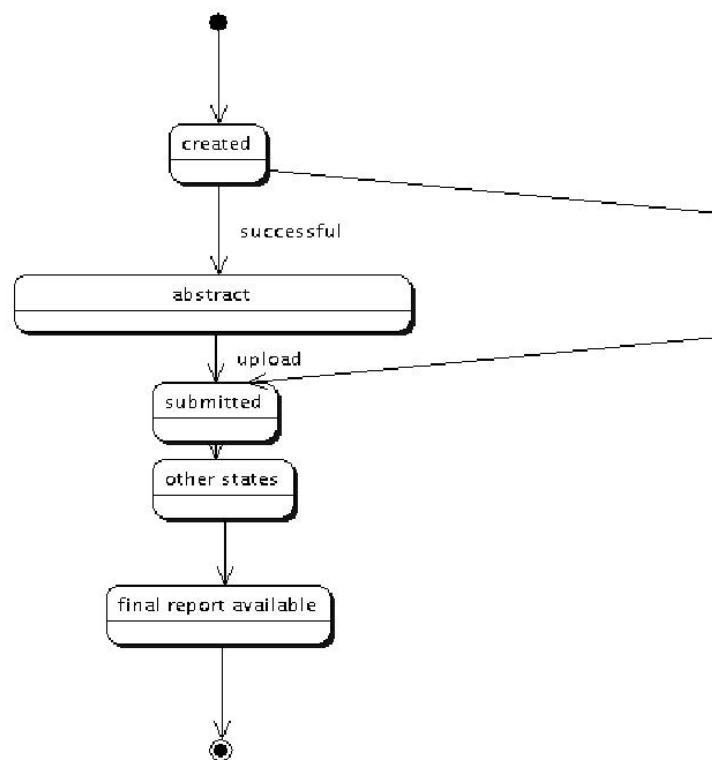


Abbildung 3.61: Activity Chart "ArticleStateChanges", Soll-Zustand

ArticleStateChanges (Abbildungen 3.60, 3.61): Wenn ein Artikel erzeugt wird befindet er sich zu aller erst im Zustand "Created". Verläuft das Einreichen des Artikels ohne Fehler, wird der Zustand auf "Submitted" gesetzt. Der neue Zustand "Abstract" soll zwischen "Created" und "Submitted" eingefügt werden. Wird der Abstract-Submission Workflow verwendet, so befindet sich der neue Artikel nach dem erfolgreichen Workflow im Zustand "Abstract". Wenn dann der Workflow SubmitFullPaper für den Artikel durchgeführt wird, wird der Zustand "Submitted" erreicht. Bei einer normalen Article-Submission wird der neue Zustand übersprungen.

Implementierungsphase

Zwar wurde ursprünglich geplant, die Abstract-Submission entsprechend der Article-Submission zu gestalten, allerdings ohne das Datei-Upload Feld, dies wurde aber revidiert und der Gruppe mitgeteilt, sie solle ein Datei-Upload Feld im Formular belassen, um dem Autor zu erlauben ein Dokument, das den Abstract enthält mit abzuschicken. Dieses Feld sollte optional sein. Ausserdem wurde später entschieden, dass ein spezieller Workflow zum Modifizieren eines Abstracts nötig ist, und der normale ModifyArticle Workflow dazu nicht verwendet werden sollte. Weiterhin wurde die Idee verworfen das Kompletieren von Abstracts über den ModifyArticle Workflow abzuwickeln. Es sollte ein neuer SubmitFullPaper Workflow für Abstracts integriert werden, durch den das Artikel-Dokument eingereicht werden kann wodurch der Artikel in den Zustand "Submitted" übergeht. In dem neuen SubmitNewAbstract Workflow, als auch in dem neuen SubmitFullPaper Workflow, werden Email-Benachrichtigungen über den Vorgang an den PC Chair und an den Autor verschickt.

Testphase

Deadline: Es war nötig zu testen, ob der Workflow zum modifizieren der Deadlines mit der neu hinzugefügten Abstract Submission Deadline richtig funktioniert. Das System merkt sich alle Veränderungen (Datum und Beschreibung). Geht man noch einmal in das Deadline-Setup sind die neu eingetragenen Parameter ablesbar. Zudem muss das System mit dem neuen Conference-State "AbstractSubmissionDeadline" richtig funktionieren. Das heißt das OCS muss den Conference-State richtig anzeigen und die vom Conference-State abhängigen Rechte richtig setzen. Um das Erstere zu prüfen wurden einige Male die Deadlines verändert (abgelaufen/ noch gültig gesetzt) und auf der Login-Seite verglichen, ob sie die zugehörige Deadline-Beschreibung anzeigt, was der Fall war. Ist die AbstractSubmissionDeadline noch gültig, ist das Feature F-ART-16 "submit new abstract" gesetzt (und der Link auf der Artikel-Seite sichtbar). Falls die Deadline abgelaufen ist, ist der Link nicht mehr sichtbar. Ausserdem wurde geprüft, ob die Deadlines auch in der richtigen Reihenfolge funktionieren (wenn die erste abgelaufen ist, ist die zweite gültig und gibt den Conference-State vor).

Modifizierung der Rollen: Die Role-/Rights-Matrix funktioniert wie gewünscht. Modifiziert man eine Rolle (wählt die Features F-ART-16 und F-ART-17 an bzw. ab) werden die Änderungen übernommen (gespeichert) und die jeweiligen Features ein und ausgeschaltet (bei F-ART-16 am "submit new abstract" link, und bei F-ART-17 am "modify abstract" link zu erkennen, die falls die zugehörigen Permissions fehlen nicht angezeigt werden). Wird die Permission F-ART-16 über den Autor-Setup Workflow ein bzw. ausgeschaltet hat es den gleichen Effekt und umgekehrt.

Einreichen eines Abstracts: Falls der User die Erlaubnis hat, Abstracts einzureichen (F-ART-16) und die AbstractSubmissionDeadline noch nicht abgelaufen ist, ist der Link "submit new abstract" auf der Artikel-Seite sichtbar. Der Workflow funktioniert korrekt mit allen Eingaben. Wenn Pflichtfelder leer gelassen werden, gibt es entsprechende Fehlermeldungen. Bei optionalen Feldern gibt es dagegen keine Meldungen. Nach einer erfolgreichen Submission wird für den Artikel der Status "Abstract" angezeigt. Die Benachrichtigungs-E-mails an den PC Chair und an den Autor werden ebenfalls korrekt verschickt.

Modifizieren eines Abstracts: Falls der User die Erlaubnis hat Abstracts zu modifizieren (F-ART-17), wird bei Artikeln im Zustand "Abstract" innerhalb der Artikel-Tabelle in der Aktions-Spalte der Link "modify abstract" angezeigt. Im Formular können alle Felder verändert werden und eine neue Abstract-Datei kann abgeschickt werden (diese ersetzt die alte Datei, falls sie den selben Namen trägt, oder wird als neue Datei an die Liste angefügt). Der Zustand des Artikels bleibt dabei unverändert.

Submit Full Paper: Falls der User die Erlaubnis hat einen eigenen Artikel einzureichen (F-ART-03), wird in der Aktions-Spalte bei Artikeln im Zustand "Abstract" der Link "submit full paper" angezeigt. Der Workflow funktioniert wie gewünscht. Wenn ein Pflichtfeld (z.B. das Datei-Upload-Feld) leer gelassen wird, wird eine Fehler-Meldung angezeigt. Andernfalls kann der User den Workflow bis zum Ende ausführen und der Zustand des Artikels wird in "Submitted" geändert. Auch an dieser Stelle werden Email-Benachrichtigungen an den PC Chair und den Autor geschickt.

3.3.2 Bidding workflow

Beschreibung der Aufgabe

Dieses Gruppenkurzthema beschäftigt sich mit Themen, die das Bidding Feature erweitern und benutzerfreundlicher gestalten sollen. Es lässt sich in sechs Teilaufgaben einteilen:

1. **Synchronisation der Bidding- und Conflict Matrix:** Die Bidding Matrix zeigt nur die Biddings der PC Member an und zeigt damit keine Konflikte, die der PC Chair zuvor in der Conflict Matrix eingetragen hat. Um Konflikte aus der Conflict Matrix zu berücksichtigen, müssen diese in der Bidding Matrix angezeigt werden. Durch eine zusätzliche Schaltfläche unterhalb der Bidding Matrix soll es dem PC Chair ermöglicht werden, Konflikte aus der Bidding Matrix in die Conflict Matrix zu übertragen.
2. **Darstellung eigener Beiträge in der Bidding Liste:** Vom PC Member eingereichte Artikel werden in der Bidding Liste nicht angezeigt. Dieses führt unter Umständen zu Verwirrungen, weil manche PC Member annehmen, dass ihr Artikel gar nicht im System berücksichtigt wird. Die Bidding Liste wird dahingehend erweitert, dass die eigenen Artikel zwar angezeigt werden, aber kein Bidding darauf abgegeben werden kann.
3. **Darstellung des Delegationsstatus in der Bidding Matrix:** Damit der PC Chair den Status der bereits vergebenen Delegationen beim Delegieren über die Bidding Matrix berücksichtigen kann, soll der Status jeder Delegation mit dem Farbcode aus dem Delegation Feature kenntlich gemacht werden.
4. **Darstellung einer zurückgewiesenen Delegation in der Bidding Matrix:** Wenn ein PC Member eine zugewiesene Delegation zurückweist, wird dieses in der Bidding Matrix nicht kenntlich gemacht und es kann aus Versehen zu einer erneuten Delegation kommen. Durch den neu eingeführten Farbcode aus dem Delegation Feature wird nun zwar eine zurückgewiesene Delegation angezeigt, allerdings nur so lange, bis die Delegation im Delegation Feature entfernt wird. Das System soll diese Information aber darüber hinaus darstellen können.
5. **Konsistenz der Bidding Tabelle in der Datenbank sicherstellen:** Wenn ein PC Chair und ein PC Member gleichzeitig die Bidding Matrix und die Bidding Liste geöffnet haben, kann es zu Datenverlusten kommen, da beim Speichern nur komplette Datensätze geschrieben und damit eventuell in der Zwischenzeit vorgenommene Änderungen der jeweils anderen Rolle verworfen werden. Um Datenverlust zu vermeiden, sollen beide Features nur solche Datensatzfelder überschreiben, die für die jeweilige Änderung notwendig sind.
6. **Datum in der Email der Delegation korrigieren:** Im Anschluss an eine Delegation wird eine Email an den jeweiligen PC Member verschickt, welche ihn über die zugewiesene Aufgabe und die damit verbundenen Abgabetermine informiert. Beim Auslesen des Datums aus der dazugehörigen Template wird in der Java Klasse ein Objekt vom Typ Date erzeugt und der Inhalt dieses Objektes in die Email geschrieben. Dieses Datum wird von Java bei der Übergabe von "22.08.2004" mit dem Wert "22.08.2004 00:00:00" initialisiert. Dieses muss geändert werden, damit die Anzeige in der Email "22.08.2004 23:59:59" lautet.

Modellierungsphase

Um die Konflikte aus der Conflict Matrix in der Bidding Matrix darstellen zu können, müssen die vom PC Chair manuell gesetzten Konflikte aus der entsprechenden Datenbanktabelle ausgelesen und anschließend für die Verwendung im Bidding Matrix Template aufbereitet werden. Es werden nicht alle Informationen aus der Conflict Tabelle benötigt, daher werden diese Informationen, wie in Abbildung 3.62 gezeigt, gefiltert und in einer dafür geeigneten Datenstruktur in den Call Context geschrieben. Da jede Zelle der Bidding Matrix eine Kombination aus PC Member und Artikel bedeutet, werden die Konflikte in einer doppelten Hashtable abgelegt. Da die Tabelle zeilenweise aufgebaut wird, wobei jede Zeile einen Artikel repräsentiert und jede Spalte einen Benutzer, werden die Konflikte in einer Hashtable unter der Benutzer ID abgelegt. Diese Hashtable wird wiederum in eine weitere Hashtable gespeichert, der Schlüsselwert ist dabei die Artikel ID.

Das Template der Bidding Matrix wertet diese Informationen während der Erstellung der Tabelle aus und fragt zu jedem Artikel die zugehörigen Konflikte ab. Die Konflikte aus dem Conflict Feature werden, wie in Abbildung 3.63 gezeigt, durch ein kleines, rotes "c" kenntlich gemacht. Der Buchstabe wird innerhalb der Tabellenzelle zur jeweiligen Artikel- / Benutzerkombination zusätzlich zum eventuell vorhandenen Biddingstatus angezeigt.

Unterhalb der Bidding Matrix wird dem Benutzer eine Schaltfläche zur Verfügung gestellt, mit der er die von PC Mitgliedern angegebenen Konflikte in die Conflict Matrix übertragen kann. Dafür wird ein neuer SIB benötigt, welcher alle Biddings und alle Konflikte aus den jeweiligen Tabellen lädt und überprüft, ob ein Konflikt durch den PC Member gesetzt wurde. Ist dieses der Fall, wird überprüft, ob ein Konflikt in der Konflikttabelle existiert. Existiert kein Konflikt, wird ein neuer angelegt und gespeichert (Abbildung 3.64). Die erfolgreiche Übertragung der Konflikte wird dem Benutzer durch Ausgabe der Mitteilung "Saving of the conflicts was successful" oberhalb der Bidding Matrix angezeigt. Getestet wurde die Funktionalität der Implementation durch Anlegen neuer Benutzer, Artikel und Biddings. Als PC Chair eingeloggt wurden mehrere Konflikte in die Conflict Matrix eingetragen. In der Bidding Matrix wurden nun beide Arten von Konflikten angezeigt und auch das Übertragen von PC Member Konflikten in die Conflict Matrix funktionierte problemlos.

Um die vom zur Zeit eingeloggten PC Member eingereichten Artikel in der Bidding Liste anzuzeigen, muss zunächst sichergestellt sein, dass die Informationen über die eigenen Artikel zusammen mit den Informationen über die weiteren Artikel in den Call Context kopiert werden. Abbildung 3.65 zeigt die Vorgehensweise zur Anzeige der Artikel in der Bidding Liste abhängig von den aktiven Rechten des Benutzers. Das Template überprüft bei der Erstellung der Tabelle jeden Artikel und fügt bei Vorliegen eines eigenen Artikels statt der Radiobuttons für das Bidding den Kommentar "own article" in roter Schrift ein (Abbildung 3.66). Dadurch kann der PC Member kein Bidding für diesen Artikel abgeben. Beim Speichern des Bidding für die weiteren Artikel werden automatisch Konflikte für die eigenen Artikel in die Datenbank eingetragen, diese Konflikte werden

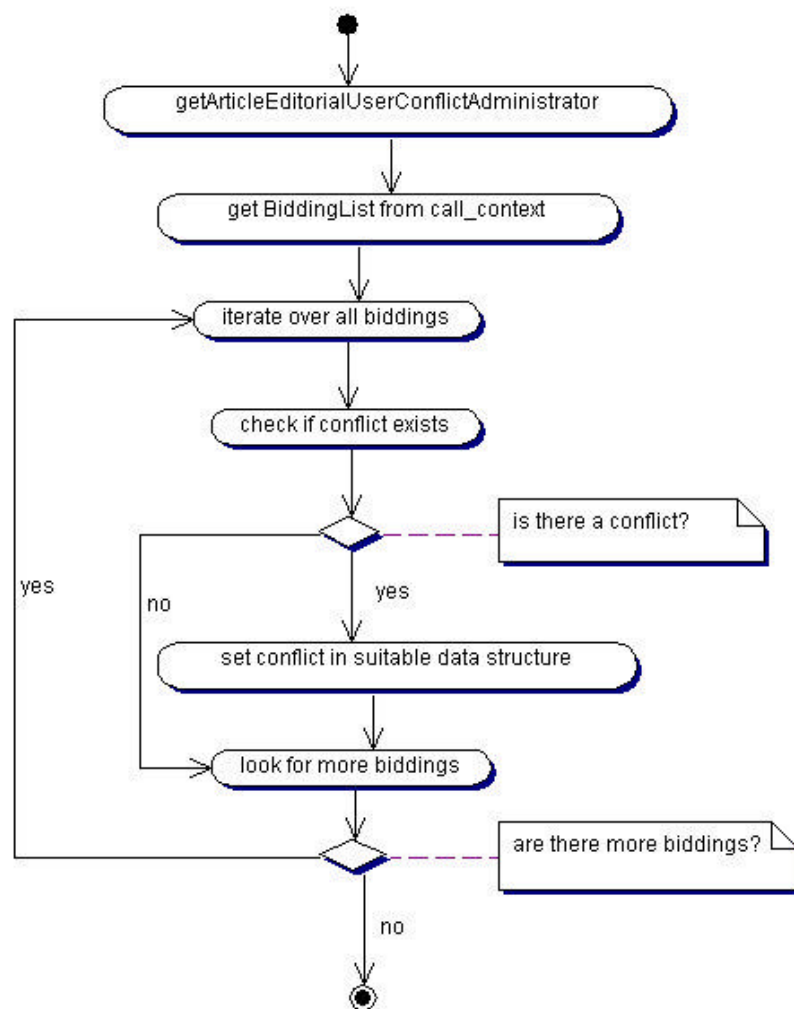


Abbildung 3.62: Auslesen von Konflikten aus der Conflict Matrix

ID	DP	TT	TA	ID	Sum:()
001	C		C	001	
002	B	c	C	002	
003	I		C	003	
004	I	R <input checked="" type="checkbox"/>	C	004	
005	C	<input type="checkbox"/>	B <input type="checkbox"/>	005	
Sum:					

Abbildung 3.63: Bidding Matrix mit Anzeige eines Konfliktes aus der Conflict Matrix und Verwendung der Farbcodes aus dem Delegation Feature

bei erneutem Aufruf der Bidding Liste neben dem Vermerk "own article" angezeigt und sind nicht veränderbar. Dieser automatisch eingetragene Konflikt ist gleichwertig zu einem normalen Bidding und wird daher wie ein gewöhnlicher Konflikt dem PC Chair in der Bidding Matrix angezeigt. Um die Anzeige der eigenen Artikel zu testen, wurden mehrere PC Member Logins angelegt und über diese jeweils mehrere Artikel eingereicht. Bei jedem Login wurde kontrolliert, ob die jeweils eigenen und die Artikel der anderen PC Member im Bidding Feature angezeigt werden. Auch die automatisch gesetzten Konflikte bei eigenen Artikeln wurden durch erneuten Aufruf des Bidding Feature, sowie anhand der Bidding Matrix des PC Chair überprüft.

Für die Darstellung des Delegationsstatus in der Bidding Matrix müssen alle Delegationen aus der Datenbank gelesen und der Status jeder Delegation in einer Datenstruktur abgelegt werden, die es ermöglicht, anhand der Kombination von Artikel ID und Benutzer ID den Status auszulesen. Diese Datenstruktur wird dann in den Call Context kopiert und von dem Template bei Erzeugung einer Zelle der Bidding Matrix abgefragt, um die Hintergrundfarbe der Tabellenzelle entsprechend zu setzen. Sinnvollerweise wird hier das zu einer Artikel-/Benutzerkombination gehörende Objekt ArticleDelegation zur Speicherung in der Datenstruktur verwendet. Zugewiesene Delegationen, die der PC Member noch nicht angenommen oder abgelehnt hat, sind gelb hervorgehoben. Angenommene und noch nicht abgeschlossene Begutachtungen sind hellgrün, abgeschlossene Begutachtungen dunkelgrün markiert. Lehnt ein PC Member eine Artikelbegutachtung ab, wird die dazugehörige Tabellenzelle dunkelrot hinterlegt. Delegationen, die der PC Chair in der Zukunft vornehmen möchte, diese aber noch nicht direkt durchführen will, werden weiterhin hellblau hinterlegt. Abbildung 3.67 zeigt eine Bidding Matrix mit mehreren PC Mitgliedern und Artikeln und den aus den verschiedenen Delegationsstadien resultierenden Farbcodes. Als Datenstruktur für den einfachen Zugriff auf den Delegationsstatus pro Artikel/Benutzer wird, wie bei der Anzeige der Konflikte, eine Hashtable verwendet. Auch hier wird der Status erst in einer Hashtable unter der PC Member ID abgelegt, diese dann wiederum in eine weitere Hashtable unter der Artikel ID geschrieben. Der Test

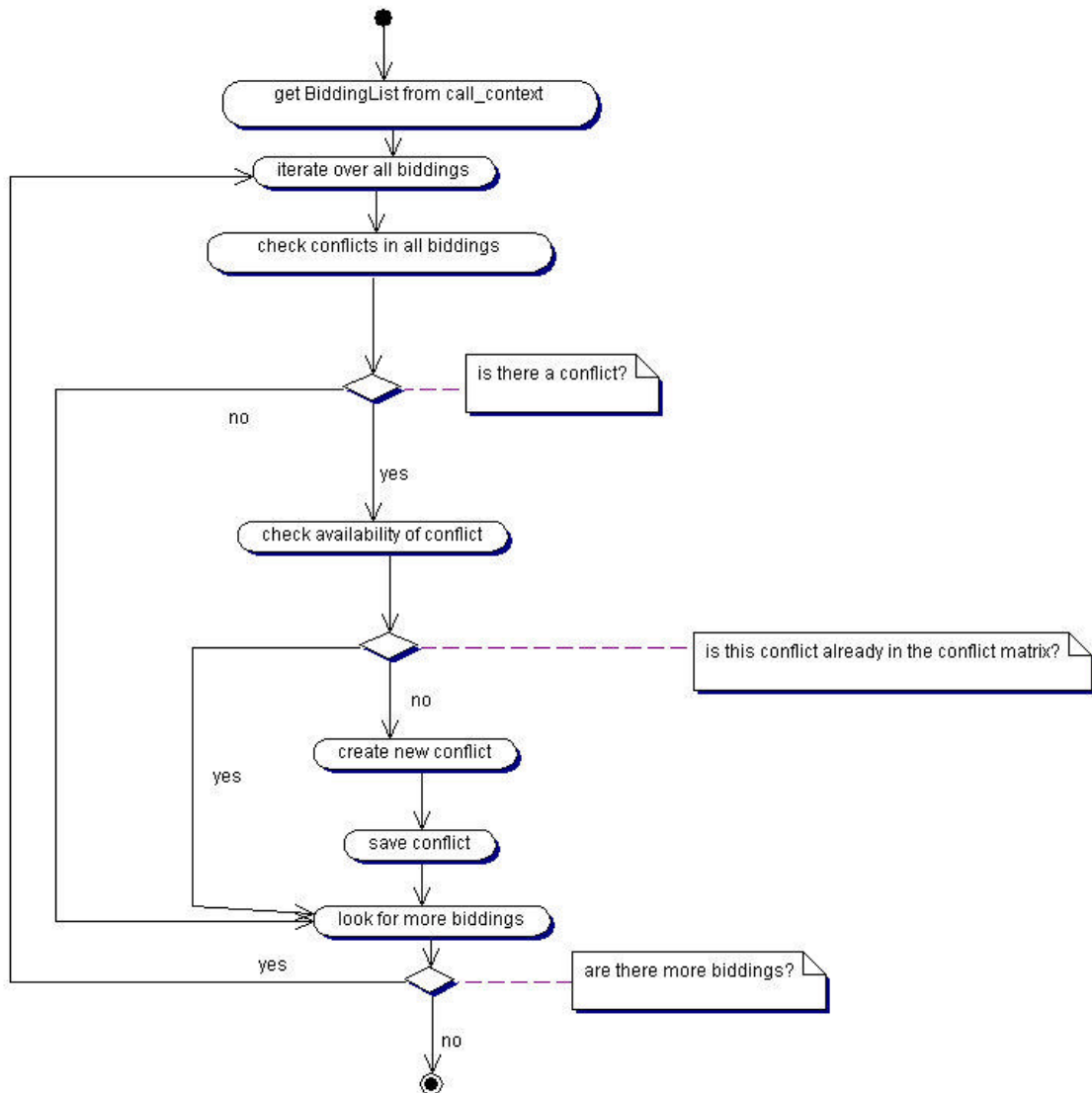


Abbildung 3.64: Übertragen von Bidding - Konflikten in die Conflict Matrix

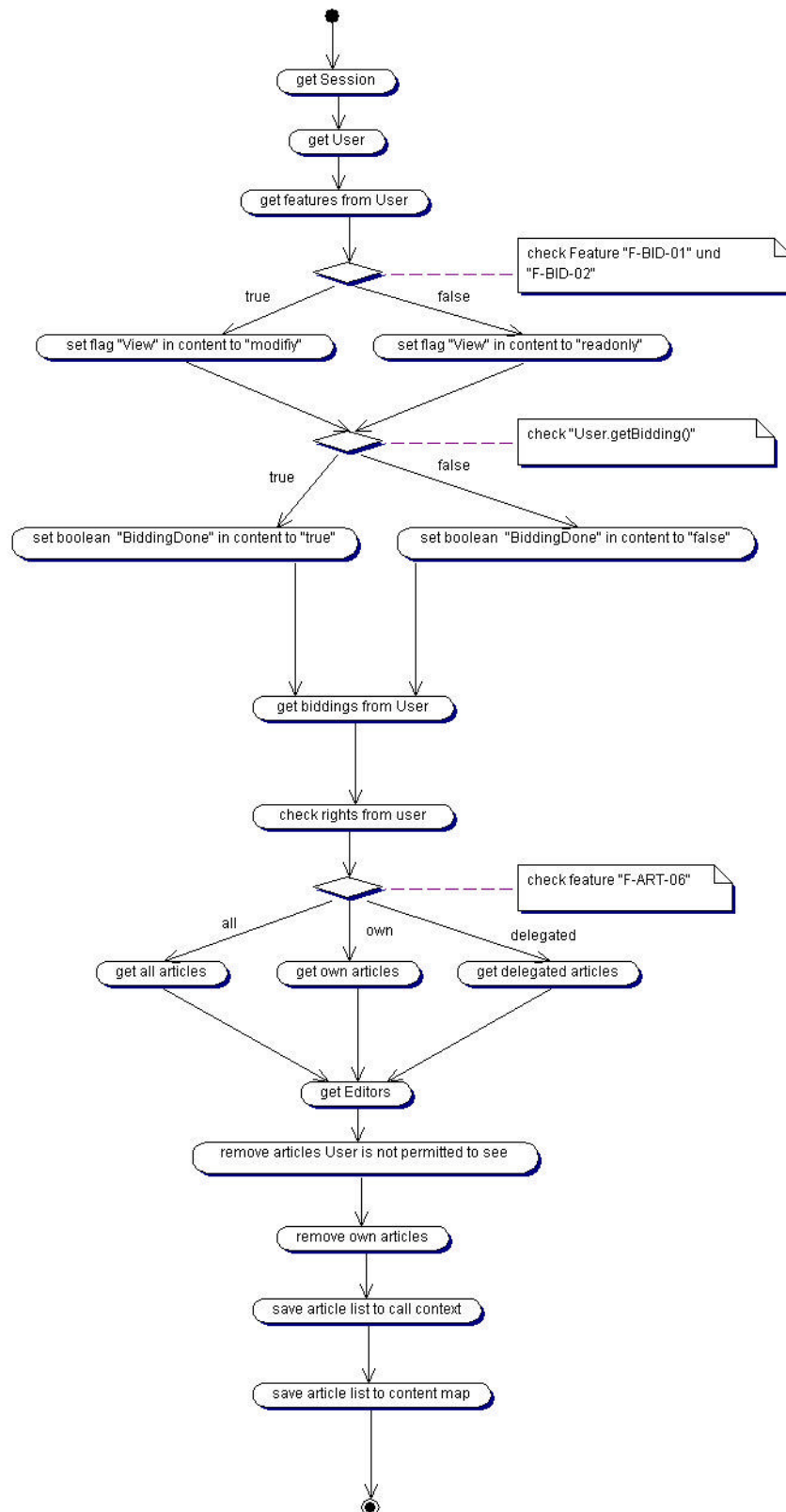


Abbildung 3.65: Activity Chart:Anzeige eigener Artikel

Bidding

This page allows you to select some articles you wish to review.

To indicate selection

- Use **B** ("bid") to indicate a strong interest in reviewing the corresponding articles.
- Use **I** (indifferent) to mark articles you don't bid for, but you would review if assigned.
- Use **U** (unable) to mark articles you do not feel sufficiently competent to review.
- Use **C** (conflict) to indicate articles you are in conflict with, e.g. if you are author or co-author yourself, or if authors of that article are quite "close" to you. Please refer in this case to the specific policy of this conference. In case of questions, please contact the PC Chair.
- If you submitted an article it will be shown in the bidding list and marked with **own article**. When you save your bidding a conflict is set automatically.

After clicking the "Continue" button the service leads you to a confirmation page, where you can check/modify your choice. Click on the article ID to get more information (abstract, file) about an article.

ID	Title	Corresp. Author	Further Authors	Bidding	Type
001	Testartikel Nr. 1	root	Testautor, Testautor2	<input type="radio"/> B <input type="radio"/> I <input type="radio"/> U <input checked="" type="radio"/> C	Regular Paper
002	Testartikel	Author, Test	Testfurther Author1	<input checked="" type="radio"/> B <input type="radio"/> I <input type="radio"/> U <input type="radio"/> C	Regular Paper
003	Testartikel zum Test SubReviewer	Author, Test		<input type="radio"/> B <input checked="" type="radio"/> I <input type="radio"/> U <input type="radio"/> C	Regular Paper
004	test artikel 3	Author, Test		<input type="radio"/> B <input checked="" type="radio"/> I <input type="radio"/> U <input type="radio"/> C	Regular Paper
005	testtitel	Pierlings, Daniel		C own article	Regular Paper

Abbildung 3.66: Bidding Liste mit Anzeige eigener Artikel

der Implementation wurde mit bis zu zwanzig PC Member Logins und über neunzig eingereichten Artikeln durchgeführt, um die Geschwindigkeit des Seitenaufbaus zu überprüfen. In jedem PC Member Login wurden Biddings durchgeführt und anschließend Delegationen über den PC Chair Account vorgenommen. Diese Delegationen wurden dann teilweise akzeptiert, zurückgewiesen und abgeschlossen. Die sich daraus ergebenden Delegationsstati und die jeweiligen Hintergrundfarben in der Tabelle wurden anschließend verglichen, um die einwandfreie Zuordnung der Farben zum derzeit vorliegenden Status sicherzustellen.

Damit eine Zurückweisung einer Delegation durch den PC Member über das Entfernen der Delegation hinaus in der Bidding Matrix kenntlich zu machen ist, muss diese Information außerhalb des Delegation Feature hinterlegt werden. Da es im wesentlichen darum geht, diese Information in der Bidding Matrix anzuzeigen, eignet sich das Bidding zur entsprechenden Artikel-/Benutzerkombination besonders dazu, diese Information vorzuhalten. Nach dem Entfernen der Delegation wird das entsprechende Bidding für die betroffene Artikel-/Benutzerkombination geladen oder neu erzeugt, das Feld "bidderbid" auf "R" gesetzt und anschließend wieder in der Datenbank abgelegt. Dieser Bidding Status wird dann in der Bidding Matrix automatisch wie ein reguläres Bidding angezeigt und bleibt über das Entfernen der Delegation hinaus erhalten, sofern der PC Member kein neues Bidding abgibt. Trotz einer zurückgewiesenen Delegation hat der PC Chair weiterhin die Möglichkeit, diese Delegation erneut aus der Bidding Matrix heraus an den selben PC Member zu vergeben (Abbildung 3.68). Getestet wurde diese Funktion durch Zurückweisung einer Delegation durch den PC Member und anschließende Kontrolle des entsprechenden Feldes in der Datenbank.

The Bidding and Assignment Matrix

This matrix gives an overview of all biddings the PC Members have done. Additionally it shows the planned (blue cells) and according to the delegation matrix the different color codes of a delegated article. The matrix contains for each PC Member one column, which again contains one cell per article.

Once a PC Member has done her/his bidding one of the following characters will be shown:

- B ("bid") indicates a strong interest in reviewing the corresponding articles.
- I ("indifferent") marks articles the PC Member doesn't bid for, but she/he would review if assigned.
- U ("unable") marks articles the PC Member doesn't feel sufficiently competent to review.
- R ("rejected") marks articles the PC Member has rejected to review.
- C ("conflict") indicates articles the PC Member are in conflict with, e.g. if she/he is author or co-author herself/himself, or if authors of that article are quite "close" to the PC Member.
- c ("conflict") indicates articles the PC Chair has set to conflict in the conflict matrix.

A cell additionally contains a checkbox if an article is not already delegated to a PC Member and if a PC Member has not set a conflict (C). With this checkbox you can specify an assignment. By default only checkboxes are checked where the PC Member has bidden (B) for an article.

Cells with planned assignments are highlighted blue.

Once an assignment is done and the delegation is waiting for decision the corresponding cell is highlighted yellow.

When a delegation is accepted by a PC Member the cell is highlighted green.

When a delegation is rejected by a PC Member the cell is highlighted red.

Note: The conflicts (C) can automatically be stored to the conflict matrix by pushing 'Store Conflicts'.

Note: If you want to remove a done delegation you must use the "Delegation" feature.

ID	DP	TT	TA	dM	wt	ab	si	At	tc	Ht	kl	ng	FT	Mt	ht	eF	kG	jt	FB	zz	ID	Sum:()
001	c C	I	c I	R	c C	c C	B	B	B	B	I	I	I	B	B	R	B	B	I	B	001	
002	B	I	c C	B	c C	U	R	B	I	B	I	I	I	B	B	B	I	B	c C	B	002	
003	I	I	c C	I	c C	c C	B	B	U	B	I	B	B	B	B	c B	B	B	c C	B	003	
004	I	B	c C	R	c C	B	B	B	U	B	I	I	B	B	B	I	B	B	U	B	004	
005	c C	B	B	U	B	I	B	c C	c C	c C	I	B	c C	B	B	c C	R	B	I	B	005	
006	I	B	c C	R	B	I	B	c C	c C	c C	I	I	c C	B	B	c C	I	B	c C	B	006	
007	I	I	c C	B	R	U	B	c C	B	c C	R	I	I	B	B	I	U	B	B	B	007	
008	B	I	c C	I	B	U	R	c C	B	c C	I	B	B	c C	B	U	c C	B	U	c C	008	
009	B	U	c C	U	R	I	R	B	I	c C	R	I	B	c C	B	B	c C	B	B	B	009	
010	U	U	I	I	B	R	B	B	I	c C	B	c C	U	c C	B	B	B	B	R	c C	010	

- **CheckSum** computes the sum of Reviewers per article and the sum of articles per Reviewer. A green number means that you can do more delegations for the corresponding article. A black number indicates that the current number of distributions is equals to the desired value. The number will be highlighted red if there are too many delegations for a PC Member.
- **Save** writes the matrix entries into the database without doing any delegations. So that you can finish your planned assignments later.
- **Save & Delegate** stores the matrix entries and distributes the review tasks to the PC Members.
- **Store Conflicts** stores the conflicts manually set by the PC Members in their biddings to the conflict matrix.

Legend

Initials	First Name	Last Name	Email Address
DP	Daniel	Pierlings	pierlings@metaframe.de
TT	Testvorname	Testnachname	testauthor@metaframe.de

Abbildung 3.67: Bidding Matrix mit Farbcodes aus der Delegation

ID	DP	TT	TA	ID	Sum:()
001	C		C	001	
002	B	c	C	002	
003	I		C	003	
004	I	R	C	004	
005	C		B	005	
Sum:					

Abbildung 3.68: Anzeige eines zurückgewiesenen Artikels in der Bidding Matrix

Um zu verhindern, dass bei paralleler Benutzung der Bidding Liste und Bidding Matrix durch einen PC Member und einen PC Chair beim Speichern der Änderungen durch den jeweiligen Benutzer Informationen verloren gehen, muss dafür gesorgt werden, dass nur die für den PC Member oder PC Chair relevanten Tabellenspalten geschrieben werden. Die Bidding Liste und die Bidding Matrix arbeiten jeweils auf unterschiedlichen Feldern, daher ist eine Trennung der jeweils bearbeiteten Datenfelder möglich. Es werden vor dem Speichern alle Biddings aus der Datenbank geladen. Danach wird über alle Artikel iteriert und jeweils überprüft, ob ein Bidding zum Artikel existiert. Wenn ein Bidding vorhanden ist, wird es aus der Tabelle geladen. Alle Werte, die in der Bidding Liste oder Bidding Matrix jeweils nicht bearbeitet werden, müssen anschließend aus dem Bidding Objekt aus der Datenbank gelesen und in das entsprechende Bidding Objekt aus dem Call Context kopiert werden. Danach wird das Bidding aus dem Call Context in die Datenbank geschrieben und dabei das vorher schon vorhandene Bidding ersetzt. Das Activity Chart 3.69 zeigt die Vorgehensweise zum Abgleich der Datenfelder. Um die neue Funktionalität zu testen, wurde der ursprüngliche Zustand der Datenbanktabelle ausgelesen, dann von zwei Rechnern aus jeweils mit PC Member Login die Bidding Liste bzw. mit PC Chair Login die Bidding Matrix gleichzeitig aufgerufen. Speicherte der PC Member seine Änderungen vor dem PC Chair, wurden dessen Eintragungen beibehalten und nicht überschrieben, ebenso im umgekehrten Fall. Die Eintragungen in der Datenbank wurden dabei über die Linux Konsole verfolgt.

Für die Korrektur des Datums in der Emailbenachrichtigung über eine erteilte Delegation ist das SIB "PrepareBiddingMatrixMailParameter" zu modifizieren. Änderungen am Graphen sind nicht notwendig. Der PC Chair gibt bei Delegationen nur den Stichtag an, bis zu dem die zugewiesene Aufgabe erledigt sein soll. Da diese Zeitraumangabe den genannten Tag mit einschließt, muss die Uhrzeitangabe in der versandten Email auf "23:59:59" gesetzt werden. Das aus dem Request geladene Datum wird in ein Objekt des Typs Date gewandelt, dieses wird von Java bei Übergabe des Datums als String automatisch im Format "01.01.2004 00:00:00" initialisiert. Die Methoden zur Manipulation der Uhrzeit im Date Objekt werden allerdings von Java nicht mehr offiziell unterstützt, daher wird ein Gregorian Calendar erzeugt und mit dem aus dem Template gelesenen Datum initialisiert. Anschließend werden mit den vom Gregorian Calendar

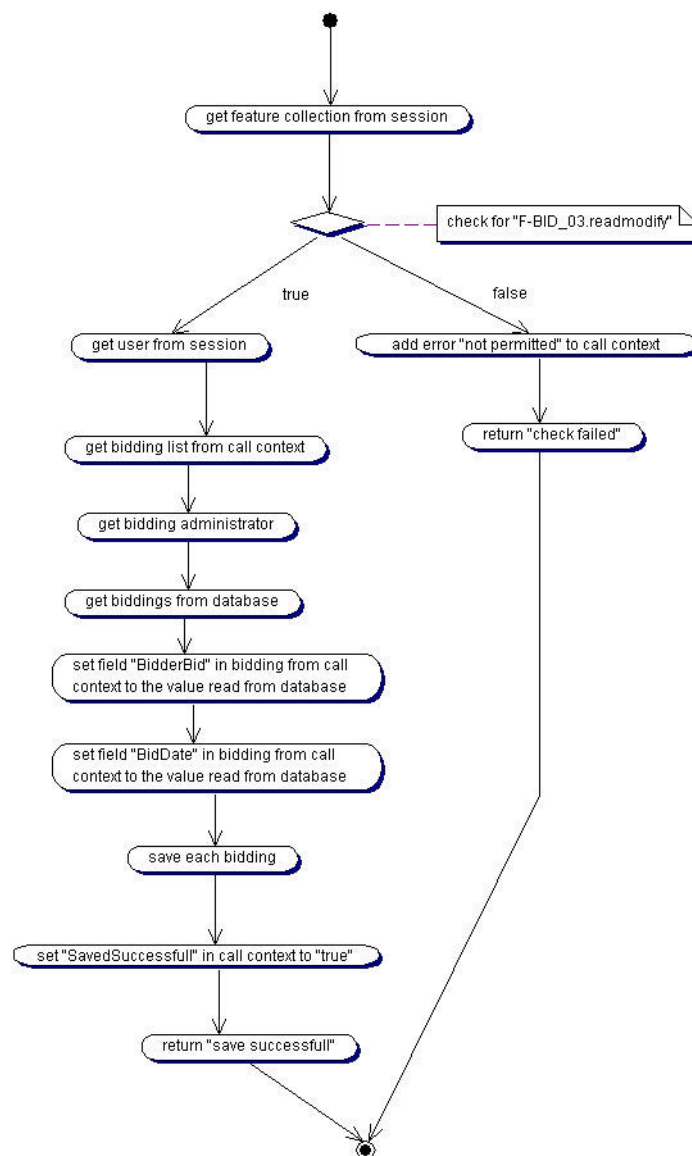


Abbildung 3.69: Speichern von Biddings, ohne Datenfelder des PC Chair zu überschreiben

bereitgestellten Methoden die Stunden, Minuten und Sekunden gesetzt und dann aus dem Calendar wieder ein Date Objekt erzeugt. Dieses korrigierte Datum wird dann in die Map für das Emailtemplate kopiert. Der Test erfolgte durch Delegation einiger Reviews und anschließender Kontrolle der versandten Emails.

Implementierungsphase

Die Implementation der einzelnen Teilaufgaben erfolgte weitestgehend wie in der Modellierungsphase vorgesehen. Die einzige Konzeptänderung betraf die Zurückweisung einer Delegation. Die Information sollte ursprünglich in einer neuen Spalte der Bidding Tabelle hinterlegt werden. Da diese Information allerdings nur bis zu einem erneuten Bidding benötigt wird, konnte die Tabellenspalte für das Bidding verwandt werden. Dadurch wurden auch weniger Änderungen am Code für die Generierung der Bidding Matrix nötig.

Alle sechs Teilaufgaben wurden gleichzeitig in drei separaten Graphen bearbeitet, jedes Gruppenmitglied bearbeitete zwei Aufgaben. Überschneidungen bei der gleichzeitigen Implementation traten kaum auf, auch die abschließende Integration aller Änderungen in einen gemeinsamen Graphen verlief problemlos.

3.3.3 File download workflow

Beschreibung der Aufgabe

Bisher stand dem Benutzer nur das Downloaden von einzelnen Artikeln zur Verfügung. Da eine Konferenz eine hohe Anzahl von Beiträgen hat, muss der Benutzer unnötig Seitenwechsel und Downloadvorgänge durchführen, was sehr zeitintensiv ist.

Das Download Feature bietet dem Benutzer zwei Möglichkeiten des Downloads von Artikeln. Entweder werden dem Benutzer alle Artikel angezeigt, die er sehen darf, oder alle Artikel, die ihm delegiert wurden. Nach dieser Entscheidung dürfen Artikel aus der angezeigten Liste ausgewählt werden. Es besteht die Möglichkeit alle Artikel auf einmal, alle Artikeldateien eines Artikels, oder einzelne Artikeldateien auszuwählen. Die nächste Entscheidung, die getroffen werden muss, ist die Art der Zip-Datei, die heruntergeladen wird. Die beiden Möglichkeiten sind die "plain"-Methode und die "multi"-Methode. Die "plain"-Methode komprimiert alle Artikeldateien eines Artikels in ein Verzeichnis. Die "multi"-Methode komprimiert alle Artikel in je eine extra Zip-Datei. Diese Zip-Dateien werden dann in einer Zip-Datei zusammengefasst.

Modellierungsphase

Um das Herunterladen von Artikeln zu vereinfachen, wird folgender Workflow durchlaufen: Der Benutzer wählt zwischen den beiden Möglichkeiten, alle erlaubten oder alle delegierten Artikel herunterzuladen. Abhängig davon werden die Artikel angezeigt.

Durch das Markieren der Checkboxen werden vom Benutzer Artikeldateien ausgewählt. Das System stellt dann die ausgewählten Daten zusammen und zeigt dem Benutzer eine Übersicht. Dieser kann den Download jetzt starten, ihn abbrechen, oder seine Auswahl ändern. Wird der Download begonnen, so wird ein Vektor erzeugt, in dem alle gewählten Artikeldateien zu finden sind. Aus diesem Vektor wird anschliessend die Zip-Datei erstellt. Abhängig von der gewählten Zip-Methode wird entweder ein Verzeichnis oder eine Zip-Datei erstellt, in der sich alle zugehörigen Artikeldateien befinden. Mit Hilfe des Vektors wird aus dem CVS die aktuelle Datei gelesen und dem Verzeichnis oder der Zip-Datei hinzugefügt. Diese Schritte werden wiederholt, bis alle Dateien zusammengefasst worden sind. Anschließend wird die große Zip-Datei erzeugt und dem Benutzer zum Download bereitgestellt.

Das Aktivitätsdiagramm (siehe Abbildung 3.70) beschreibt, wie der Benutzer die ausgewählten Artikeldateien als eine Zip-Datei herunterladen kann. Zuerst kann der Benutzer auf der Seite "ShowArticleList" Dateien aus der Liste aller Artikel oder aus der Liste der ihm delegierten Artikel zu wählen. Dann werden die entsprechenden Listen in einer neuen Seite angezeigt, in der der Benutzer noch einmal die gewünschten Artikeldateien und die Zip-Methode wählen kann. Mit der Methode "multi_zip" wird eine große Zip-Datei erzeugt, die jeweils eine kleinere Zip-Datei für jeden Artikel enthält. Bei der Auswahl der Methode "plain_zip" erhält der Benutzer eine Zip-Datei, die ein Unterverzeichnis für jeden Artikel beinhaltet.

Implementierungsphase

Zunächst muss das Makro "Article" (siehe Abbildungen 3.71 und 3.72) modifiziert werden, das den Artikel-Teilbaum enthält, der durch die Kante "article" aus dem "Main Service-Entry" SIB erreicht werden kann. Um das Herunterladen der Artikel zu ermöglichen, musste eine wml-Seite verändert und zwei neu geschrieben werden.

HTML „articlelist-cont.wml“ : Zuerst wurde die bestehende "articlelist-cont.wml" Datei modifiziert. Dort wurden lediglich zwei neue Links für den Download aller oder aller delegierten Artikel hinzugefügt. In der ersten Version wurde dies mit einer Auswahlliste realisiert, durch einfache Links spart man jedoch Klicks, wodurch die zweite Lösung effizienter war.

Die Seite, die alle herunterladbaren Artikel anzeigt, sollte die Artikeldateien in die richtigen Spalten einsortieren. Die Spalten sind von den Einstellungen in der "setup.properties" Datei abhängig. In dieser Datei können viele Einstellungen des OCS vorgenommen werden. Unter anderem wird festgelegt, welche Dateiformate hochgeladen werden dürfen. Diese Dateiformate stellen die Spalten der Tabelle dar, in der sich die herunterladbaren Artikel befinden.

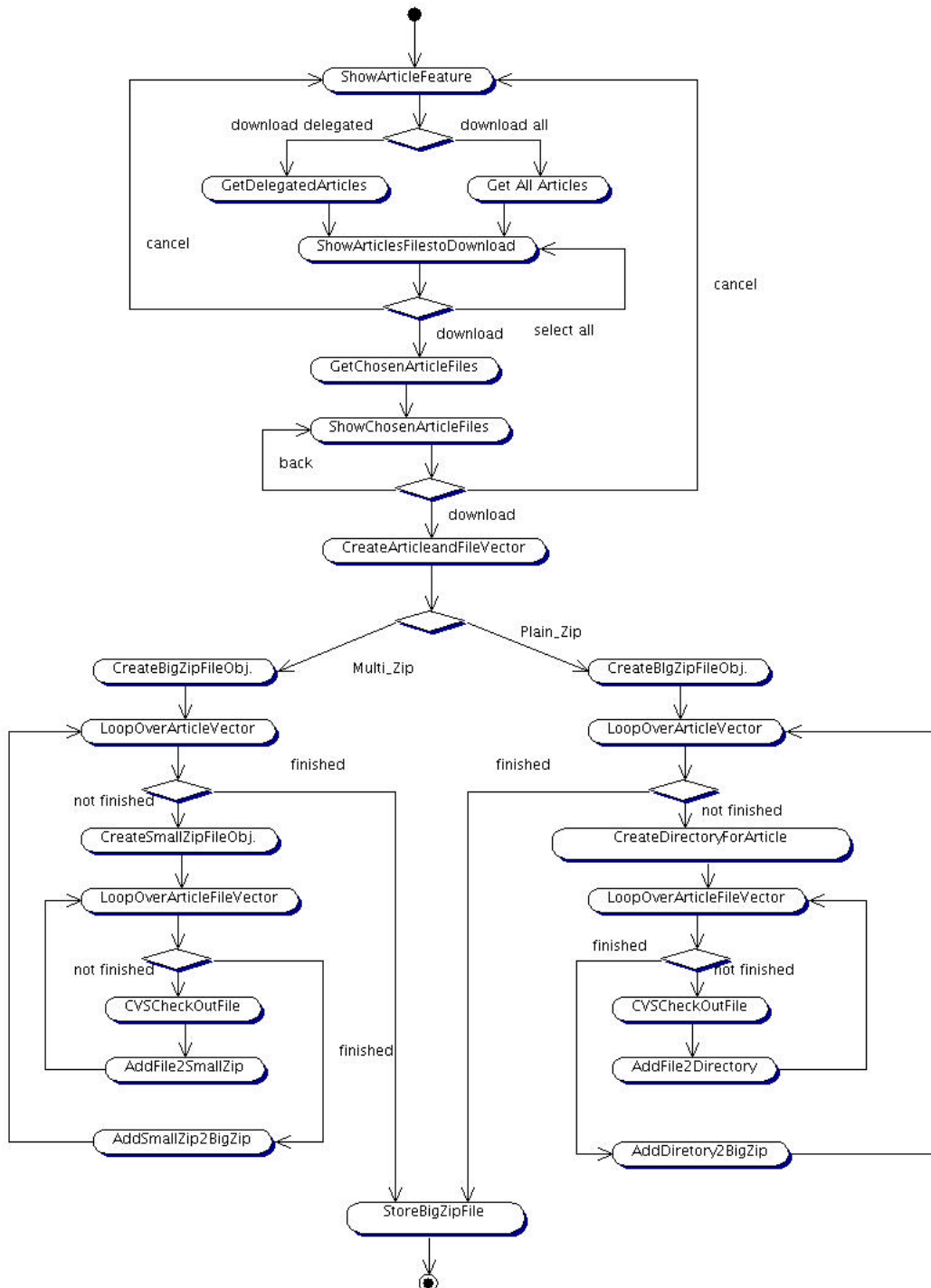


Abbildung 3.70: Activity Chart "Download Article Files as Zipfile"

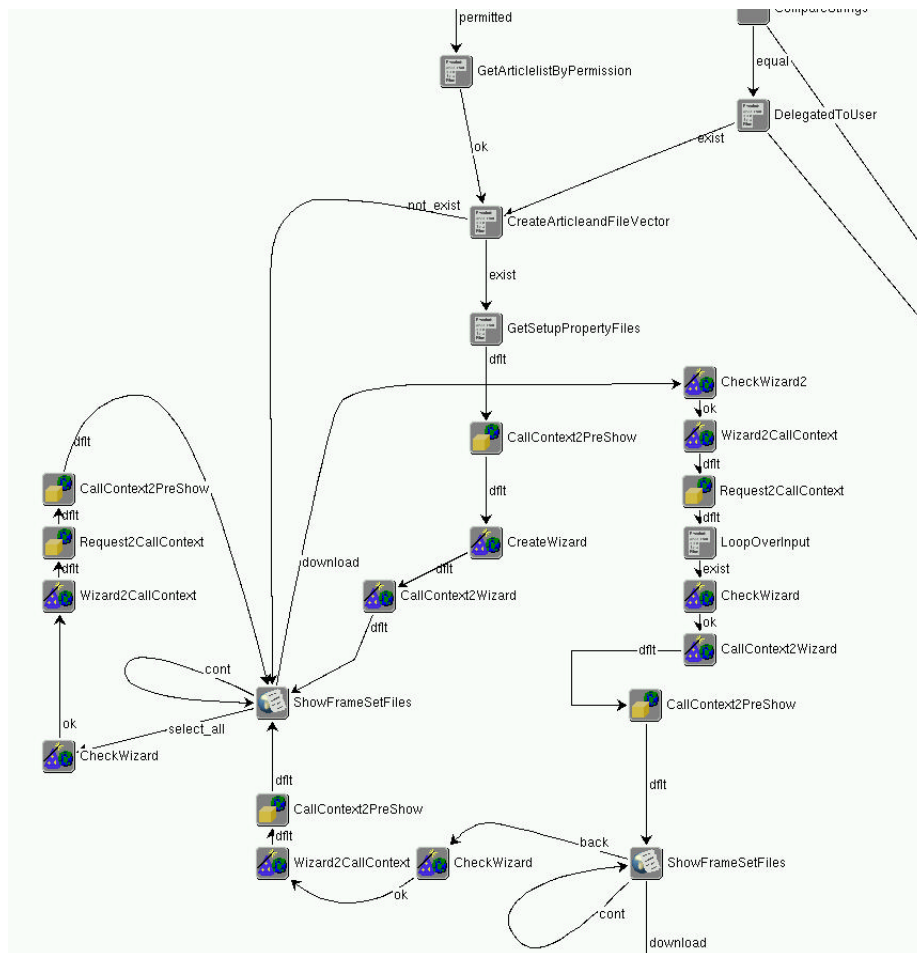


Abbildung 3.71: Makro "Article Macro 1"

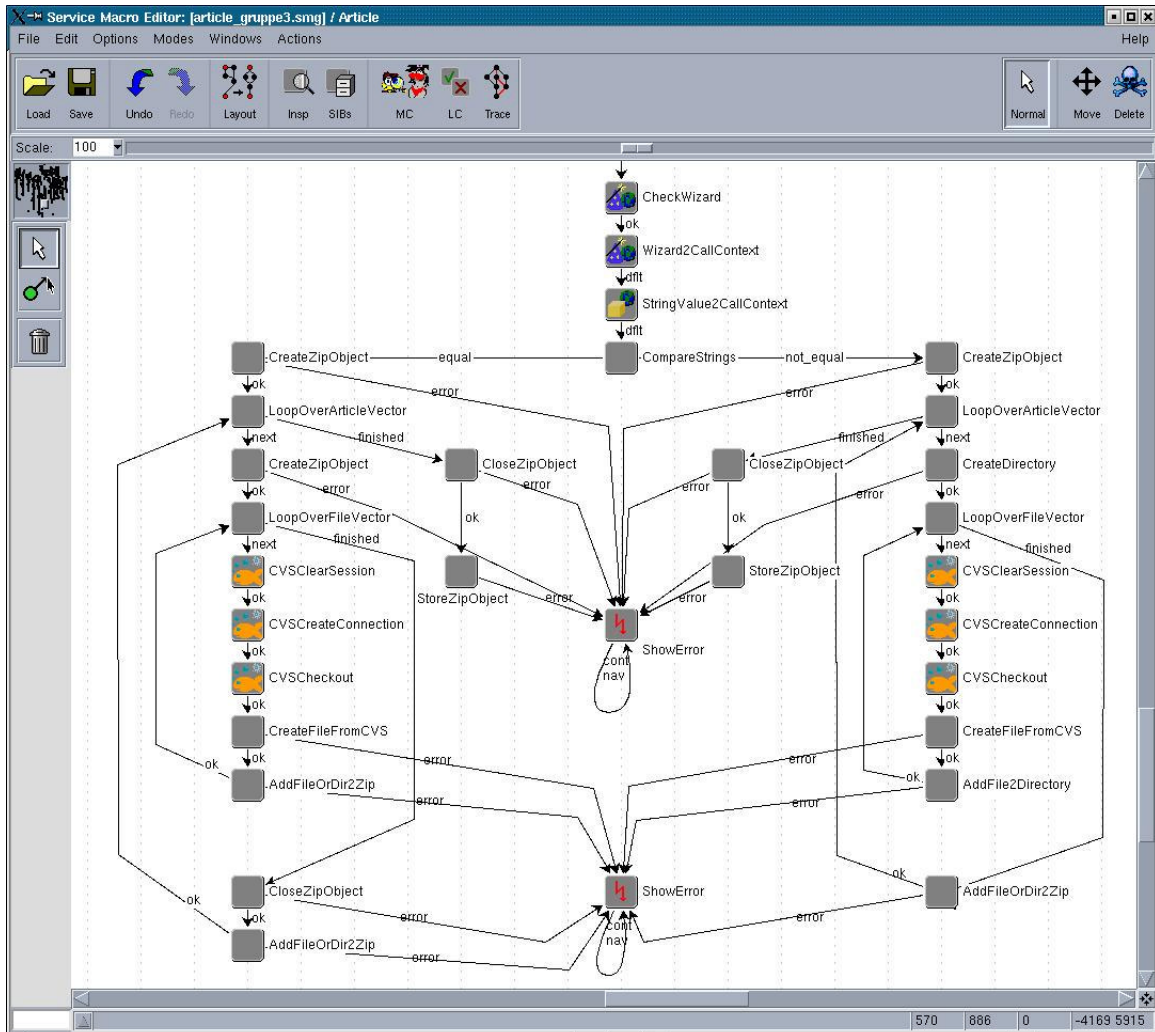


Abbildung 3.72: Makro "Article Macro 2"

SIB „GetSetupPropertyFiles“ : Das Lesen der benötigten Einstellungen wurde vom *„GetSetupPropertyFiles“* SIB vollzogen. Mit Hilfe bereits vorhandener SIBs wurde die benötigte Artikelliste erzeugt und in den Call Context gespeichert.

SIB „DelegatedToUser“ : Um an die delegierten Artikel zu gelangen, wurde das SIB *„DelegatedToUser“* geschrieben. Es werden alle an einen Benutzer delegierten Artikel ausgelesen.

HTML „downloadarticlelist-cont.wml“ : Aus all diesen Daten erzeugt die neugeschriebene wml-Datei *„downloadarticlelist-cont.wml“* eine Tabelle. In der ersten Spalte der Tabelle befinden sich die Titel der Artikel samt der Artikel ID. Die folgenden Spalten hängen von den oben erläuterten Einstellungen ab. Die letzte Spalte fasst alle Artikel zusammen, deren Endung nicht mit den vorgegebenen Formaten übereinstimmt. Um die Auswahl der Artikel zu vereinfachen, existieren drei Buttons. Mit ihrer Hilfe kann der Vorgang gestartet oder abgebrochen werden. Der dritte Knopf ermöglicht das Auswählen aller Artikel. Außerdem wird die Zip-Methode durch das Wählen der zugehörigen Radiobuttons vor-eingestellt.

SIB „LoopOverInput“ : Um die gewählten Checkboxes auszulesen, wurde das SIB *„LoopOverInput“* erstellt. Dieser SIB durchläuft zuerst die Artikel, zu denen der Benutzer alle Dateien herunterladen möchte. Es werden alle zugehörigen Artikeldateinamen unter dem Schlüssel, der mit der zugehörigen Artikel ID gehört, gespeichert. Eine ähnliche Prozedur wird anschliessend für die einzelnen gewählten Artikeldateien durchlaufen, die jedoch viel spezieller ist, da viele Ausnahmefälle auftreten können.

SIB „CreateArticleandFileVector“ : Anschliessend erzeugt der *„CreateArticleandFileVector“* SIB einen Artikelvektor, der alle gewählten Artikel IDs enthält. Außerdem wird eine Map erzeugt. In dieser Map befinden sich Dateivektoren, die alle zu einem Artikel zugehörigen Artikeldateinamen enthalten. Die Map und der Vektor werden unter einem bestimmten Schlüssel in den Call Context geschrieben.

HTML „confirmdownloadarticles.wml“ : Die wml-Seite *„confirmdownloadarticles.wml“* liest die gewählten Artikeldateien aus den Vektoren und der Map und zeigt sie dem Benutzer zur Bestätigung an. Durch den Start-Button wird der Download gestartet, durch den Cancel-Button abgebrochen und durch den Back-Button kann die Auswahl nochmal verändert werden.

SIB „CreateZipObject“ : Dieser SIB wird zur Erzeugung von ZipOutputStreams verwendet. Zuerst wird ein ZipObjectName aus dem Call Context unter dem ZipObjectName_key gelesen. Dann wird ein ZipOutputStream mit diesem Namen in einem temporären

Verzeichnis erzeugt, das vorher als Parameter "DefaultZipFileDirectory" mit dem Wert z.B. "/ziptmp" in "/TOMCAT-HOME/webapps/etapsdemo /WEB-INF/web.xml" und in "/eureka/ocs/config /web.xml.dist" schon definiert wurde. Anschließend wird dieser ZipOutputStream in den Call Context unter dem ZipObject_Key geschrieben.

SIB „LoopOverArticleVector“ : Der Zip SIB benötigt drei Parameter: ArticleVectorCC_Key, FileMapCC_Key und FileVectorCC_Key. Mit dem ArticleVectorCC_Key wird ein Artikelvektor aus dem Call Context gelesen. Eine Map, die auch aus dem Call Context unter dem Schlüssel FileMapCC_Key herausgelesen wird, enthält einige Dateivektoren als Elemente. Bei jedem Durchlauf liefert diese Map dem Programm den Dateivektor, der unter der entsprechenden Artikel ID abgelegt ist. Danach wird unter dem in FileVectorCC_Key abgelegten Wert dieser Vektor in den Call Context geschrieben. Zusätzlich werden einige Informationen über das Zip-Objekt und den Speicherpfad für spätere Verwendung in den Call Context geschrieben.

SIB „LoopOverFileVector“ : Dieser SIB braucht einen Parameter FileVectorCC_Key. Dadurch wird ein Dateivektor aus dem Call Context geholt. Bei jeder Schleife erhält man einen Dateinamen aus diesem Dateivektor. Außerdem werden die CVS Verzeichnis- und Dateiinformationen in den Call Context geschrieben, damit der Dateiinhalte später aus dem CVS-Repository wieder herausgelesen werden kann.

SIB „CreateFileFromCVS“ : Durch diesen SIB wird eine Datei aus dem CVS ausgelesen und unter dem Parameter File_Key in den Call Context geschrieben. Die CVSFileinfo wird aus dem Call Context geholt, um einen FileOutputStream zu erzeugen, der mit dem richtigen Dateiinhalte aus dem CVS Repository gefüllt wird.

SIB „AddFileOrDir2Zip“ : Dieser SIB hat zwei Parameter File_Key und ZipObject_Key. Durch den ZipObject_Key wird ein ZipOutputStream aus dem Call Context gelesen. Durch den File_Key wird eine Datei oder ein Verzeichnis aus dem Call Context geholt, welche oder welches in einen ZipOutputStream gepackt werden soll. Nach der Komprimierung wird der ZipOutputStream wieder unter ZipObject_Key in den Call Context geschrieben.

SIB „CloseZipObject“ : Dieser SIB besitzt drei Parameter: CloseZipFile_Key, ZipObjectName_Key und ZipObject_Key. Zunächst wird ein ZipOutputStream durch ZipObject_Key aus dem Call Context geholt und geschlossen. Bis hier ist die Komprimierung bereits fertig, also eine Zip-Datei schon vorhanden. Durch den ZipObjectName_Key wird der Zip-Dateiname aus dem Call Context gelesen, unter dem diese abgeschlossene Zip-Datei im Call Context zu finden ist. Dann wird diese Zip-Datei unter dem CloseZipFile_Key in den Call Context geschrieben, damit man nachher wieder darauf zugreifen und

sie weiter verarbeiten kann, wie z.B diese Zip-Datei später in einen anderen ZipOutput-Stream packen oder unter einem vom Benutzer des Dienstes gewünschten Namen auf der Festplatte abzuspeichern.

SIB „StoreZipObject“ : Für diesen SIB ist ein Parameter ZipFile_Key schon ausreichend. Durch den ZipFile_Key kann eine Zip-Datei im Call Context gefunden und unter einem vom Benutzer des Dienstes gewünschten Namen auf der eigenen Festplatte abgespeichert werden. Außerdem wird das temporäre Verzeichnis, welches für die Komprimierung benutzt wurde, vollständig entfernt.

SIB „CreateDirectory“ : Dieser SIB ist nur für „plain_zip“ entstanden. Er besitzt zwei Parameter DirectoryName_Key und Directory_Key. Durch DirectoryName_Key wird ein Verzeichnis mit einem bestimmten Namen erzeugt und anschließend unter Directory_Key in den Call Context geschrieben.

SIB „AddFile2Directory“ : In diesem SIB werden zwei Parameter Directory_Key und File_Key eingesetzt. Durch Directory_Key wird ein Verzeichnis aus dem Call Context geholt, in das eine Datei, die durch File_Key im Call Context zu finden ist, kopiert wird. Am Schluss wird dieses Verzeichnis wieder unter Directory_Key in den Call Context geschrieben.

Testphase

Nach allen Tests funktionierte das neue Feature einwandfrei. Auf den Webseiten wurden alle Möglichkeiten einen Button zu klicken durchprobiert. Alle Schaltflächen scheinen ihre Aufgabe zu erledigen. Gab es keine Artikel zu downloaden, wurden auch immer die korrekten Ausnahmeseiten angezeigt. So wurde ausgeschlossen, dass eine leere Zip-Datei erzeugt wird. Natürlich wurde auch getestet, ob die gewählten Artikel auch wirklich in der Zip-Datei enthalten sind. Hierbei traten keine Fehler auf. Als Fazit wurde festgestellt, dass alle implementierten Dateien funktionieren.

3.3.4 Email Template Setup Workflow

Beschreibung der Aufgabe

Der OCS benutzt vordefinierte Email-Templates, welche automatisch vom Dienst bei gegebenem Anlass verschickt werden. Beispiele für solche Emails sind die Registrierung eines neuen Benutzers, die Meldung eines neuen Eintrags in der Aufgabenliste eines Benutzers oder eine Einreichung eines Artikels.

Der PC Chair hatte keine Möglichkeit im laufenden Betrieb des Dienstes die Texte der Templates zu verändern. Dies sollte durch die Bearbeitung des Gruppenthemas geändert

werden. Aufgabe war es, dem PC Chair das Ändern dieser Templates zu ermöglichen und verschiedene Versionen zu speichern. Da innerhalb der Templates Velocity-Variablen verwendet werden können (etwa der Name des Autors oder die Nummer des Artikels, auf den sich die Email bezieht), sollte sichergestellt werden, dass nur syntaktisch korrekte Templates gespeichert werden können.

Modellierungsphase

Dem Setup sollte ein neuer Unterpunkt hinzugefügt werden, unter dem die Email-

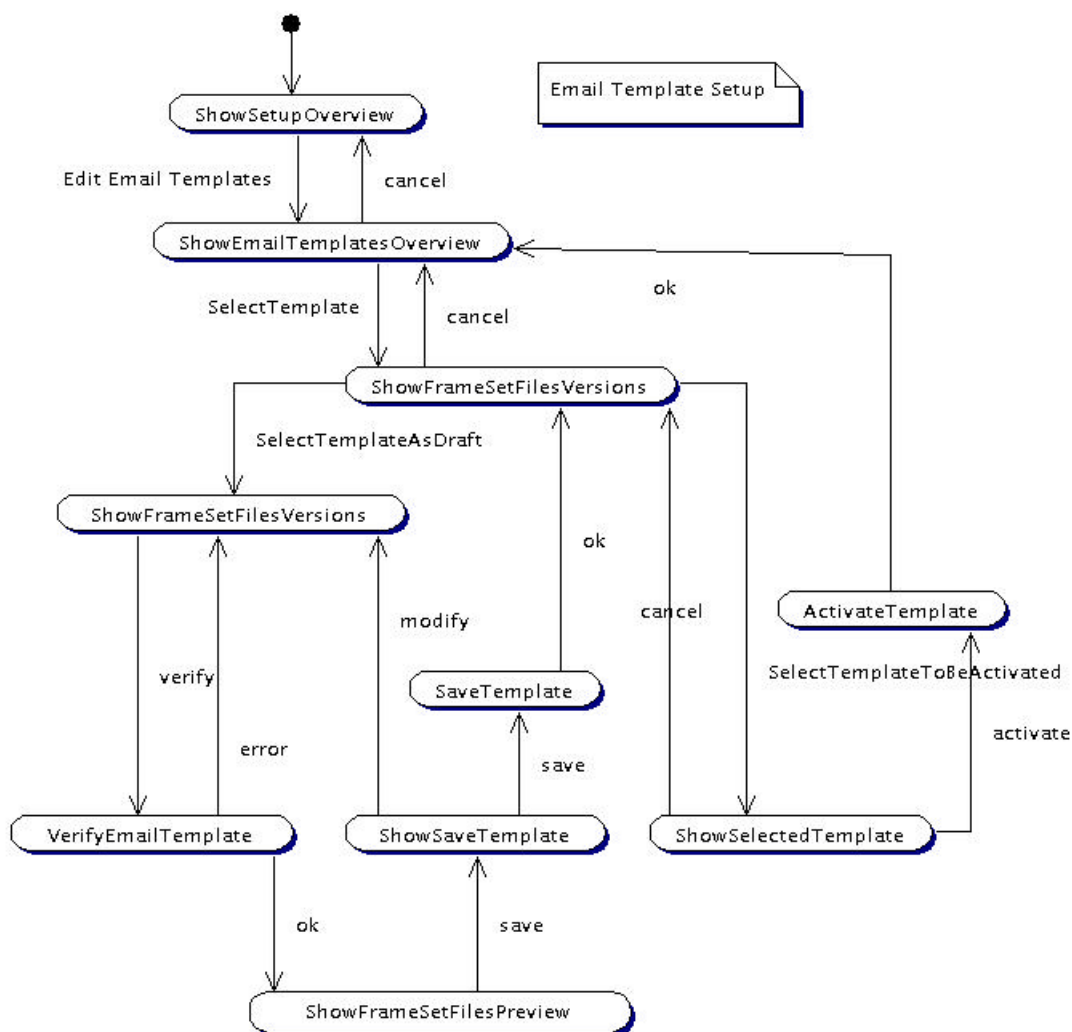


Abbildung 3.73: Editieren der Email Templates

Templates verändert werden können (vgl. Abb. 3.73). Zunächst wird eine Übersicht aller verfügbarer Templates und ihre Beschreibung angezeigt. Hier kann der PC Chair ein

Template selektieren, das er anpassen möchte. Daraufhin gelangt er zu einer Übersicht aller vorhandener Versionen des Templates. Das aktuell vom Dienst verwendete Template ist in fetter Schrift hervorgehoben.

Der PC Chair wählt nun eine Version, die er als Vorlage für die neue Version benutzen will und klickt "select as draft". Die folgende Seite zeigt den Titel, die Beschreibung und den Text des Email-Templates. Der PC Chair kann nun den Text und die Beschreibung für die neue Version editieren. Unterhalb des Eingabefeldes für den Text werden Eingabefelder für vom Template verwendete Variablen aufgelistet, in die Werte eingetragen werden können, die für eine spätere Vorschau des neuen Templates benutzt werden sollen. Nachdem der Eingabevorgang abgeschlossen ist, kann über "verify" der Verifikationsprozess für die Syntax gestartet werden. Wenn Syntaxfehler gefunden wurden, wird die Eingabeseite mit entsprechenden Fehlermeldungen erneut angezeigt.

Ist die Verifikation fehlerfrei verlaufen, wird dem PC Chair das neue Template und das Template mit den Werten, die für die Variablen angegeben wurden, angezeigt. Nun kann er entscheiden, ob er zurück zur Bearbeitungsseite möchte, oder ob das Template gespeichert werden soll. Im letzteren Fall wird dem Template automatisch eine Versionsnummer zugewiesen, damit keine bestehenden Versionen überschrieben werden. Danach gelangt der PC Chair zurück zur Übersichtsseite der Templateversionen, auf der auch die neue Templateversion aufgelistet ist. Um das neue Template zu aktivieren, wählt er "activate".

Implementierungsphase

Der Workflow zum Verändern der Email-Templates wurde in das Setup-Makro eingefügt. Der SIB ShowSetupOverview erhält eine neue Verzweigung, die auf den Email-Template Setup Workflow verweist. Dies wurde auch in der Datei setup-overview-cont.wml durch einen weiteren Link umgesetzt.

Die Templates, die vom Dienst benutzt werden können, befinden sich in dem Verzeichnis webapps/etapsdemo/WEB-INF/mail/service. Zu jedem Template existiert eine Standardversion, die nach dem Schema `templatename_dflt.txt` benannt ist. Die verschiedenen Versionen sind mit `templatename_version.txt` benannt. Der Name der jeweils aktuellen Version wird in der Datei `templatename.active` gespeichert und der Text des aktuell benutzten Templates in der Datei `templatename.txt`. Die zugehörige Beschreibung ist jeweils in einer Datei namens `templatename_version.desc.txt`, bzw. `templatename_dflt.desc.txt` abgelegt.

Um den weiteren Workflow umzusetzen, wurden einige neue SIBs geschrieben. Diese benutzen oft Parameter, bei denen es sich um Platzhalter im Call Context handelt, unter denen Variablen abgelegt werden. Folgende neue SIBs wurden geschrieben:

- `getEmailTemplatesOverview` liest die vorhandenen Templates aus dem Verzeichnis, das als Properties Schlüssel in dem Parameter `template_dir` angegeben ist (z.B. `Mail.MailDir`). Es werden alle Templates, sowie deren zugehörige Beschreibungen geladen, die auf `dflt.txt` enden. Unter dem Schlüssel in dem Parameter `descriptions` werden die Beschreibungen als Vektor in den Call Context geschrie-

ben. Analog enthält der Parameter `templates` den Call Context Schlüssel, unter dem die Namen der Templates als Vektor abgelegt werden.

- `GetEmailTemplateVersions` liest alle Versionen eines ausgewählten Templates und die zugehörigen Beschreibungen ein. Die Parameter sind identisch mit denen des `GetEmailTemplatesOverview` SIBs.
- `GetEmailTemplate` liest den Template Text sowie die Beschreibung eines Email Templates ein. Das Verzeichnis der Templates wird als Properties Schlüssel in dem Parameter `template_dir` angegeben. Der Name des zu ladenden Templates kann über den Parameter `template` spezifiziert werden. Der SIB legt daraufhin den Template Text unter dem Schlüssel, der in `template_body` angegeben ist, ab. Dazu wird die Beschreibung unter dem Schlüssel, der in `description` angegeben ist, in den Call Context abgelegt. Darüberhinaus filtert der SIB alle verwendeten Velocity Variablen aus dem Template und stellt sie im Call Context unter dem Schlüssel, der in dem Parameter `vars` angegeben ist, zur Verfügung.
- `VerifyEmailTemplate` überprüft ein Template auf syntaktische Korrektheit. Der Parameter `description` dient nur dem Weitertragen der Beschreibung, ebenso wie der Parameter `template_name` den Namen des Templates für spätere Verwendung sichert. Der Text des Templates wird über den Schlüssel, der im Parameter `template_body` angegeben ist, aus dem Call Context gelesen. Der SIB überprüft daraufhin, ob nur Variablen, die in `vars` vorhanden sind, benutzt wurden und diese syntaktisch korrekt benutzt wurden. Im Falle eines Fehlers, wird eine Fehlermeldung in den Call Context geschrieben. Der Parameter `values` dient dazu, die Werte für die Variablen, die für die Vorschau eingegeben wurden, an nachfolgende SIBs weiter zu reichen.
- `SaveEmailTemplate` speichert eine neue Version eines Templates. Dazu werden die Parameter `template_name`, `template_body`, `template_dir` und `description` wie oben beschrieben benutzt. Allerdings handelt es sich hier bei `template_name` noch nicht um den neuen Namen, sondern um den Namen des Templates, das als Vorlage gedient hat. Der SIB ermittelt selbständig den zu verwendenden Versions-Suffix und speichert das Template sowie die Beschreibung dementsprechend. Ist die maximale Versionsnummer (hier 999) erreicht, wird eine Fehlermeldung ausgegeben.

Um die von den oben beschriebenen SIBs generierten Informationen anzuzeigen, wurden dazu noch einige WML-Dateien geschrieben, auf die im Folgenden eingegangen werden soll.

- `emailtemplates-cont.wml` zeigt eine Tabelle mit allen Templates und ihren Beschreibungen, die automatisch vom Dienst versendet werden. Über einen Radiobutton kann jeweils ein Template ausgewählt werden, dessen vorhandene Versionen

sich der PC Chair anzeigen lassen will. Die entsprechenden Daten, die angezeigt werden, werden von `GetEmailTemplatesOverview` generiert.

- `emailtemplateversions-cont.wml` zeigt eine Tabelle aller vorhandenen Versionen und ihrer Beschreibungen eines ausgewählten Templates an. Die aktuell vom Dienst benutzte Version ist in fetter Schrift hervorgehoben. Über einen Radiobutton kann eine Version als Vorlage für eine neue Version gewählt werden. Die angezeigten Daten werden von `GetEmailTemplateVersions` generiert.
- `emailtemplate-cont.wml` zeigt ein ausgewähltes Template und die zugehörige Beschreibung in Textfeldern an. Der PC Chair kann auf dieser Seite die Beschreibung und den Template Text verändern, um eine neue Version des Templates zu erzeugen. `GetEmailTemplateVersions` generiert die hier angezeigten Daten. Desweiteren werden Eingabefelder für die im zugehörigen Standardtemplate verwendeten Variablen angezeigt, in die der PC Chair Werte für die Vorschau der neuen Version eintragen kann. Die angezeigten Daten werden von `GetEmailTemplate` generiert.
- `emailtemplatepreview-cont.wml` zeigt die neue Version eines Templates an. In einer Gegenüberstellung bekommt der PC Chair sowohl den Template Text und den Template Text mit den durch ihre zuvor eingegebenen Werten ersetzten Variablen angezeigt. Die dargestellten Daten werden von `VerifyEmailTemplate` erzeugt, wenn keine Syntaxfehler gefunden wurden.

Testphase

Um sicherzustellen, dass die neuen Templates ihre Arbeit wie oben beschrieben erfüllen, wurde der gesamte Workflow getestet. Zu Beginn erfolgt das Laden der Übersicht der Templates. Hierfür ist `GetEmailTemplatesOverview` zuständig. Bedingung war hier, dass nur Namen der Templates, die auf `_dfmt.txt`, bzw. Beschreibungen die auf `_dfmt.desc.txt` enden, geladen werden. Dieser nicht sonderlich komplizierte Filter arbeitete von Anfang an korrekt.

Der nächste Schritt, das Anzeigen aller Versionen eines gewählten Templates durch `GetEmailTemplateVersions` wird mittels eines Filters, der dem in `GetEmailTemplatesOverview` gleicht, realisiert. Einziger Unterschied ist hier, dass auf `template_name_` gefolgt von `.txt`, bzw. `.desc.txt` gefiltert wird.

Anspruchsvoller war das Testen von `GetEmailTemplate`, da dieser SIB auch dafür zuständig ist, die Variablen, die in einem Template benutzt werden, zu erkennen. Nach einigen Anpassungen war `GetEmailTemplate` in der Lage, die benutzten Variablen als solche zu erkennen. Näheres zu der Syntax der Variablen im nächsten Abschnitt "Der Parser".

Der Parser, der für das Erkennen der Variablen innerhalb der Templates zuständig ist, wird sowohl von `GetEmailTemplate` als auch von `VerifyEmailTemplate` eingesetzt. Die beiden SIBs unterscheidet lediglich, dass `GetEmailTemplate` keine Überprüfung der

syntaktisch korrekten Verwendung der Variablen macht. Daher wird der Parser anhand von `VerifyEmailTemplate` beschrieben.

Variablen müssen mit einem Dollarzeichen beginnen. Darauf dürfen alphanumerische Zeichen, der Punkt und der Unterstrich folgen. Klammern in Variablennamen sind ebenfalls zulässig, sofern eine gleiche Anzahl öffnender wie schließender Klammern gegeben ist (es handelt sich um runde Klammern). Nach einer Variablen dürfen direkt Satzzeichen stehen. Eine Ausnahme bildet der Punkt. Er muss mit einem Leerzeichen von der Variablen getrennt werden.

Der Parser wurde auf die soeben beschriebene Syntax getestet und erkannte die Variablen erfolgreich. Lediglich solche Variablen, die direkt am Ende des Templates standen, wurden erst nach kleineren Anpassungen in der Parserimplementierung erkannt. Im Einsatz innerhalb des OCS gelten alle Variablen, auch syntaktisch korrekte, als unbekannt, die nicht schon von `GetEmailTemplate` im Standard Template gefunden werden. Dadurch wird sichergestellt, dass keine Variablen verwendet werden können, die unter Umständen beim Versenden des jeweiligen Templates unbekannt sind.

Das Speichern der Templates wird von dem SIB `SaveEmailTemplate` vorgenommen. Zu den Aufgaben des SIBs gehört auch das Finden der richtigen Versionsnummer. Die gefundene Versionsnummer wird sowohl für die Datei, in der das Template steht, als auch für die Datei, in der die Beschreibung abgelegt ist, benutzt.

`SaveEmailTemplate` war in der Lage, die Versionsnummer eines gegebenen Templates korrekt zu erhöhen und speicherte es unter dem Dateinamen `template_name_version.txt`. Wenn das übergebene Template das Standardtemplate war, wurde die Versionsnummer 001 gewählt. Da das Inkrementieren der Versionsnummer geschieht, bevor eine Datei geschrieben wird, ist sichergestellt, dass kein vorhandenes Template überschrieben wird. Wenn das SIB eine Versionsnummer ermittelt, die grösser als 999 ist, schreibt es eine Fehlermeldung in den Call Context und speichert das Template nicht.

Kapitel 4

PG Hauptthemen

In der zweiten Hälfte der Projektgruppe wurden vier große Themen von den Gruppen separat bearbeitet. Dabei handelte es sich einerseits um Erweiterungen des OCS um zwei neue Features, andererseits um zwei zusätzliche Dienste. Im Vordergrund stand dabei die Erweiterung des OCS um Zusatzfunktionalitäten, die aufgrund von Anregungen durch Benutzer und Administratoren des OCS implementiert werden sollten.

Das MOS (Management Overview System) soll OCS Mitgliedern, die in mehr als einem Dienst angemeldet sind, eine bessere Übersicht der zur Verfügung stehenden Dienste und Logins bieten. Über den Support- und Feedbackdienst soll es möglich sein, eine rollenabhängige FAQ zu generieren, sowie Supportanfragen und Umfragen strukturiert bearbeiten zu können. Das Reporting Feature dient zur Generierung von Statusreports, welche Rollenabhängig den Entscheidungsprozess aufbereiten und mittels des Production Feature sollen nach Abschluß der Diskussion druckreife Dokumentationen erstellt werden.

4.1 Management Overview System (MOS)

Beschreibung der Aufgabe

Das Management Overview System (MOS) ist ein personalisierter Dienst, der die Kontrolle und Koordination mehrerer aktiver Instanzen des OCS erleichtert. Das MOS ist auf der gleichen technologischen Basis entwickelt worden wie der Online Conference Service (OCS).

Modellierungsphase

Aufgrund des Umfangs dieser Arbeit ist eine ausgiebige Modellierung in UML hier unverzichtbar. Zunächst zeigt das Use Case in Abbildung 4.1 die Funktionen, die das System einem MOS User bietet. Dieser hat die Möglichkeit, sich nach dem erfolgreichen Einloggen einen Überblick über aktuelle Instanzen zu verschaffen (Conferences) und

diese auch zu verwalten. Dabei ist er in der Lage Rollen im System zu verwalten und sofern der User über mehrere verfügt, diese auch zu wechseln.

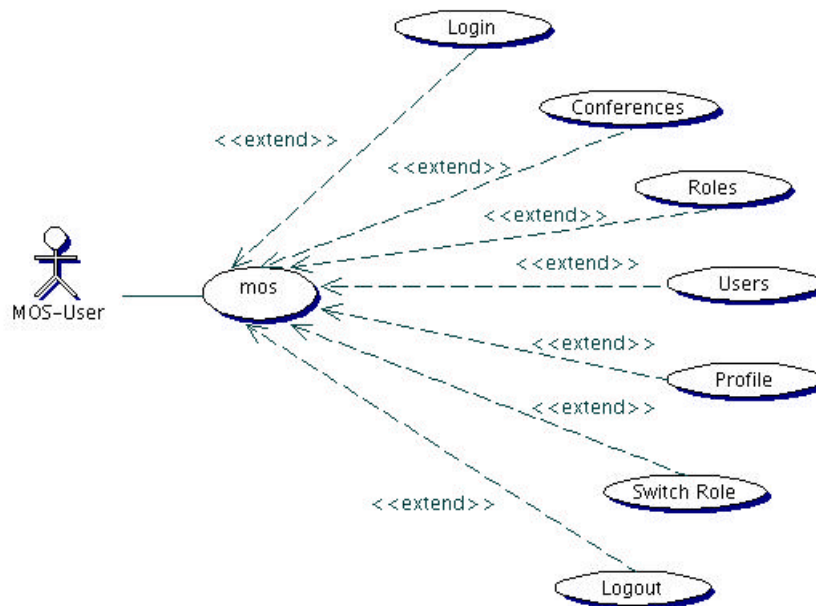


Abbildung 4.1: Use Case: „MOS Features“

Konferenz-Liste

Über eine tabellarische Liste (Abb. 4.2) der aktiven Konferenzen können diese bearbeitet werden: Der User kann Konferenzen löschen, sich darin einloggen, den Statusreport der Konferenz anzeigen lassen (siehe Gruppenthema 4), den Namen, die URL und das Passwort der Konferenz ändern. Desweiteren kann er hier die Zugriffsberechtigung der MOS User zur Konferenz verwalten.

Datenbankmodell

Das MOS benötigt zwar eine dem OCS ähnliche, jedoch komplett neue Datenbank. Diese besteht aus 9 Tabellen: template, textmacro, permission, role, rolepermission, userrole, conferences, userconferenceaccess und edtuser (Abb. 4.3)

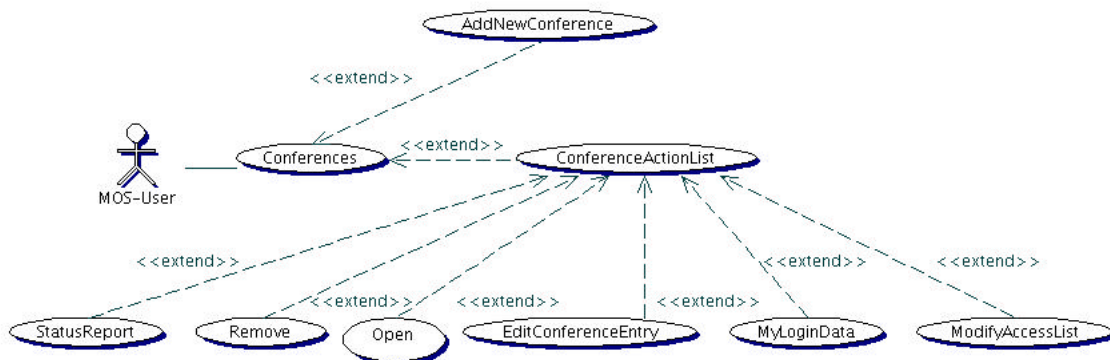


Abbildung 4.2: Use Case: „Conference-Features“

Hinzufügen neuer Konferenzen

Das Aktivitätsdiagramm in Abbildung 4.4 illustriert das Hinzufügen einer neuen Konferenz. Nach dem Ausfüllen des Formulars (Name, MOS-Passwort, URL) werden die Eingaben überprüft. Sind die Eingaben korrekt, wird nach einer Bestätigung die neue Konferenz in die Datenbank aufgenommen.

Einloggen in eine OCS-Instanz

Das Einloggen in eine OCS Instanz über das Management Overview System geschieht über die URL, die Emailadresse des Users und dem eindeutigen Shadow Passwort pro User und pro Konferenz. Sofern einem User die Zugangsberechtigung zu einer OCS Instanz erteilt worden ist, verfügt er über ein Shadow Passwort zu dem OCS Dienst, das in der Datenbanktabelle *userconferencesaccess* eingetragen ist. Der User wird automatisch mit Hilfe seiner Emailadresse und dem Shadow Passwort in den OCS eingeloggt (Abb. 4.5).

Zugang zum Statusreport

Zum Betrachten des Statusreports muss der User ebenfalls auf bestimmte Weise in den OCS Dienst eingeloggt werden. Damit dies unabhängig davon geschehen kann, ob dem User die Zugangsberechtigung im OCS gegeben wurde, erfolgt der Einlogg-Mechanismus über das MOS Passwort. Hierüber kann der OCS den User authentifizieren und ihn als einen speziell hierfür angelegten User *DefaultMosUser* einloggen (Abb. 4.6). Ihm steht nur eine eingeschränkte Sicht auf den OCS Dienst zur Verfügung. Er kann lediglich den

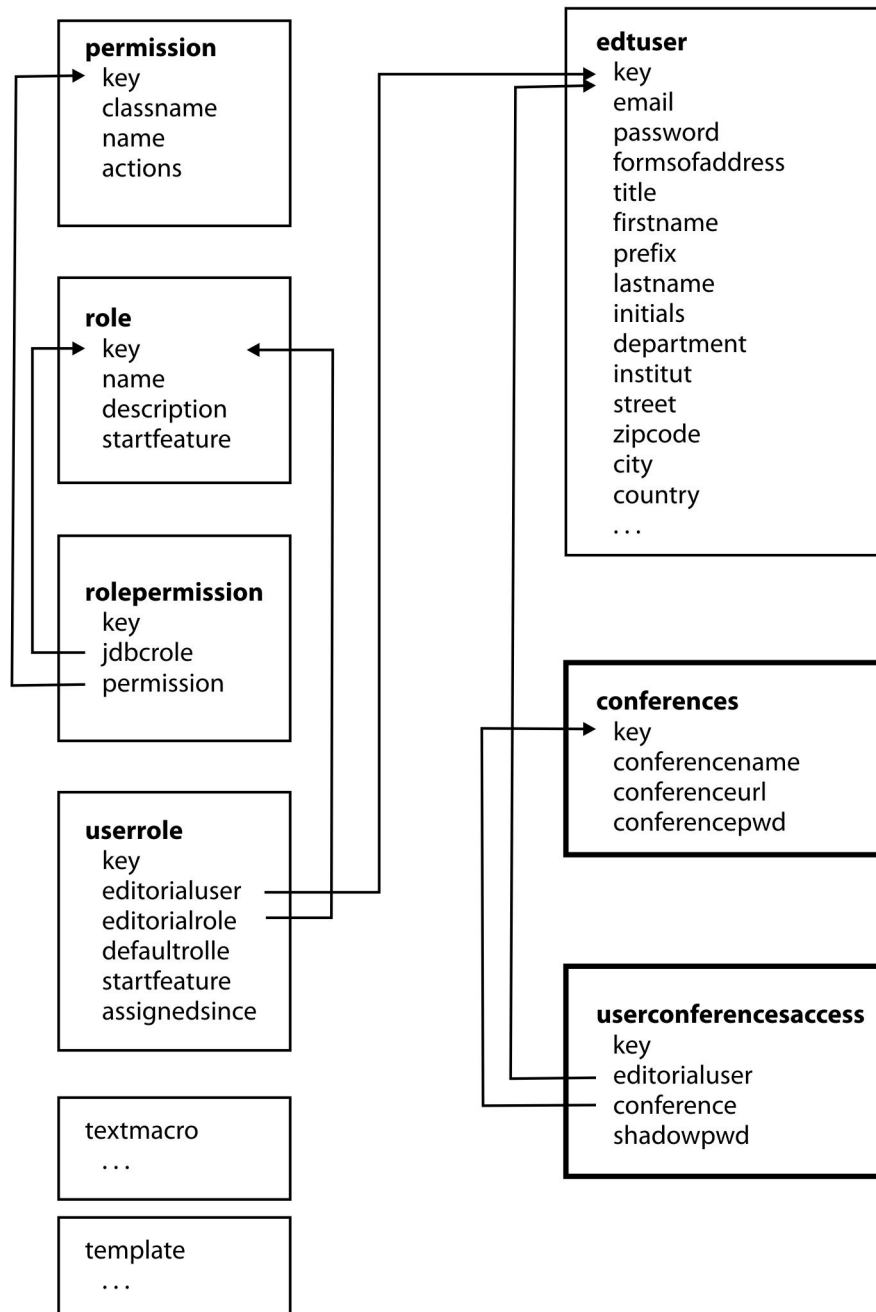


Abbildung 4.3: MOS: „Datenbankmodell“

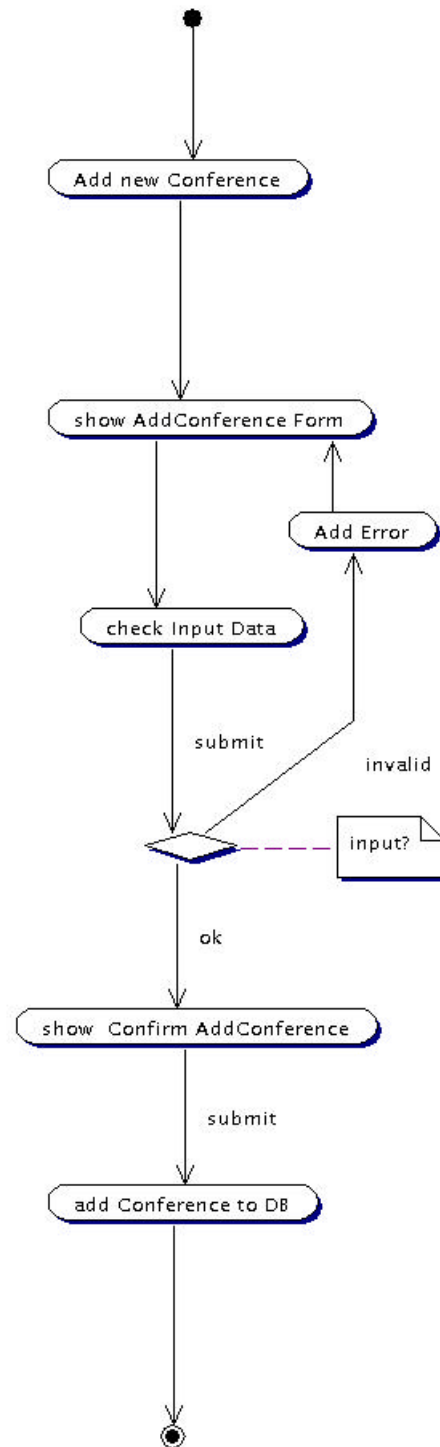


Abbildung 4.4: Activity Chart: „Neue Konferenz hinzufügen“

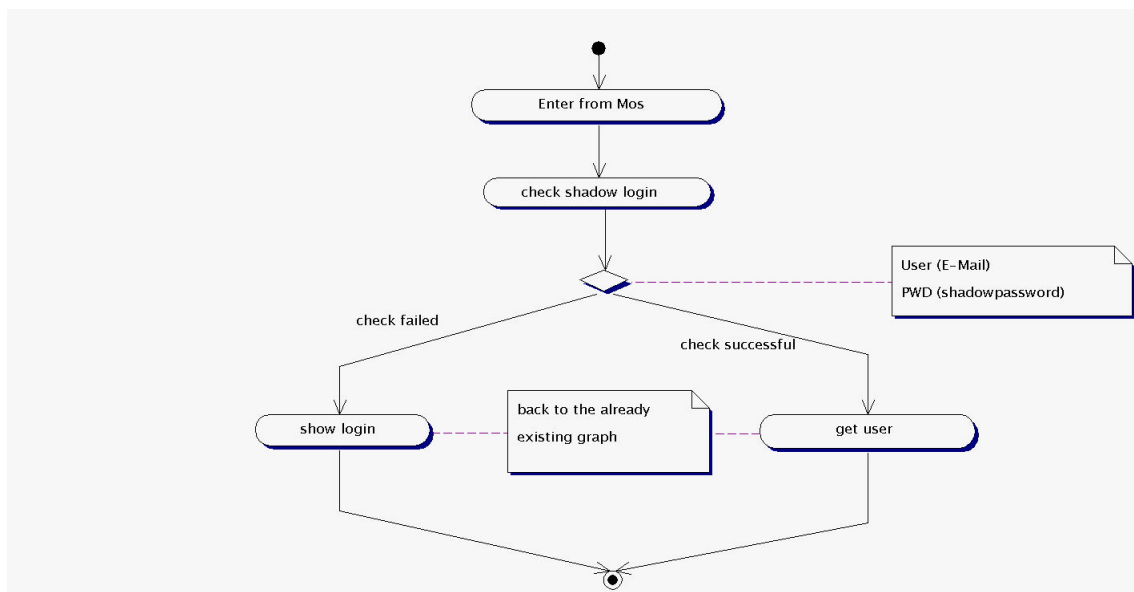


Abbildung 4.5: Activity Chart: „Login eines MOS-Users in einen OCS-Dienst“

Statusreport lesen. Im übrigen werden auch die MOS Passwörter eines OCS Dienstes in dessen web.xml festgelegt (MosPWD). Der *DefaultMosUser* wird bei Initialisierung der OCS-Datenbank angelegt. Seine Emailadresse und sein Passwort stehen außerdem in der web.xml als Parameter, können also nachträglich verändert oder gelöscht werden (falls kein Zugang zum Status-Report für den MOSUser gewünscht wird).

Konferenzdaten modifizieren

Über die Aktionsliste für jeden Eintrag in der Konferenzliste können die gewünschten Konferenzdaten bearbeitet werden. Name, URL und MOS Passwort können verändert werden Abb.(4.7). Nach dem Überprüfen der Eingaben und einer Bestätigung erfolgt die Aktualisierung der Datenbank.

Zugangsberechtigungen zu Konferenzen

Das Ändern der Zugangsberechtigungen zu einer Konferenz (die Möglichkeit sich direkt in den OCS-Dienst einzuloggen) erfolgt im Wesentlichen in zwei Schritten, wobei jeder Schritt eine Kontaktierung mit dem OCS beinhaltet. Die zu bearbeitende Konferenz wird festgelegt, indem aus der Aktionsliste die zugehörige Aktion ausgewählt wird. Zunächst werden anhand einer Checkbox die User ausgewählt, die Zugang zu dieser Konferenz haben sollen. Aus den ausgewählten Usern, die noch keine Zugangsberechtigung haben,

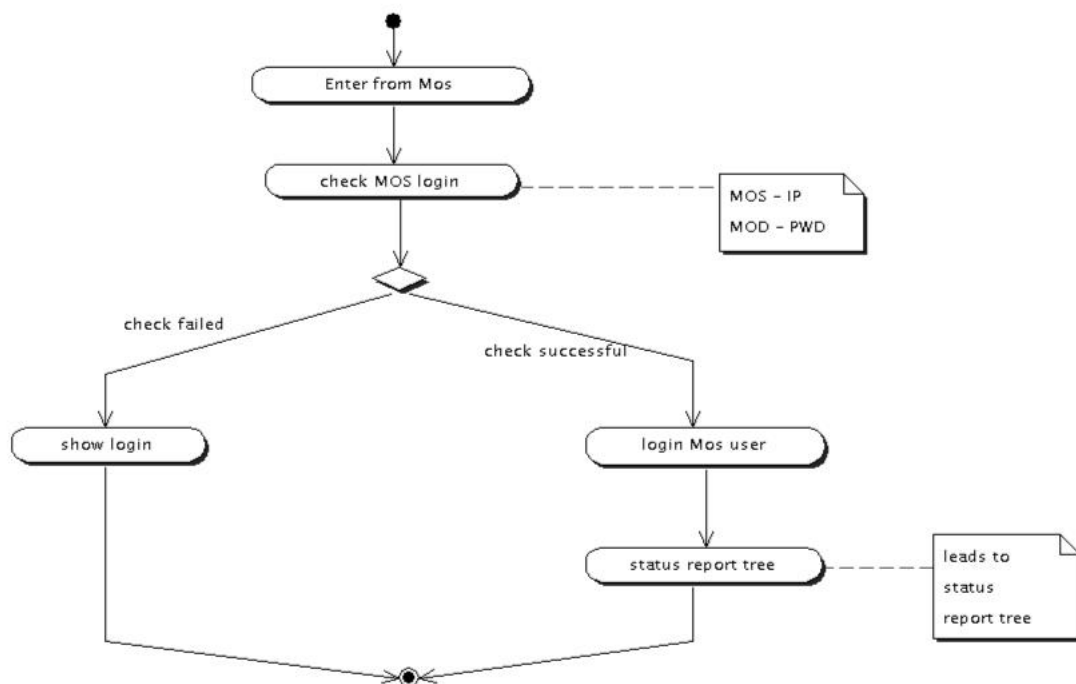


Abbildung 4.6: Activity Chart: „Login zum Statusreport (OCS-Seite)“

wird die Liste *grantUserAccess* erzeugt. User, die nicht ausgewählt wurden und bisher über eine Berechtigung verfügt haben, wird diese entzogen (es wird der entsprechende Eintrag in der *userconferencesaccess* Tabelle gelöscht). Im ersten Schritt zur Kontaktaufnahme mit dem OCS wird nun abgefragt, welche User aus der Liste *grantUserAccess* dort registriert sind. Bereits registrierte User kommen nun in eine Liste *foundlist*, welche die Emailadressen und die zugehörigen Shadow Passwörter enthält (jeder OCS-User hat ein eindeutiges Shadow Passwort). Die User aus dieser Liste erhalten nun in der *userconferencesaccess* Tabelle des MOS einen Eintrag (Abb. 4.8).

Im zweiten Schritt zur Kontaktaufnahme mit dem OCS werden die User in der *grantUserAccess* Liste, die noch nicht im OCS registriert waren, als User registriert. Der OCS Instanz werden hierzu alle notwendigen persönlichen Informationen über die User übermittelt. Alle über diesen Prozess registrierten User erhalten die Rolle *Mos User*. Als Antwort liefert die OCS Instanz wieder eine Liste mit Emailadressen und den zugehörigen Shadow-Passwörtern, die soeben erzeugt worden sind. Nun kann der MOS Dienst die Zugangsberechtigungen in die Tabelle eintragen.

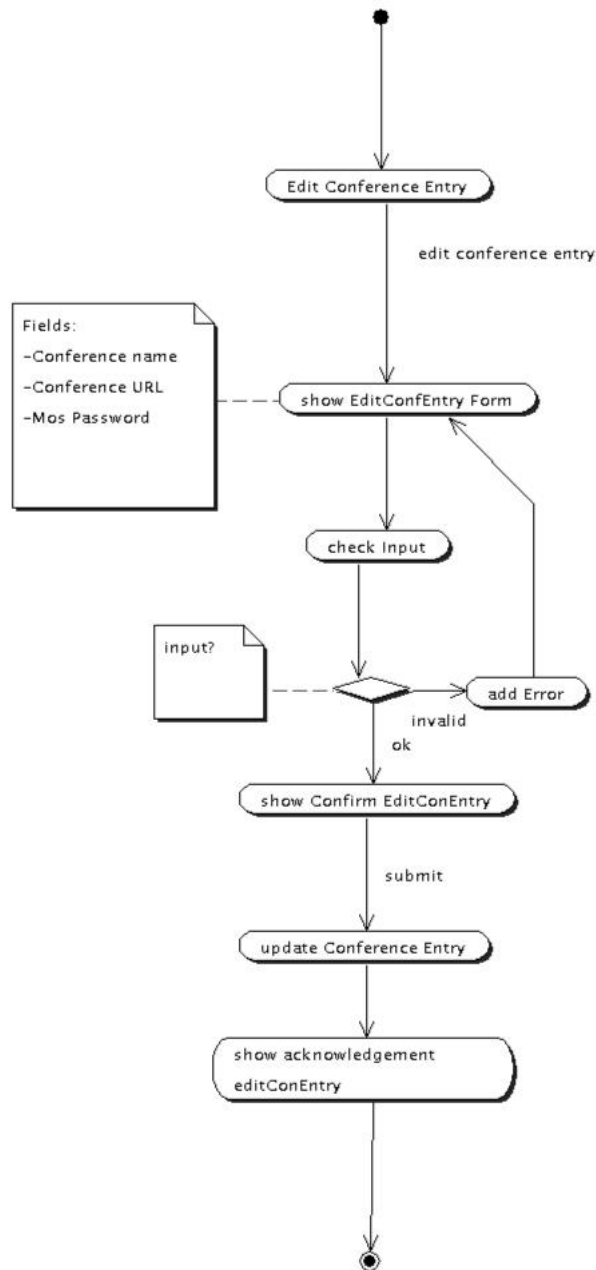


Abbildung 4.7: Activity Chart: „Konferenzdaten modifizieren“

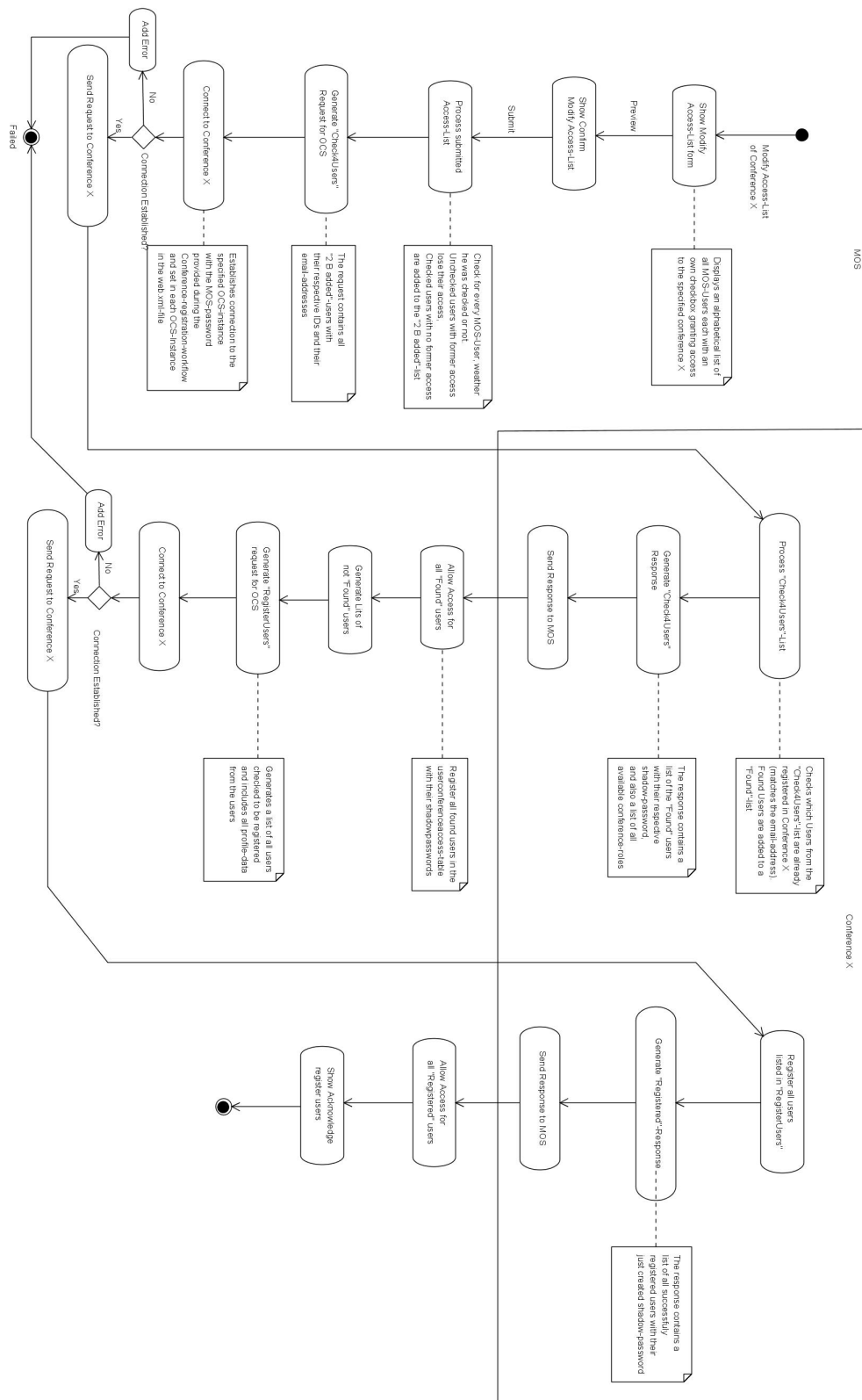


Abbildung 4.8: Activity Chart: „Bearbeiten der Zugangsberechtigungen zur Konferenz“

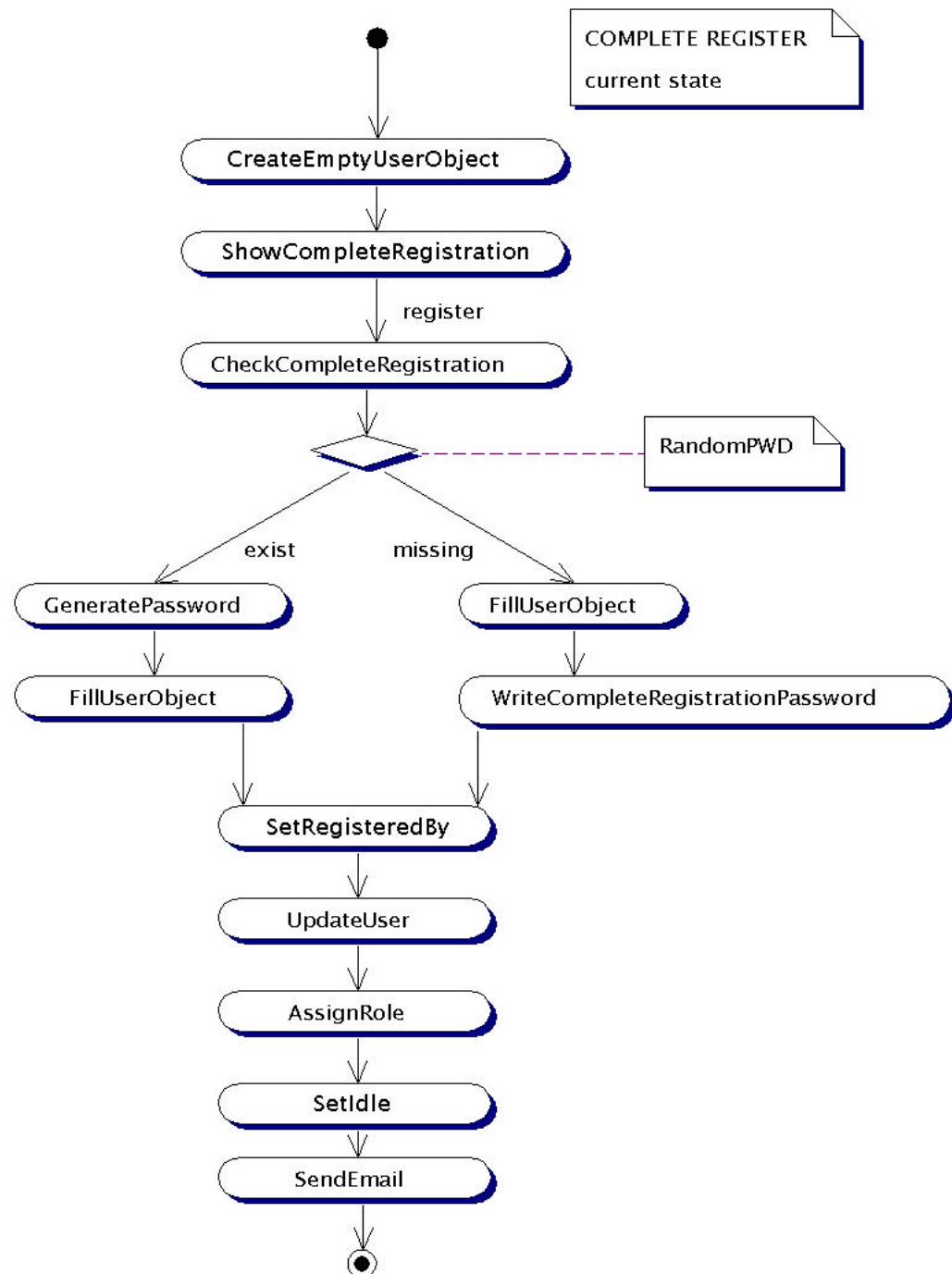


Abbildung 4.9: Activity Chart: „User-Registrierung ohne Shadow-Passwort“

User Registrierung

Das Registrieren eines Users im Online Conference Service muss nur leicht modifiziert werden (vgl. Abbildung 4.9 und Abbildung 4.10). Wie bereits erwähnt benötigt ein User des MOS, um sich darüber in einen OCS-Dienst einzuloggen, ein so genanntes Shadow-Passwort. Damit dieses zur Verfügung steht, muss es während des Registrierungsprozesses nach dem Ausfüllen des Registrierungsformulars generiert werden. Dieses Passwort existiert nur im Hintergrund - der User erfährt im Idealfall nie, dass es dieses Passwort gibt. Zum Beispiel kann er das Shadow-Passwort auf der Login-Seite der OCS-Instanz nicht zum Einloggen verwenden. Außerdem steht es dem User nicht frei, dieses Passwort zu verändern oder einzusehen.

Implementierungsphase

Bevor die Neuerungen, neue Features und Funktionen für MOS und OCS, implementiert werden konnten, musste eine funktionierende Basis für das Management Overview System erstellt werden. Dabei handelt es sich um eine vereinfachte Version des OCS, die nur über die für das MOS wichtigen Features verfügt (User-Management, Rollen-Management, Login, Profile). Erst nachdem eine funktionierende Basis für das MOS bereitstand, konnte damit begonnen werden den neuen Conferences-Bereich für das MOS und die Schnittstellen im OCS zu implementieren.

Neuerungen auf MOS-Seite:

- MOS-01 Konferenz-Liste (F-CNFR-01, F-CNFR-01.all, F-CNFR-01.own) (Abb. 4.11)
- MOS-02 Neue Konferenz hinzufügen (F-CNFR-02) (Abb. 4.12)
- MOS-03 Konferenz bearbeiten (F-CNFR-03)
- MOS-04 Zugangsberechtigungen zur Konferenz bearbeiten (F-CNFR-04) (Abb. 4.13)
- MOS-05 Konferenzen entfernen (F-CNFR-05)
- MOS-06 Status-Report lesen (F-CNFR-06)
- MOS-07 In die Konferenz einloggen (bei erteilter Zugangsberechtigung)

Neuerungen auf OCS-Seite:

- OCS-01 Automatische Generierung von Shadow-Passwörtern
- OCS-02 Login-Unterstützung von MOS-Usern über Emailadresse und Shadow-Passwort
- OCS-03 Schnittstelle zur Ermittlung bereits registrierter User und deren Shadow-Passwörter

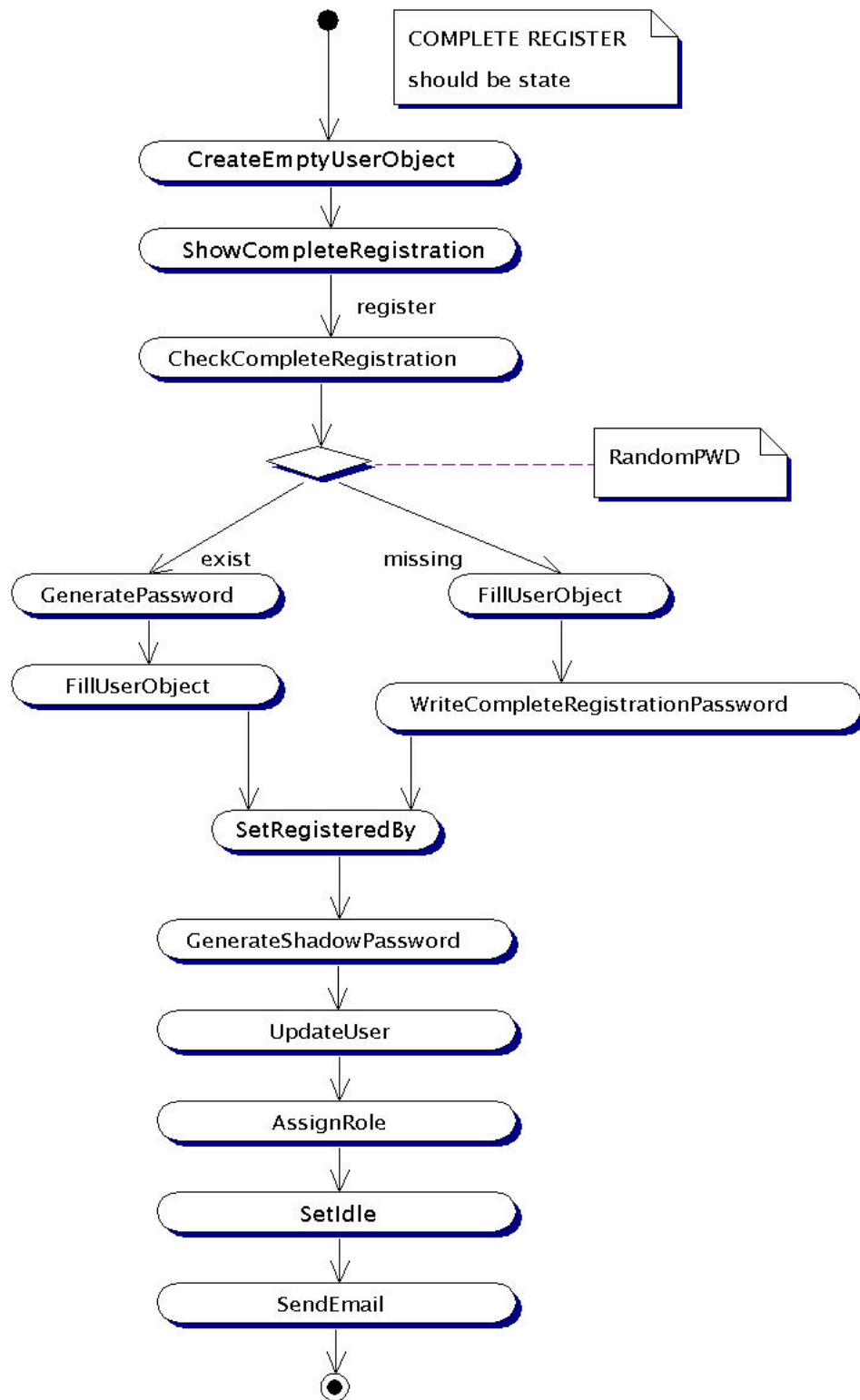


Abbildung 4.10: Activity Chart: „User-Registrierung mit Shadow-Passwort“

OCS-04 Schnittstelle zur Registrierung von Usern und der Rückgabe deren Shadow-Passwörter

OCS-05 Schnittstelle zum speziellen Einloggen in den OCS um den Status-Report zu lesen

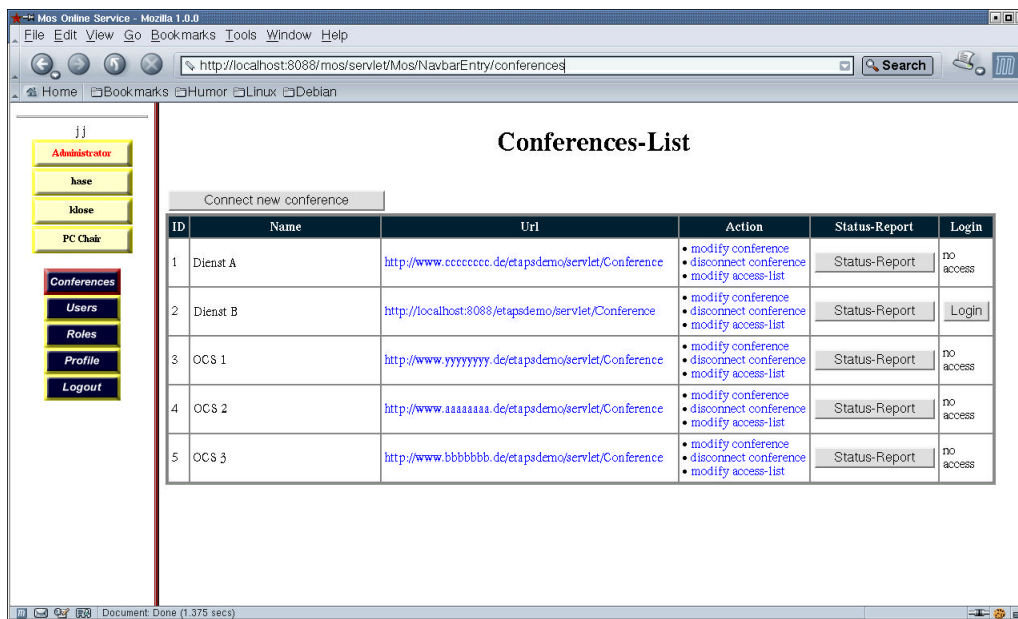


Abbildung 4.11: „Konferenz-Liste“

Neue SIBs

Beim Erstellen des Management Overview Systems war eine pattern-orientierte Wiederverwendung mehrerer Komponenten und Features aus dem Online Conference Service möglich, da beide Systeme auf derselben technologischen Basis beruhen. Verständlicherweise mussten dennoch diverse SIBs und Funktionseinheiten neu geschrieben werden. Die neu geschriebenen SIBs sind nachfolgend nach ihrer Funktion geordnet aufgelistet.

Neue SIBs auf OCS-Seite

MosCommunicationEntry (OCS-03, OCS-04, OCS-05)

Eintritts-SIB im OCS für jegliche Kommunikation zwischen MOS und OCS. Hier wird festgestellt, ob der Request Parameter *mospwd* nicht leer ist. Dies ist ein Test, um festzustellen, ob der Request von einem MOS Dienst stammt.

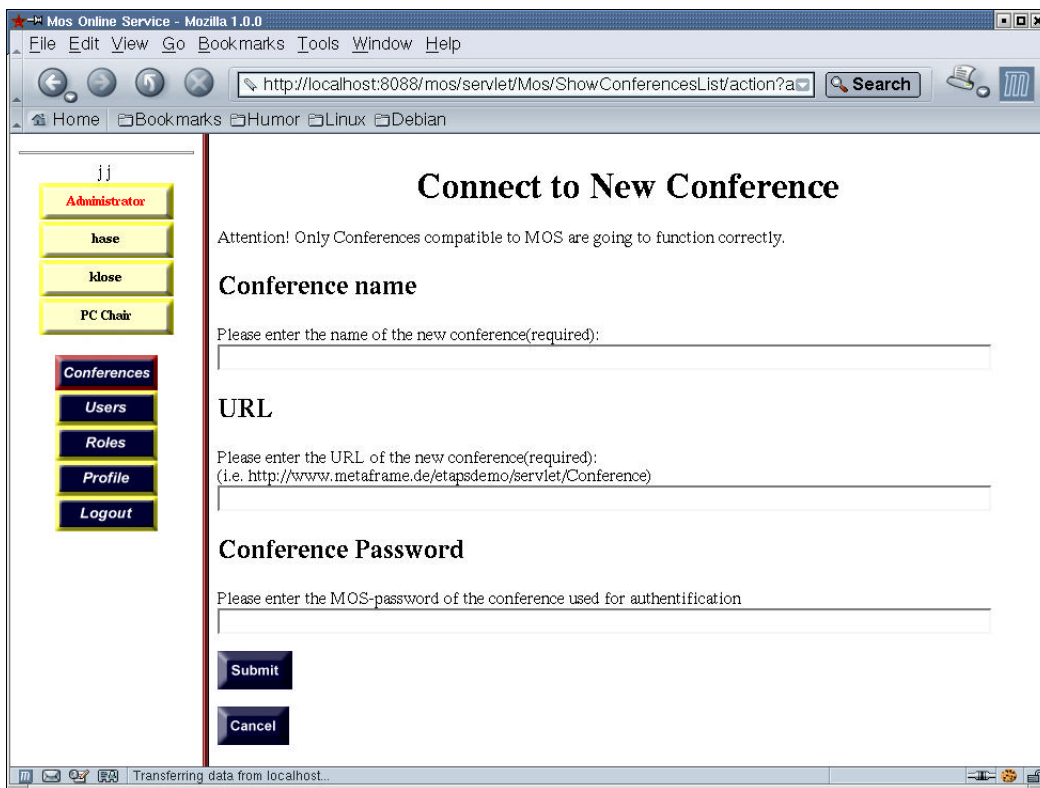


Abbildung 4.12: „Neue Konferenz hinzufügen“

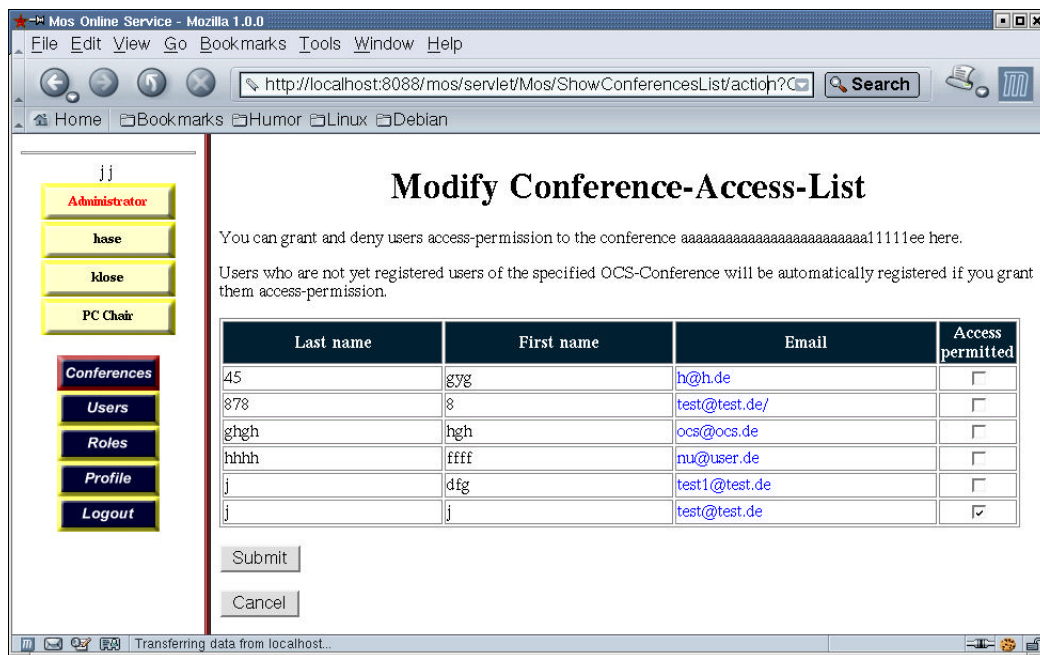


Abbildung 4.13: „Zugangsberechtigungen zur Konferenz modifizieren“

CheckMosPWD (OCS-03, OCS-04, OCS-05)

Dieser SIB überprüft, ob das MOS Passwort aus dem Request gültig ist. Wenn dies der Fall ist, wird der *successful* Zweig durchlaufen, andernfalls der *failed* Zweig. Falls der SIB Parameter *checkIP* true ist, wird außerdem überprüft, ob die zum MOS Passwort zugehörige IP-Adresse korrekt ist. Die komma-separierten Parameter *MosPWD* und *MosIP* sind in der *web.xml* des OCS eingetragen.

Check4Users (OCS-03)

Der SIB erwartet im Request den Parameter *check4users*: Eine komma-separierte Liste aus Emailadressen, nach denen gesucht werden soll. Alle OCS User, die erfolgreich identifiziert werden, werden in einen Vektor *foundlist* kopiert, der in den Call Context geschrieben wird.

RegisterMosUsers (OCS-04)

Der SIB erhält einen String im Request (*users2register*). Darin sind jeweils 18 einzelne User-Informationen aufgelistet, die voneinander mit " , " getrennt sind. Die Parameter-Listen der einzelnen User sind durch " | " getrennt. Der SIB versucht alle User zu registrieren. Alle User, bei denen dieses funktioniert, werden zum Vektor *foundlist* hinzugefügt,

der abschließend in den Call Context geschrieben wird. Falls die Anzahl der Parameter nicht korrekt war, oder es bereits einen User mit der Emailadresse gibt, ist dies nicht der Fall.

FoundUsersOutput (OCS-03, OCS-04)

Dies ist der Ausgabe-SIB, der die Antwort an das anfragende MOS generiert. Er erwartet im Call Context den Vektor *foundlist*, anhand dessen er einen String generiert, der sowohl die Emailadressen als auch die Shadow Passwörter der gefundenen User enthält.

MosRequestErrorOutput (OCS-03, OCS-04)

Dieser SIB gibt Fehlerinformationen für fehlgeschlagene MOS Anfragen (*mos-error* im Call Context) aus. Ist keine Fehlerinformation gesetzt worden, wird eine Standardfehlermeldung an das MOS ausgegeben.

LoginAnonymMosUser (OCS-05)

Der SIB bereitet den Login eines MOS Users in den OCS Dienst vor, um den Statusreport anzuzeigen. Er schreibt die Parameter *defaultmospassword* und *defaultmosname* in den Call Context.

CheckMosLogin (OCS-05)

Dieser SIB übernimmt die Funktion des Login SIBs *CheckLogin* im Teilgraphen für die Anzeige des Status-Reports. Er überprüft, ob die vorher in den Call Context geschriebenen Login-Parameter *defaultmospassword* und *defaultmosname* korrekt sind, und ob es einen registrierten User mit dieser Emailadresse und Passwort gibt.

GenerateShadowPWD (OCS-01)

Der SIB erzeugt ein zufälliges Shadow-Passwort, das für den Login von MOS Usern in OCS Instanzen benötigt wird.

Neue SIBs auf MOS Seite

ShowConferencesList (MOS-01)

Zeigt die tabellarische Übersicht aller Konferenzen mit den jeweiligen Aktionen an. Der SIB filtert diejenigen Konferenzen, die dem User angezeigt werden dürfen.

Verwendetes Template: `conferencelist-cont.html`

ConferencesListActions (MOS-01)

Dieser SIB ist für die Weiterleitung zu den Aktions-Teilgraphen zuständig, die in der Konferenz-Liste verfügbar sind (create-new-conference, remove-conference, modify-conference, modify-accesslist).

GetConferenceThroughID (MOS-05)

Gibt ein *Conferences* Objekt anhand der *ConferenceID* im Call Context weiter.

ShowConfirmConferenceRemove (MOS-05)

Zeigt die Seite zum Bestätigen für das Entfernen einer Konferenz aus dem Dienst an. Verwendetes Template: confirmconferenceremove-cont.html

RemoveConferenceFromDB (MOS-05)

Löscht eine Konferenz (über eine *ConferenceID*) aus der Datenbank.

PreShowCreateNewConferencesForm (MOS-02)

Bereitet die Daten für das Formular zur Erzeugung einer neuen Konferenz vor.

ShowCreateNewConferences (MOS-02)

Zeigt das Formular (URL, Konferenz-Name und MOS-Passwort) zum Hinzufügen einer neuen Konferenz an. Verwendetes Template: createconference-cont.html

CheckSubmitConferencesForm (MOS-02)

Überprüft eingegebene Daten im Formular zum Hinzufügen einer neuen Konferenz. Es müssen alle Felder ausgefüllt sein und die angegebene URL muss mit "http://" oder "https://" beginnen.

CreateConference (MOS-02)

Erzeugt die im Formular (s.o.) beschriebene neue Konferenz.

PreShowModifyConferenceSubmitForm (MOS-03)

Bereitet die Daten für das Formular zur Bearbeitung einer bestehenden Konferenz vor.

ShowModifyConference (MOS-03)

Zeigt das Formular (URL, Konferenzname und MOS Passwort) zur Bearbeitung einer existierenden Konferenz an. Verwendetes Template: modifyconfsubmit-cont.html

CheckModifyConferenceForm (MOS-03)

Überprüft die angegebenen Daten im Formular zur Bearbeitung einer existierenden Konferenz.

ShowConfirmModifyConference (MOS-03)

Zeigt die Seite zum Bestätigen für das Bearbeiten einer existierenden Konferenz an. Verwendetes Template: confirmmodifyconference-cont.html

UpdateModifiedConference (MOS-03)

Aktualisiert den Datenbankeintrag der Konferenz mit den veränderten Daten.

PreModifyAccessList (MOS-04)

Bereitet die Daten für das Formular zur Veränderung der Zugangsberechtigungen einer Konferenz vor. Der SIB stellt fest, welche User eine Zugangsberechtigung zu der ausgewählten Konferenz besitzen. Die Konferenz ist über den Parameter *ConferenceID* im Request festgelegt. Der SIB setzt ein EditorialUser Array aller User in den Call Context unter dem Namen *UserList*. Dazu entsprechend wird ein String Array unter dem Namen *isChecked* in den *Call Context* geschrieben, in dem jeweils verzeichnet ist, ob der User mit dem gleichen Array-Index Zugang zur Konferenz hat (den Wert *checked* oder (leerer String) trägt).

ShowModifyAccessList (MOS-04)

Zeigt das Formular für die Veränderung der Zugangsberechtigung einer Konferenz an. In einer mit Checkboxen versehenen Liste von Benutzern können diesen Zugangsberechtigungen vergeben oder entzogen werden. Verwendetes Template: modifyconferenceaccess-cont.html

PreProcessAccessList (MOS-04)

Nachdem die modifizierte Access-Liste versandt wurde, generiert dieser SIB einen Vektor aus Usern (EditorialUser Objekte), deren Zugangsberechtigung auf die festgelegte Konferenz gelöscht werden soll (*removeUserAccess*) und einen Vektor aus Usern, denen eine Berechtigung gewährt werden soll (*grantUserAccess*).

RemoveUserAccess (MOS-04)

Der SIB entzieht den im Vektor *removeUserAccess* verzeichneten Usern die Zugangsberechtigung zu der ausgewählten Konferenz.

FindUserInOcs (MOS-04)

Der SIB stellt eine Verbindung zur OCS Instanz her, authentifiziert sich anhand des MOS Passwortes und erfragt, welche User in der *grantUserAccess* Liste in dem OCS-Dienst als User registriert sind. Anhand der zurück gelieferten Emailadressen und Shadow Passwörter der User erstellt der SIB einen EditorialUser Array aus den gefundenen Usern namens *foundusers* und einen String Array namens *foundshadowPWDs*, welcher die zugehörigen Shadow Passwörter enthält.

GrantUserAccess (MOS-04)

Der SIB gewährt allen Usern aus dem *foundusers* Array (vgl. SIB FindUserInOcs) eine Zugangsberechtigung zur ausgewählten Konferenz.

RegisterUserInOcs (MOS-04)

Der SIB stellt fest, welche User aus dem Vektor *grantUserAccess* nicht vom OCS identifiziert werden konnten und somit registriert werden müssen. Nachdem sich der SIB bei der Konferenz über das MOS Passwort authentifiziert hat, sendet er einen String, der alle User-Informationen der zu registrierenden User enthält, an den OCS-Dienst. Als Antwort erhält der SIB die Emailadressen und die Shadow Passwörter der erfolgreich registrierten User. Die registrierten User trägt der SIB in das Array *foundusers* ein und deren Shadow Passwörter in das Array *foundshadowPWDs*, damit sie als nächstes vom SIB *GrantUserAccess* die Zugangsberechtigung zur Konferenz erhalten können.

ShowAcknowledgeModifyAccesslist (MOS-04)

Der SIB zeigt zur Bestätigung der Modifikation der Zugriffsberechtigungen an, welche User Zugang haben, und welche nicht. Verwendetes Template: `acknowledgemodifyconferenceaccess-cont.html`

Sonstige SIBs

GetUserThroughID

Gibt einen User (EditorialUser) aus der Datenbank über seine ID zurück.

Neue Makros

conferences.smg (verwendet im MOS)

Das Makro enthält den gesamten Teilgraphen der Konferenzverwaltung.

user.smg (verwendet im MOS)

Für das MOS ist eine leicht modifizierte Version des *User* Makro notwendig. Das SIB *GetUserThroughID* wird benötigt, um die User ohne den Artikel-Bereich verwalten zu können.

Neue Administratoren

../editorial/conferences/ConferencesAdministrator.java

Der *ConferencesAdministrator* verwaltet die Tabelle *conferences*, in der die Konferenzen, die dem MOS bekannt sind, gespeichert werden.

Vom Administrator benötigte neue Klassen:

Conferences.java, *ConferencesIDGenerator.java*, *ConferencesIDComparator.java*, *ConferencesException.java*, *ConferencesDBMatching.java*, *ConferencesAdministratorException.java*

../editorial/conferences/UserConferencesAccessAdministrator.java

Der Administrator verwaltet die *userconferencesaccess* Tabelle, in der verzeichnet ist, welche User zu welchen Konferenzen Zugang haben und welche Shadow Passwörter sie darin haben.

Vom Administrator benötigte neue Klassen:

UserConferencesAccess.java, *UserConferencesAccessAdministratorException.java*, *UserConferencesAccessIDGenerator*, *UserConferencesAccessDBMatching.java*

Andere neue Klassen

../editorial/forms/CreateNewConferencesForm.java

Das Form wird für das Füllen der Formularfelder, für das Hinzufügen einer neuen Instanz eines OCS Dienstes, benötigt.

../editorial/forms/ModifyConferenceSubmitForm.java

Die Form wird für das Füllen der Formularfelder, für das Modifizieren einer bestehenden Instanz eines OCS-Dienstes, benötigt.

Änderungen

Editorial.java

Da die Klasse sowohl für OCS als auch für MOS verwendet werden muss, die beiden Dienste jedoch unterschiedliche Administratoren benötigen (das MOS braucht eine Vielzahl der vom OCS verwendeten Administratoren nicht, und der OCS benötigt die für das MOS geschaffenen neuen Administratoren nicht), musste hier dafür gesorgt werden, dass keine Administratoren initialisiert werden, für die die Datenbank Tabellen im jeweiligen Dienst fehlen. Um dieses zu gewährleisten wird, bevor ein Administrator initialisiert wird, geprüft, ob es eine Variable <administratorName>.doinit in der Datei web.xml gibt, die darüber Auskunft gibt, ob der Administrator initialisiert werden soll oder nicht. Falls die Variable nicht vorhanden ist, wird der Administrator initialisiert (diese Default-Behandlung stellt eine Rückwärtskompatibilität zu älteren Diensten sicher).

web.xml des OCS

Neue Parameter zur Kompatibilität zu MOS-Diensten

MosPWD: Eine durch Kommata getrennte Liste gültiger MOS Passwörter.

MosIP: Eine durch Kommata getrennte Liste gültiger IP Adressen von MOS Diensten (zu jedem MOS Passwort gehört eine separate IP).

DefaultMosUserName: Der Standard-Login für MOS User zum Einloggen in den Status-Reports.

DefaultMosUserPWD: Das Standard-Passwort für MOS User zum Einloggen in den Statusreport.

ConferencesAdministrator.doinit: Der Eintrag verhindert die Initialisierung des Conferences-Admins

UserConferencesAccessAdministrator.doinit: Der Eintrag verhindert die Initialisierung des UserConferencesAccess-Admins.

../create/edtuser.sql des OCS

Die Datenbank-Tabelle edtuser des OCS wurde um ein Feld namens *shadowpwd* erweitert, um das Shadow Passwort eines Users speichern zu können.

../dump/user.sql des OCS

Hier wurde der *DefaultMosUser* zum automatischen Eintragen in die Datenbank bei der Erzeugung eingetragen.

../dump/role.sql des OCS

Wird ein User des MOS automatisch als User im OCS registriert, so benötigt er eine Standardrolle, die ihm für den OCS-Dienst zugeteilt wird. Die *Rolle Mos User* muss also bei Initialisierung der OCS-Datenbank erzeugt werden.

../dump/userrole.sql des OCS

Um zu ermöglichen, dass der *DefaultMosUser* eingeloggt werden kann, benötigt der User zumindest eine Rolle. Bei Initialisierung der OCS-Datenbank wird dem *DefaultMosUser* die Rolle *Mos User* zugewiesen.

EditorialUser.java

Die Klasse *EditorialUser.java* wurde um die nötigen set- und get- Methoden erweitert, um auf die Variable *shadowPWD* zugreifen zu können.

CheckLogin

Der SIB wurde dahingehend erweitert, dass er prüft, ob der Request Parameter *spwd* gesetzt ist. Falls dies der Fall ist, handelt es sich um den Versuch eines MOS Users sich über einen MOS Dienst in die OCS Instanz einzuloggen. Es wird also nicht das reguläre Passwort *pwd* zum einloggen verwendet, sondern das Shadow Passwort *spwd*.

Testphase

Im Folgenden ist aufgelistet, welche Bereiche getestet wurden, um sicherzustellen, dass keine Fehler oder Mängel im Online Conference Service und im Management Overview System bestehen. Sollte es nicht ausdrücklich anders erwähnt sein, haben die Tests die Resultate geliefert, die erwartet wurden oder die entdeckten Fehler wurden entsprechend korrigiert.

Zwar wurden die Funktionen, die in der MOS-Basis zur Verfügung stehen, vom OCS übernommen (ihre Fehlerfreiheit dort ist anzunehmen), es ist dennoch nötig zu überprüfen, ob alle Features korrekt funktionieren. Ein vollständiger Regressionstest kam allerdings aus Zeitgründen nicht in Frage.

- Registrierung neuer User
- Erzeugung neuer Rollen
- Ändern des User-Profiles
- Wechseln zwischen Rollen
- Die Einhaltung der Permissions für den Conferences-Bereich
- Das Hinzufügen neuer Konferenzen
- Das Modifizieren neuer Konferenzen
- Das Entfernen von Konferenzen
- Das Einloggen in eine Konferenz
- Der Abruf des Status-Reports einer Konferenz
- Die Vergabe und der Entzug von Zugangsberechtigungen zu einer Konferenz

4.2 Support- und Feedbacksystem

Beschreibung der Aufgabe

Die Aufgabe der Gruppe 2 ist es, einen ergänzenden Dienst zum OCS zu erstellen, der folgendes ermöglicht:

- Strukturierte Bearbeitung von über Webformularen eingereichten Supportanfragen.
- Generierung und Verarbeitung von Umfragen zur Verbesserung des OCS Dienstes.
- Halbautomatische Erstellung und Verwaltung von FAQs, basierend auf bereits eingegangenen Supportanfragen.
- Pflfegbarkeit der Formulare, die dem Benutzer im öffentlichen Teil des Dienstes zur Dateneingabe bereitgestellt werden.

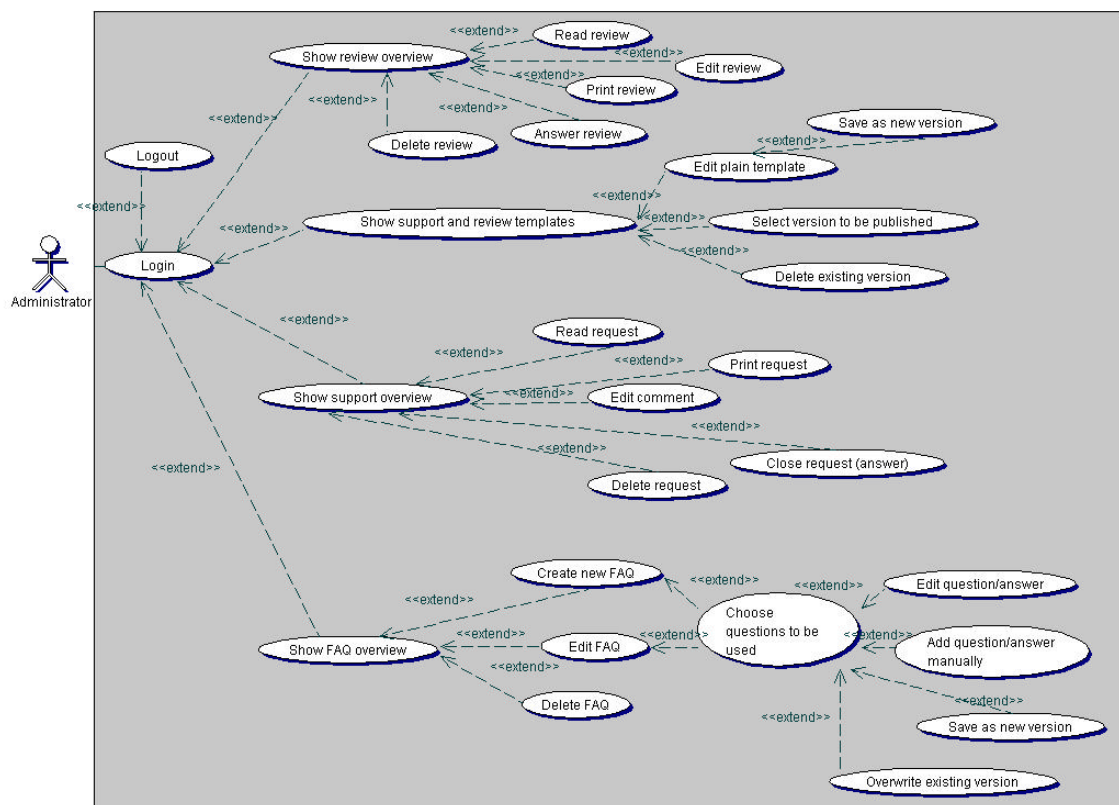


Abbildung 4.14: Use Case des OCS Support Dienstes

Der Dienst basiert auf einem bereits existierenden Modul *ewis survey* und einem daraus zu Demonstrationszwecken erstellten Dienst. Dieser Demodienst wurde als Grundlage

für den *OCS Support* Dienst verwendet und erweitert. Der Dienst soll später universell eingesetzt werden und muss daher in den wesentlichen Punkten konfigurierbar sein. Das Use Case in Abbildung 4.14 zeigt die möglichen Aktionen im Dienst.

Modellierungsphase

Der neue Support und Survey Dienst unterteilt sich in einen öffentlichen und einen administrativen Bereich. Der öffentlich zugängliche Teil besteht aus Online Supportanfrage via Webformular, Online Fragebogen und FAQ. Die FAQ existiert in mehreren Varianten für die verschiedenen Rollen im OCS Dienst, diese sogenannten *FAQ Kategorien* sind jederzeit erweiterbar. Der administrative Teil stellt verschiedene Bereiche zur Pflege der Webformulare, Generierung der FAQ und Beantwortung der Supportanfragen zur Verfügung, das Aktivitätsdiagramm in Abbildung 4.15 gibt eine Übersicht über den Aufbau des Dienstes.

Die von außen zugänglichen Webformulare werden aus XML Dateien generiert, welche die Struktur für die Eingabeformulare bereitstellen. Die FAQ wird hingegen aus den gespeicherten Reports mittels diverser Relationen generiert.

Die Webformulare sind im administrativen Teil nur als XML Datei direkt editierbar, bei einer späteren Erweiterung könnte hier noch ein zusätzliches grafisches Interface geschaffen werden.

Die Administration des Dienstes besteht aus folgenden Teilen:

- **Supportanfragen verwalten:** Erhaltene Supportanfragen werden tabellarisch dargestellt und können gelesen, kommentiert und beantwortet werden. Dabei wird jede Anfrage mit einem Status versehen, damit der Bearbeitungsstand der einzelnen Anfrage nachvollzogen werden kann. In einer FAQ benutzte Anfragen sind mit einem gesonderten Status versehen. Beantwortete Anfragen können gelöscht werden.
- **Feedback verwalten:** Ähnlich wie die Supportanfragen werden hier erhaltene Umfrageergebnisse gesammelt. Diese können wiederum gelesen oder beantwortet werden, falls der Benutzer im Formular einen begleitenden Kommentar oder eine Frage angefügt hat.
- **Templates editieren:** Hier ist es dem Administrator möglich die XML Dateien zur Generierung der Formulare für Support und Feedback als Klartext zu editieren, vor der endgültigen Abspeicherung wird die Eingabe auf XML Konformität überprüft. Jedes geänderte Formular wird als neue Version gespeichert, es ist anschließend möglich eine Template zu veröffentlichen. Alte Templates sind löschar.
- **FAQ verwalten:** Da es für verschiedene Benutzergruppen des OCS verschiedene FAQs geben kann, wählt der Administrator zunächst die zu bearbeitende FAQ-

Kategorie aus, danach gelangt er in eine Übersicht aller bisher erstellten FAQs der gewählten Kategorie. Alle generierten FAQs werden versioniert abgelegt und in einer Liste angezeigt. Es ist möglich eine bestimmte Version zu veröffentlichen, nur diese wird dann im öffentlichen Bereich angezeigt. Bereits angelegte FAQs können editiert und neue angelegt werden. Sowohl bei Erstellung als auch Veränderung von FAQs können abgeschlossene Supportanfragen und deren Antworten als Grundlage für die FAQ ausgewählt und einer Sektion zugeteilt werden. Die Sektionen dienen bei Anzeige der FAQ zur Unterteilung nach Themengebieten.

Da die meisten Anfragen nicht der Form entsprechen, die eine FAQ benötigt, sind Fragen und dazugehörige Antworten editierbar, diese veränderten Texte werden bei Anlegen einer FAQ in die zugehörige Supportanfrage zurückgeschrieben. Soll ein Thema in die FAQ aufgenommen werden, welches noch nicht durch eine Supportanfrage abgedeckt wurde, kann eine Anfrage *künstlich* erstellt werden.

- **FAQ Einstellungen editieren:** Da der Supportdienst für verschiedene Webdienste eingesetzt werden kann, ist es nötig, die Kategorien (Rollen im OCS) und Sektionen (Unterteilung der FAQ, im OCS nach Features) editieren zu können.

Da die Fragen der FAQ aus den Supportanfragen und -antworten bestehen soll, ist eine Datenhaltung außerhalb der Reports nicht sinnvoll. Die vorgesehene Generierung der FAQ aus einer XML Datei wurde während der Designphase verworfen. Statt dessen sollen nur Relationen zwischen einer FAQ und mehreren Anfragen abgelegt werden. Zur Generierung der FAQ genügt es, eine Datei mit allen Assoziationen zu laden, die benötigten Reports nach zugeteilter Sektion auszulesen, diese zu laden und aus den Reports wiederum die Texte für Frage und Antwort der FAQ zu extrahieren. Dem Dienst steht keine Datenbank zur Speicherung der Informationen zur Verfügung, daher werden sämtliche Informationen in Textdateien abgelegt. Der Verzicht auf eine Datenbank erlaubt es, den Dienst relativ einfach auf einem Server zu installieren.

Wird eine neue FAQ angelegt, werden in der Eingabemaske für eine FAQ zunächst die Frage- und Antwortpaare aller geschlossenen (d.h. beantworteten) Supportanfragen angezeigt. Die Fragen und Antworten werden in einem Textfeld dargestellt, damit die Formulierungen der Anfragen bei Bedarf an den Stil einer FAQ angepasst werden können. Der Administrator hat die Möglichkeit Frage- und Antwortpaare wie in Abbildung 4.17 dargestellt individuell über eine Checkbox hinzu- oder abzuwählen. Eine Supportanfrage, die in einer FAQ verwendet wird, bekommt den Status *FAQUSED*. Dieser Status wird auch in der Supportverwaltung (Abbildung 4.18) angezeigt. Die Zuordnung zu einer FAQ Sektion erfolgt mittels Auswahl per Dropdown. Da es vorkommen kann, dass eine in die FAQ aufzunehmende Frage noch nicht über den Supportdienst beantwortet und damit im FAQ System verfügbar gemacht wurde, kann der Administrator in der FAQ Editiermaske ein zusätzliches Frage- und Antwortpaar eingeben. Der Dienst generiert daraus eine reguläre, geschlossene Supportanfrage, die in der FAQ verwendet werden kann. Beim Editieren einer FAQ werden die bereits benutzen Anfragen am Anfang der Editiermaske in ihren jeweiligen Sektionen dargestellt. Erst darunter folgen alle

weiteren, noch nicht verwendeten Fragen in einem separaten Abschnitt der Eingabemaske.

Um die Relation zwischen einer FAQ und den zugrunde liegenden Supportanfragen herzustellen, verwendet der Support Dienst Properties-Dateien, in denen die IDs der Anfragen gespeichert werden. Zur Unterteilung der FAQ in Sektionen sind die IDs unter dem jeweiligen Sektionsnamen in der Datei abgelegt und über diesen auslesbar. Die zur Verfügung stehenden Sektionen sind in einer einfachen Textdatei im Klartext abgelegt, wobei die Werte hintereinander durch ein Komma getrennt abgelegt werden. Diese Datei ist im Dienst über eine Webseite als Text editier- und erweiterbar (Abbildung 4.16).

Aufgrund der Allgemeinheit des Support Dienstes kann eine beliebige Anzahl von individualisierten FAQs generiert werden. Dieses ist nötig, um zum Beispiel für den Online Conference Service getrennte FAQs für die Rollen PC Chair und PC Member zu erstellen. Im Support Dienst werden diese Rollen neutral als *Kategorie* bezeichnet. Die im Dienst verfügbaren Kategorien werden ebenfalls in einer Textdatei durch Komma getrennt abgelegt und sind auch im Dienst über eine Webseite veränderbar (ebenfalls Abbildung 4.16). Die Anzahl und die Reihenfolge der Kategorienamen ist dabei beliebig.

Innerhalb jeder Kategorie können mehrere versionierte FAQs abgelegt werden, von denen eine zur Anzeige im öffentlichen Teil des Dienstes ausgewählt werden kann. Bereits angelegte FAQs können editiert und gelöscht werden. Eine generierte und veröffentlichte FAQ wird im öffentlichen Teil wie in Abbildung 4.19 dargestellt.

Der bereits existierende Demodienst verwendet XML Dateien als Grundlage für die Erzeugung der Eingabeformulare, die der Benutzer beim Einreichen einer Umfrage oder Formulierung einer Supportanfrage benutzt. Diese Dateien sollen durch ein Webinterface im administrativen Teil des Dienstes leichter pflegbar gemacht werden. Die neue Templateverwaltung ermöglicht es, nach Auswahl einer der beiden Template Arten, die der jeweiligen Template zugrunde liegende XML Datei wie in einem Texteditor zu editieren. Die Eingabemaske für den XML Code ist in Abbildung 4.21 dargestellt. Veränderte oder neue Templates werden automatisch als neue Version gespeichert. Die neue Template wird vor der Speicherung auf XML Konformität überprüft. Sollte der Aufbau nicht der XML Spezifikation genügen, wird die Datei erneut mit einer Fehlermeldung versehen zur Überarbeitung angezeigt. Der Administrator kann in der Template Übersicht (Abbildung 4.20) eine Template zur Veröffentlichung auswählen oder alte Templates löschen. Darüber hinaus bietet die Templateübersicht noch die Möglichkeit, das fertig generierte Eingabeformular als Vorschau einer XML Template anzuzeigen.

Implementierungsphase

Die bereits bestehenden Teile des Demodienstes wurden nur wenig überarbeitet. Die wichtigste Änderung ist das Hinzufügen einer Löschfunktion in der Supportverwaltung, um bereits beantwortete Supportanfragen entfernen zu können. Aufgrund der Tatsache,

dass das System nur noch Assoziationen zur Erzeugung der FAQ ablegt und dieses auch ohne Datenbank zu erfolgen hat, müssen für die Implementation mehrere Dateien, deren Inhalte und deren Aufbau definiert werden. Das FAQ System benötigt folgende Dateien, um FAQs, FAQ Kategorien und FAQ Sektionen miteinander in Zusammenhang zu setzen:

FAQ_categories.txt: Definition der FAQ Kategorien, im OCS entsprechend der angelegten Rollen. Die Werte werden in dieser Datei durch Komma getrennt abgelegt.

Aufbau:

PCChair,PCMember,Author

FAQ_sections.txt: Definition der Sektionen, in welche die FAQ unterteilt werden kann, im OCS entsprechend der Features. Auch hier werden durch Komma getrennte Werte gespeichert.

Aufbau:

Bidding,Submission,Delegation,Setup

PCChair_001.properties: Datei, welche die eigentliche FAQ darstellt. Hier werden die Werte entsprechend einer Properties Datei abgelegt. Zu jeder Sektion der FAQ werden die IDs der dazugehörigen Supportanfragen gespeichert. Im Beispiel würde die Sektion *Delegation* keinen Eintrag enthalten.

Aufbau:

Bidding=5,7,3,14
Submission=9,16,1
Delegation=
Setup=12,10,2

FAQ_published.properties: Datei, welche den Dateinamen der veröffentlichten FAQ zu jeder Kategorie speichert.

Aufbau:

PCChair=PCChair_003.properties
PCMember=PCMember_0021.properties
Author=Author_003.properties

Die Verzeichnisse, in denen diese Dateien abgelegt werden, sind entweder über SIB Parameter oder Werte in der WEB-XML Datei des Dienstes einstellbar. Der Dateiname

der FAQ Properties Datei wird automatisch aus der zugehörigen Kategorie und der Versionsnummer gebildet. Die Dateinamen der Kategorie- oder Sektionsdatei sind im Prinzip variabel, vorausgesetzt die SIB Parameter der verarbeitenden SIBs werden entsprechend angepasst.

Folgende Java Klassen und SIBs wurden für die Generierung der FAQs neu geschrieben:

FAQElement Diese Klasse speichert die Informationen, die für die Anzeige oder Bearbeitung einer FAQ pro Frage- und Antwortkombination benötigt werden. Durch diese Klasse ist es nicht notwendig auf einem kompletten Report Objekt zu arbeiten. Die Klasse besitzt die Methoden *getAnswer()*, *setAnswer()*, *getQuestion()*, *setQuestion()*. Die Klasse wird von *PreShowFAQEdit* benutzt, um die Daten für die FAQ Editiermaske aufzubereiten.

PreShowFAQCategories Dieser SIB liest die zur Verfügung stehenden FAQ Kategorien aus und stellt sie der Template zur Auswahl der Kategorie in einem Vektor zur Verfügung. Die WML Template generiert aus dem Vektor ein Dropdown zur Kategorieauswahl.

PreShowFAQOverview Anhand der in *PreShowFAQCategories* ausgewählten Kategorie werden im Verzeichnis der FAQ Propertiesdateien auf dem Server alle passenden Dateinamen ausgelesen und in einen Vektor kopiert. Zusätzlich wird aus der Datei, welche alle veröffentlichten FAQs speichert, der Dateiname der aktuell für diese Kategorie veröffentlichten FAQ bestimmt. Der Vektor wird zusammen mit dem Dateinamen in den Call Context kopiert und das WML Template zur Anzeige der FAQ Übersicht generiert daraus die tabellarische Darstellung aller abgelegten FAQs zur ausgewählten Kategorie. Die momentan veröffentlichte FAQ wird in der Tabelle gesondert hervorgehoben und kann nicht gelöscht werden.

PreShowFAQEdit Diese Klasse lädt zunächst alle Supportanfragen mit dem Status *closed* oder *faqused* mit Hilfe des *report_handler*. Die ausgelesenen Objekte vom Typ *Report* werden in einer Hashtable *TempReportPool* abgelegt. Danach wird die Properties Datei zur aktuell zu editierenden FAQ geladen, über die angelegten FAQ Sektionen iteriert und die entsprechenden Beitrags IDs ausgelesen. Anschließend wird über die IDs der aktuell durchlaufenen Sektion iteriert. Enthält der *TempReportPool* die aktuelle ID, wird sie aus der Hashtable entfernt, anhand der ID der passende Report geladen und die Frage und Antwort dieser Anfrage in ein *FAQ Element* kopiert. Dieses FAQ Element wird dann dem Vektor *Used Reports* hinzugefügt. Nach durchlaufen aller IDs zur aktuellen Kategorie wird dieser Vektor in einer Hashtable unter dem Kategorienamen abgelegt.

Nach dem Durchlauf aller Sektionen werden die Report Objekte zu den verbliebenen IDs im *TempReportPool* geladen, die Fragen und Antworten in *FAQ Elemente* kopiert und an einen Vektor *NotUsedReports* angehängen.

Beide Vektoren werden später vom WML Template ausgewertet und die entsprechende Eingabemaske generiert.

StoreFAQ Dieser SIB wertet alle durch das Template *faq_edit* im User Request übermittelten Informationen aus und generiert die Report ID Listen zu jeder Kategorie, welche anschließend als FAQ Properties File gespeichert werden. Je nach Benutzerwahl wird entweder eine bereits existierende FAQ überschrieben oder eine neue erzeugt.

StoreFAQSettings Dieser SIB speichert die durch Komma getrennten Werte der FAQ Kategorien und Sektionen, welche mittels der zuvor angezeigten Template *faq_settings* editiert werden können. Eine spezielle Datenaufbereitung zur vorherigen Anzeige der Werte ist nicht notwendig, da zwei hintereinander geschaltete *ReadTextFile* SIBs die Strings aus beiden Dateien auslesen und in den Call Context kopieren, von wo aus das Template diese nur noch anzeigen muss. Nur zum Speichern der Werte ist dieses SIB notwendig, der restliche Zweig des Graphen besteht daher aus den standard SIBs *ReadTextFile* und *ShowFile*.

DeleteItem Anhand eines SIB Parameters bestimmt diese Klasse das Unterverzeichnis *Survey* oder *Support* und löscht den zuvor vom Benutzer ausgewählten Report.

PreShowFAQ Dieser SIB wertet die angesprungene URL und extrahiert aus dem angehangenen Parameter *category* die zu ladende FAQ Kategorie. Anschließend wird die derzeit für diese Kategorie veröffentlichte FAQ Properties Datei geladen und die FAQ generiert.

PublishFAQ Mit Hilfe dieses SIBs lässt sich eine FAQ veröffentlichen. Der gültige Dateiname wird unter dem Kategorienamen in einer Properties Datei abgelegt.

PreShowConfirmDeleteItem Dieses SIB liest den Typ und die ID des zu löschenden Reports aus und bereitet diese Informationen für die Anzeige einer Bestätigungsseite auf.

Folgende Java Klassen und SIBs wurden für die Bearbeitung der XML Templates neu geschrieben:

GetXMLTemplateOverview Anhand der zuvor vom Administrator ausgewählten Templatekategorie (Support oder Survey) durchsucht der SIB das entsprechende Verzeichnis und generiert eine Liste aller gespeicherten Templateversionen. Diese Liste wird aufbereitet und den nachfolgenden Template zur Anzeige übergeben.

GetXMLTemplate lädt ein XML Template, damit dieses editiert werden kann.

VerifyXML überprüft das editierte Template auf XML Konformität.

SaveXMLTemplate speichert das XML Template, wenn diese erfolgreich validiert werden konnte.

ActivateXMLTemplate kopiert ein XML Template in den für den Dienst festgelegten XML Template Dateinamen. Diese Datei wird dann vom Dienst als Grundlage für die Erzeugung des Eingabeformulars benutzt.

PreviewSurvey generiert ein Eingabeformular als Vorschau basierend auf einem abgelegten Template.

DeletePageItem löscht ein XML Template.

Testphase

Um den OCS Support Dienst zu testen, wurden diverse Supportanfragen und Reviews eingegeben und anschließend weiter verarbeitet. Die FAQ Generierung wurde getestet, indem diverse Kategorien und Sektionen über die *FAQ Settings* Eingabemaske eingetragen wurden. Dabei wurde insbesondere überprüft, ob die Kategorien und Sektionen anfällig gegenüber Leer- oder Sonderzeichen waren, da diese Zeichenfolgen später beim Auslesen der Properties Dateien eine zentrale Rolle spielen. Anschließend wurden mehrere FAQs zu jeder Kategorie angelegt.

Die Versionierung von FAQ Dateien, das Editieren bereits bestehender FAQs und die Anzeige der richtigen FAQ im öffentlichen Teil funktionierten ohne Probleme.

In der Testphase wurde ein Fehler behoben, der die Eingabe von Supportanfragen und Reviews betraf. Gab der Benutzer frei formulierten Text ein, wurde sämtlicher Text nach einem manuellen Zeilenumbruch mittels *Enter* abgeschnitten. Dieses Problem konnte durch Verwendung eines bereits modifizierten SIBs zur Speicherung der Reports behoben werden.

Beim Editieren der Formular Templates traten zunächst Fehler im Zusammenhang mit dem *Report Handler* auf, der zum Laden der Reports benutzt wird. Wird eine Template editiert, muss dieser neu initialisiert werden, da die Reports sonst nicht geladen werden können. Die Validierung der vom Benutzer editierten XML Datei wurde auf korrekte Funktion überprüft.

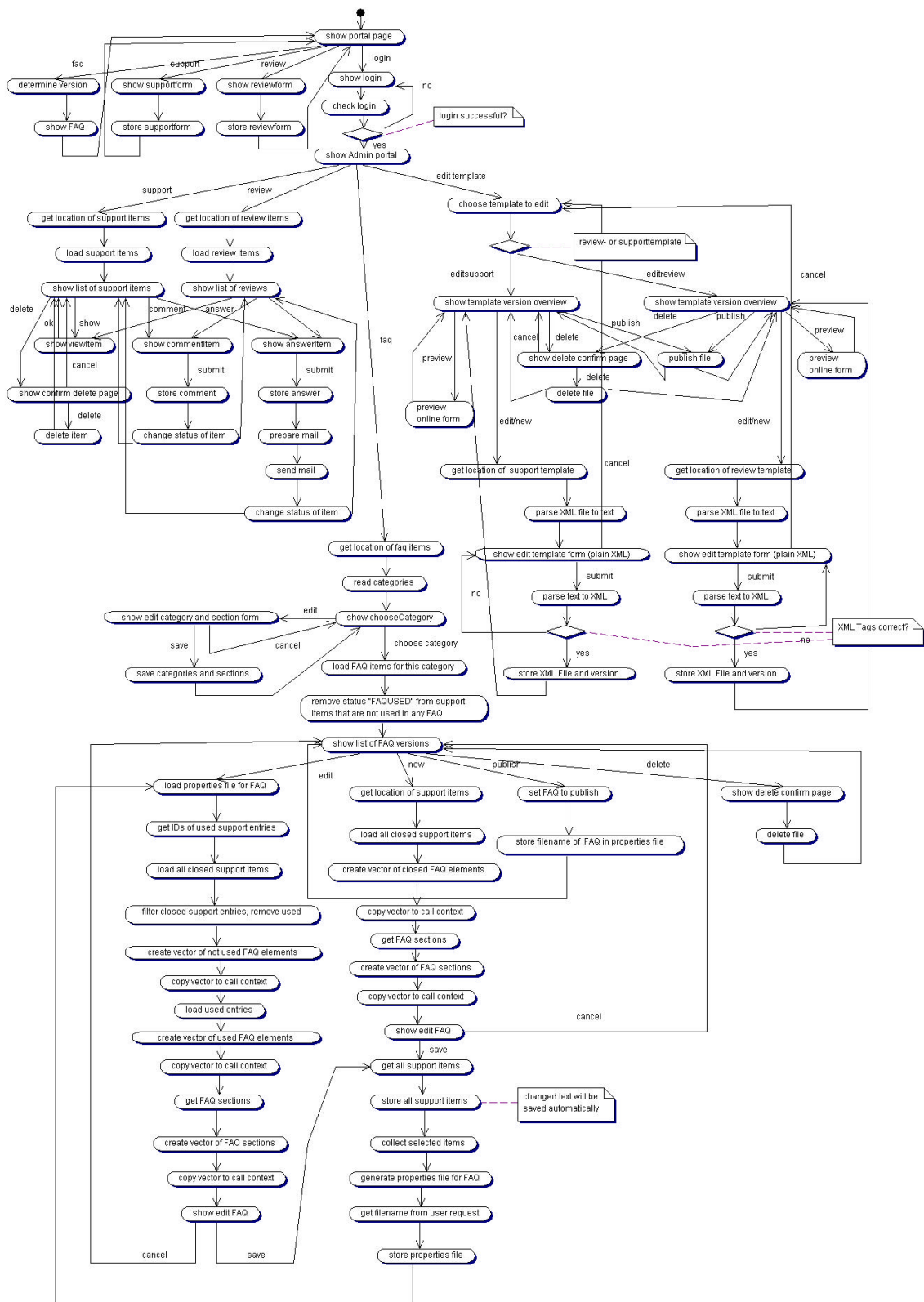


Abbildung 4.15: Activity Chart des gesamten Dienstes

[\[Logout\]](#) [\[Support overview\]](#) [\[Review overview\]](#) [\[FAQ overview\]](#) [\[Edit templates\]](#)

FAQ Settings

Here you can edit the FAQ sections and categories.

All values must be separated by a comma and have to be written directly after each other.

FAQ Categories:

FAQ Sections:

Abbildung 4.16: Eingabemaske für die FAQ Kategorien und Sektionen

[\[Logout\]](#) [\[Support overview\]](#) [\[Review overview\]](#) [\[FAQ overview\]](#) [\[Edit templates\]](#)

Edit FAQ - PC Chair

Please edit the FAQ by selection or deselection of support entries. Also you can change the section an entry appears in. Alternatively you can enter a new support question and answer in the text fields at the bottom of this page. Before saving the FAQ you can choose if you want to write a new FAQ file or if you want to replace the existing one.

Already contained in the FAQ support questions:

Currently not used support entries:

Question:

Answer:

Section:

ID: 5

Use in FAQ?

Question:

Answer:

Section:

ID: 3

Use in FAQ?

Abbildung 4.17: Eingabemaske für die FAQ

[\[Logout\]](#) [\[Support overview\]](#) [\[Review overview\]](#) [\[FAQ overview\]](#) [\[Edit templates\]](#) [\[Download CSV\]](#) [\[Enter Fax Review\]](#)

Support Overview

Sort by: ID, [Date](#), [Name](#), [State](#).

ID	Date	Name	E-Mail	State	Question	Answer	Action
1	Tue Jul 06 14:35:33 CEST 2004	Dennis Sassmannshausen	test@test.de	closed	It is not possible to register a new user. A Tomcat 500 Error keeps appearing all the time with different Text.	A Tomcat 500 error ist not quite uncommon and can have various reasons. Please contact one employee of our support staff if this error keeps occuring.	View Print Edit Delete
2	Tue Jul 06 14:35:33 CEST 2004	Heinz Mustermann	muster@muster.de	closed	It is not possible for me to submit a new article.	Submitting a new article is not allowed to all members of the OCS service. It may even depend of the state the evaluation of papers has currently reached. If the submission deadline is not yet passed it may be that you do not have the appropriate rights to submit a new article.	View Print Edit Delete
3	Tue Jul 06 14:35:33 CEST 2004	Stefan Schmidt	schmidt@meier.de	closed	What do the different colours of the bidding matrix mean?	The colours represent the state of the review the paper is currently in. It also can represent a once planned and not yet submitted delegation. Please see the colour code explanation right above the bidding matrix table.	View Print Edit Delete
4	Tue Jul 06 14:31:00 CEST 2004	Peter Smith	smith@smith.de	submitted	The Service was unavailable yesterday evening!		View Print Edit Delete
5	Tue Jul 06 14:35:33 CEST 2004	Henry Richards	richards@udo.edu	closed	Is it possible to change the background colour of the service?	Yes. You can customize the service appearance by using the setup feature located in the left navigation bar.	View Print Edit Delete
6	Tue Jul 06 14:35:33 CEST 2004	Jim Perry	jim@perry.com	analyzed	It takes a very long time to generate the bidding matrix table. Is there any way of speeding this up?		View Print Edit Close Delete

Abbildung 4.18: Übersicht der eingegangenen Supportanfragen

Show FAQ - PC Chair

Here you find frequently asked questions and the suitable answers about the main topics of the Online Conference Service.

Topics of the FAQ:

[Bidding](#)

[Article](#)

[Setup](#)

[Delegation](#)

[Article Submissions](#)

[Article Abstract Submission](#)

Bidding

Question: What do the different colours of the bidding matrix mean?

The colours represent the state of the review the paper is currently in. It also can represent a once

Answer: *planned and not yet submitted delegation. Please see the colour code explanation right above the bidding matrix table.*

[home](#)

Article

Question: It is not possible for me to submit a new article.

Submitting a new article is not allowed to all members of the OCS service. It may even depend of the

Answer: *state the evaluation of papers has currently reached. If the submission deadline is not yet passed it may be that you do not have the appropriate rights to submit a new article.*

Abbildung 4.19: Generierte FAQ

[\[Logout\]](#) [\[Support overview\]](#) [\[Review overview\]](#) [\[FAQ overview\]](#) [\[Edit templates\]](#)

Please choose an action for the chosen template

Xml template name
<input type="radio"/> feedback_support_001
<input checked="" type="radio"/> feedback_support_002
<input type="radio"/> feedback_support_003
<input type="button" value="activate"/>
<input type="button" value="edit"/>
<input type="button" value="preview"/>
<input type="button" value="delete"/>

Abbildung 4.20: Übersicht der abgelegten Support Templates

[\[Logout\]](#) [\[Support overview\]](#) [\[Review overview\]](#) [\[FAQ overview\]](#) [\[Edit templates\]](#)

Please edit the template source code

Content of the XML File:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE survey SYSTEM "survey.dtd">

<survey>
  <title>Online Support System for OCS</title>

  <page>
    <group>
      <input_question>
        <title>Name</title>
        <constraints>
          <mandatory_constraint/>
        </constraints>
      </input_question>

      <input_question>
        <title>Email</title>
        <other>other</other>
      </singleselect_question>
    </group>
  </page>
</survey>
```

Abbildung 4.21: Eingabemaske für die XML Template

4.3 Production Feature

Beschreibung der Aufgabe

Zum Abschluß einer Konferenz muss ein Dokument erstellt werden, das alle wichtigen Informationen einer Konferenz enthält. Dazu zählen alle Artikel der Konferenz, ein Inhaltsverzeichnis, sowie Listen aller PC Chairs, PC Member und der Sub-Reviewer. Um den Benutzer bei dem Zusammentragen dieser Informationen zu unterstützen, wurde das Production Feature realisiert. Es unterstützt den Benutzer Schritt für Schritt bei der Zusammenstellung aller relevanten Informationen für den Tagungsband einer Konferenz. Im ersten Schritt werden alle Artikel zusammengestellt, die in der finalen Version vorliegen. Diese sind in die Reihenfolge zu bringen, in der sie im Tagungsband der Konferenz vorkommen sollen. Ausserdem wird jedem Tagungsband eine eindeutige Nummer, die LNCS-Nummer, zugewiesen. Nach dem Ordnen der Artikel muss der Benutzer das Inhaltsverzeichnis festlegen und ausfüllen. Nach diesem Schritt existieren alle relevanten Informationen, um sie in einer Textdatei zusammenzufassen. Diese Textdatei bekommt die LNCS-Nummer als Dateinamen zugewiesen. Diese Datei enthält das Inhaltsverzeichnis, sowie die Liste aller teilnehmenden Personen der Konferenz. Als letzter Schritt sind die zu berücksichtigenden Artikeldateien zu wählen. Nach der Auswahl wird alles in ein Zip-Archiv gepackt und zum Herunterladen angeboten. Somit erhält der Benutzer alle Informationen, die für den Tagungsband einer Konferenz benötigt werden.

Modellierungsphase

Bevor die Arbeit am neu zu entwickelnden Feature beginnen konnte, mussten einige Vorarbeiten geleistet werden. Im OCS existierten bisher keine Seitenangaben für Artikel. Diese sind notwendig für das Erstellen des Inhaltsverzeichnisses. Deshalb musste der "FinalArticleSubmission" Workflow modifiziert werden, sowie die "Article" Datenbankskripte. Die "Article" Datenbank wurde um die Spalte "pagenumber" erweitert. Somit kann zu jedem Artikel die Seitenanzahl angegeben werden. Da diese Angabe erforderlich für das Production Feature war, musste der Benutzer zur Eingabe gezwungen werden. Als Schnittstelle bot sich der "FinalArticleSubmission" Workflow an. In diesem Workflow wurde die zugehörige Webseite um ein Feld "pagenumber" erweitert. Dieses Feld muss vom Autor beim Einreichen der endgültigen Version eines Artikels ausgefüllt werden. Durch beide Modifizierungen wurde sichergestellt, dass alle Informationen für ein Tagungsband im OCS vorhanden sind. Ausserdem wurde der neue Workflow als ein neues Feature "Production" in den OCS eingebunden.

Zuerst wurde das Use Case Diagramm (siehe Abbildung 4.22) modelliert. Es wurden die Schritte, die der Benutzer während des Workflows durchlaufen muss, nachgebildet. Nachdem der Benutzer das Feature "Production" ausgewählt hat, ist der erste zu bewältigende Schritt die Eingabe der korrekten LNCS-Nummer. Auf der selben Seite kann ebenfalls die

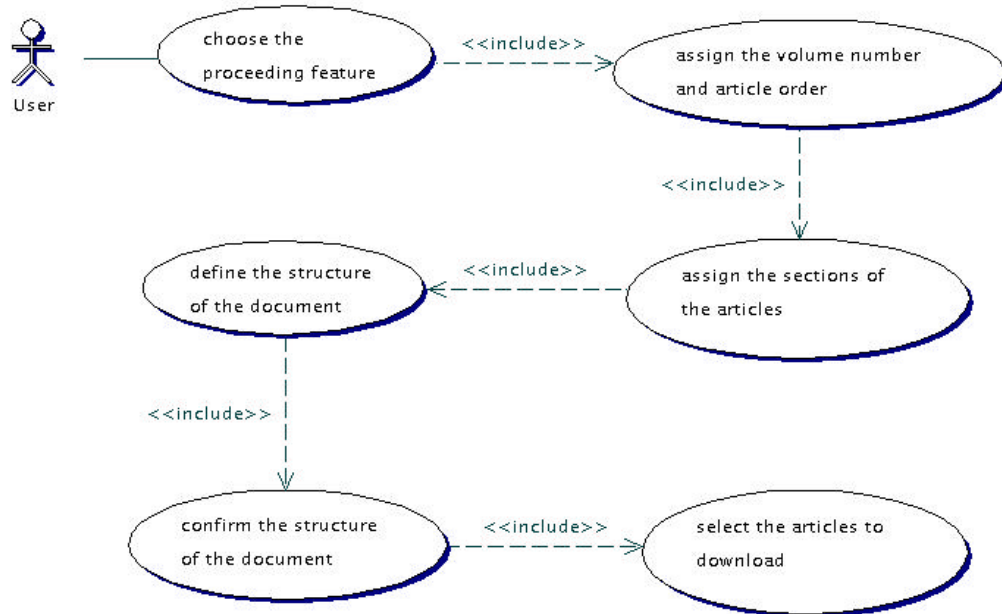


Abbildung 4.22: Das Use Case des Production Features

Reihenfolge der Artikel für das Tagungsband den Wünschen entsprechend geändert werden. Als nächstes beginnt die Zusammenstellung des Inhaltsverzeichnisses. Der Benutzer muss die Artikel den entsprechenden Kapiteln zuordnen. Die im letzten Schritt festgelegte Reihenfolge der Artikel dient bei dieser Zusammenstellung als Hilfestellung, kann aber trotzdem geändert werden. Nach dem Zuordnen der Artikel wird das Inhaltsverzeichnis erstellt und somit die endgültige Struktur des Tagungsbandes festgelegt. Das Resultat wird dem Benutzer zur Bestätigung vorgelegt, indem das Inhaltsverzeichnis in einer Textdatei zusammengefasst wird. Zuletzt müssen die richtigen Dateien der entsprechenden Artikel zum Herunterladen ausgewählt werden. Nach diesen Schritten kann ein Tagungsband anhand der vorliegenden Informationen erstellt werden.

Die Feinheiten, die auf der Implementierungsebene berücksichtigt werden mussten, werden im Aktivitätsdiagramm (siehe Abbildung 4.23) verdeutlicht. Bevor die erste Seite des Workflows angezeigt werden kann, müssen einige Informationen und Objekte zusammengestellt werden. Zuerst werden alle Artikel ausgefiltert, die nicht im "final version" Zustand vorliegen. Nur die Artikel, die in der endgültigen Version vorliegen, sind für den Tagungsband zu nutzen. Aus der Liste dieser Artikel wird eine Map erstellt, die alle relevanten Informationen der Artikel enthält. Zu diesen Daten zählen die Titel der Artikel, die Autoren, sowie die Anzahl der Seiten. Die "ArticleMap" wird im kompletten Workflow genutzt, um die Artikeldaten zur Verfügung zu haben. Die Aktivität "Check CVS Repository for ZipFile" wurde nach einer Änderung des Workflows zwischen

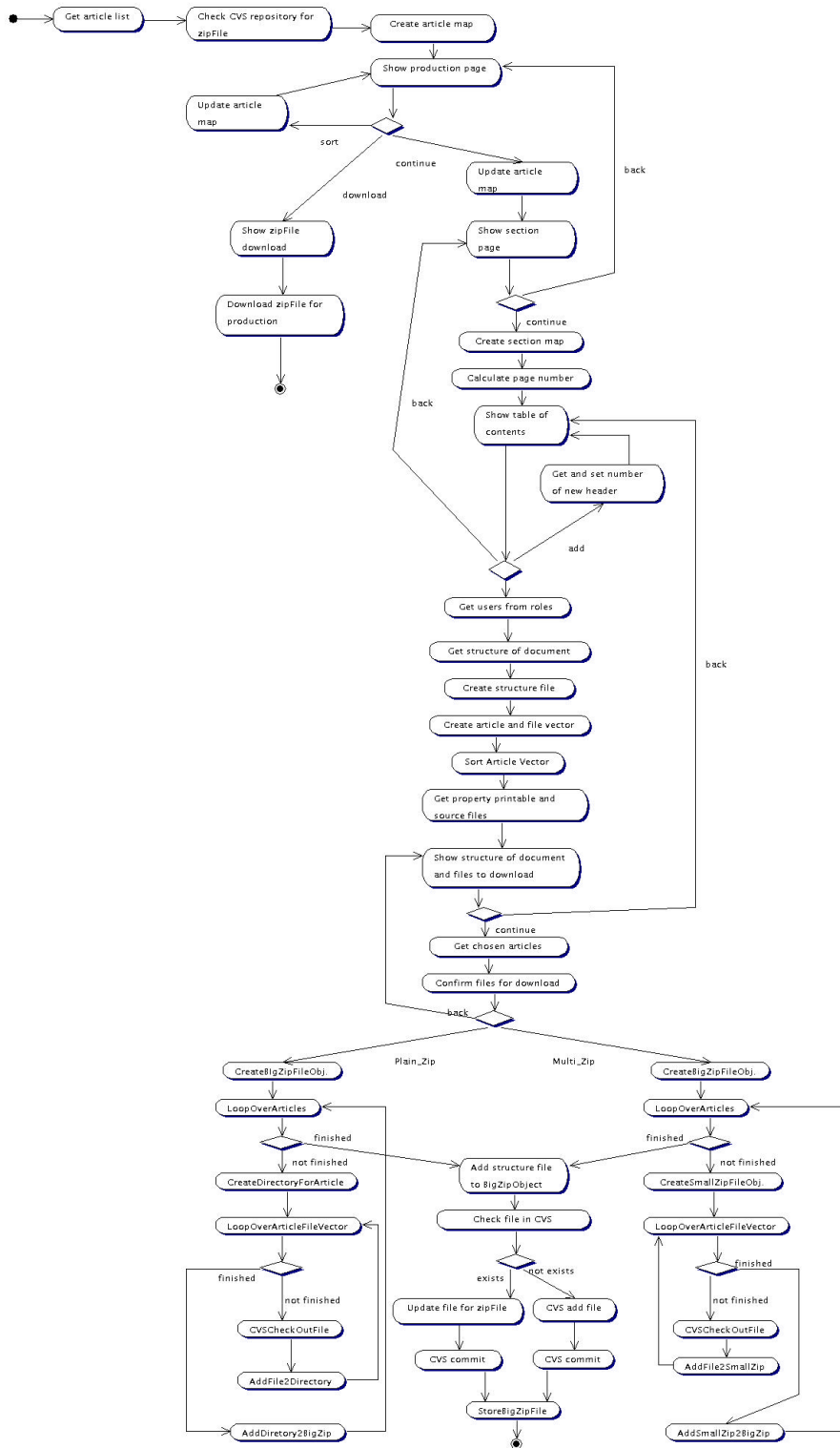


Abbildung 4.23: Das Activity Chart des Production Features

die beiden zuletzt beschriebenen Aktivitäten eingefügt. Jedes Zip-Archiv, das mit dem Production Feature erstellt wird, wird in das CVS Repository eingchecked. Somit hat der Benutzer die Möglichkeit seine Arbeit aus dem Repository auszuchecken, falls das Herunterladen des Zip-Archivs fehlschlägt.

Nach diesen Schritten verfügt das System über alle benötigten Informationen, um die erste Webseite des Workflows anzuzeigen. Auf dieser Seite wird die LNCS-Nummer, sowie die Reihenfolge der Artikel festgelegt. Wurden Änderungen an der Reihenfolge der Artikel vorgenommen, kann diese mit Hilfe des "Sort" Buttons sortiert werden. Die Sortierung erfolgt durch das Neuberechnen der "ArticleMap". Sollten Dateien im CVS Repository vorhanden sein, so können diese heruntergeladen werden, womit der Workflow beendet wird.

Nach dem Neuberechnen der "ArticleMap" werden die Artikel in der neuen Reihenfolge angezeigt. Diese sind im nächsten Schritt den gewünschten Kapiteln zuzuordnen. Um diese Zuordnung festzuhalten, werden "SectionMaps" erstellt. Anschließend können, da die Reihenfolge der Artikel feststeht, die Seitenzahlen für das Inhaltsverzeichnis berechnet werden. Aus diesen Daten wird dem Benutzer ein Inhaltsverzeichnis angeboten. In diesem sind lediglich die Kapitelüberschriften zu vergeben, da die Artikeltitle bereits eingetragen sind. Sollten weitere Felder für die Überschriften gebraucht werden, so gibt es die Möglichkeit deren Anzahl zu erhöhen.

Zu diesem Zeitpunkt verfügt das System über alle Informationen, um einen Tagungsband zu erstellen. Als nächstes werden Namen aus verschiedenen Rollen des OCS ausgelesen. In diesem Workflow werden Listen mit Namen der PC Member, PC Chairs und Sub-Reviewer generiert. Danach wird die Struktur des Dokuments mit den neu eingetragenen Überschriften gesichert. Daraus wird eine Strukturdatei im Textformat erstellt. Diese enthält das Inhaltsverzeichnis mit der Angabe der Autoren und Artikeltitle, sowie der zugehörigen Seitenzahlen. Ausserdem enthält diese Textdatei die oben erwähnte Liste der Teilnehmer der Konferenz. Um auf der folgenden Webseite die Dateien der einzelnen Artikel auswählen zu können, wird ein Vektor mit den Artikeln und Dateien erstellt. Wegen der möglicherweise geänderten Reihenfolge der Artikel, die nicht wie am Anfang des Workflows nach der Artikel ID sortiert sind, wird der eben erzeugte Vektor sortiert. Somit werden die Artikeldateien zum Herunterladen in der Reihenfolge angezeigt, in der sie im Inhaltsverzeichnis vorkommen.

Auf der folgenden Seite wird die oben erstellte Strukturdatei angezeigt, sowie die geordneten Artikeldateien zum Herunterladen angeboten. Diese Dateien sind in zwei Spalten eingeteilt. In der ersten Spalte werden die druckbaren Dateien und in der zweiten Spalte die Quelldateien angezeigt. Zu jedem Artikel darf jeweils eine Datei aus der Kategorie heruntergeladen werden. Sollten alle Dateien vom Benutzer gebraucht werden, so besteht die Möglichkeit mit Hilfe des Buttons "Download All" alle Artikeldateien herunterzuladen.

Die folgenden Schritte des Workflows ähneln denen des Kapitels 3.3.3. Abhängig, ob ein "multi" oder "plain" Zip-Archiv erstellt werden soll, wird der zugehörige Workflow durchlaufen. Der Unterschied zum "File download workflow" ist das Einchecken des Archivs in das CVS Repository. Vor dem Einchecken wird geprüft, ob ein gleichnamiges

Archiv im Repository vorhanden ist. Ist das der Fall, so wird die Datei "committed", ansonsten vorher ins Repository eingefügt. Am Ende des Workflows wird das Zip-Archiv dem Benutzer zum Herunterladen angeboten.

Implementierungsphase

Für die Umsetzung der Modellierungsphase mussten fünf Hauptwebseiten erstellt werden, sowie zahlreiche neue SIBs. Um für die erste Webseite "showproduction.wml" alle benötigten Informationen zu sammeln, musste eine "ArticleMap" erstellt werden. Dazu wurde der SIB "CreateArticleMap" implementiert. Mit Hilfe dieses SIBs werden alle Artikel in einer Map zusammengefasst, wobei die ID der Mapeinträge der Position des Artikels im Tagungsband entspricht. Auf der ersten Webseite befindet sich eine Tabelle aller Artikel. Sie enthält editierbare Artikelpositionen, Artikeltitle, Autoren und Seitenanzahlen. Um die Fehler, die ein Benutzer begehen kann, aufzufangen, wurde eine Fehlerseite "showproduction_error.wml" erstellt. Je nach Fehler bietet sie Vorschläge für ihre Beseitigung an. Fehler treten auf, wenn die LNCS-Nummer nicht vierstellig angegeben wurde oder die Artikel nicht korrekt geordnet worden sind. Damit die Artikel richtig geordnet werden können, dürfen keine Lücken zwischen den Positionsangaben vorliegen oder gleiche Positionsnummern vergeben werden. Die Sortierung der Artikel gemäß erster Webseite erfolgt mit dem "UpdateArticleMap" SIB. Dieser SIB sortiert die Artikel in der Map. Somit verfügt das System über die nach Benutzerwunsch geordnete Liste von Artikeln.

Diese Liste wird auf der nächsten Webseite "showproduction_section.wml" ausgegeben. Dort kann jedem Artikel ein Kapitel zugeordnet werden. Da zu diesem Zeitpunkt keine Struktur für den Tagungsband existiert, werden für "n" Artikel "n" Kapitel angeboten. Nach der Zuordnung der Artikel wurden SIBs implementiert, die die Zuordnung abspeichern. Der SIB "CreateSectionMap" beinhaltet eine Map, deren Key der jeweiligen Section entspricht. Dort werden die ArticleIDs abgelegt und somit die Zuordnungen zu den jeweiligen Kapiteln festgehalten. Nebenbei wird die "ArticleMap" aktualisiert, da sich durch die Zuordnungen zu den Kapiteln die Artikelpositionen geändert haben könnten.

Zu diesem Zeitpunkt steht die Struktur des Dokuments für den Tagungsband fest. Um das Inhaltsverzeichnis anzeigen zu können, werden die Seitenanzahlen mit dem SIB "CalculatePageNumber" ermittelt. Die dazu benötigten Informationen über die Seitenanzahlen werden aus der "ArticleMap" und der "SectionMap" bezogen. Schließlich wird das Inhaltsverzeichnis angezeigt. Es gibt Möglichkeiten, eine Überschrift für das gesamte Dokument und für die einzelnen Kapitel zu vergeben. Die Artikeltitle können ebenfalls modifiziert werden. Weil die Seitenanzahlen berechnet worden sind, werden sie auch angezeigt. Für jede Überschrift werden zwei Textfelder angeboten. Sollten diese nicht ausreichen, kann durch den "Add Header" Button ein weiteres Textfeld hinzugefügt und ausgefüllt werden. Um die bereits getätigten Eingaben des Benutzers vor dem Verlust zu sichern wird der "CreateVector" SIB verwendet. Dieser SIB speichert die Inhalte aller Textfelder. Nebenbei erzeugt er einen Vector mit Integerwerten. Dieser

Vektor wird benötigt, weil Velocity keine zahlenbasierten For-Next-Schleifen anbietet. Um die benötigte Anzahl von Textfeldern anzubieten, bietet sich ein Lauf mit Hilfe dieser Schleife an.

Nach dem Füllen des Inhaltsverzeichnisses muss eine Textdatei mit den Listen der PC Chairs, der PC Member und der Sub-Reviewer, sowie mit dem Inhaltsverzeichnis erstellt werden. Um die Listen der jeweiligen Teilnehmer zu erzeugen, wurde der SIB "GetUserListFromRoles" implementiert. In diesem SIB ist die gebrauchte Rolle als Parameter einstellbar, deshalb wurde er für alle drei Listen benutzt. Sollte eine dieser Rollen nicht existieren, so wird keine Liste erzeugt. Danach erzeugt der SIB "GetStructureOfDoc" die benötigte Textdatei. Diese Datei wird unter dem Namen der LNCS-Nummer im angegebenen "DefaultStructureFileDirectory" zwischengespeichert.

"showproduction_tableofcontents.wml" erzeugt die Webseite, auf welcher der Inhalt der Textdatei und die artikelzugehörigen Dateien angezeigt werden. Der Inhalt der Textdatei wird mit dem SIB "ReadFileToString" ausgelesen und in einer Textarea angezeigt. Um die Artikel zum Herunterladen in der Reihenfolge zu präsentieren, wie sie im Inhaltsverzeichnis vorkommen, wurde der SIB "SortArticleVector" geschrieben. Dieser SIB ordnet die Artikelobjekte in einen Vektor ein. Die Reihenfolge der Artikel wird aus der "ArticleMap" bezogen, da dort die Positionen im Inhaltsverzeichnis gespeichert sind. Der Versuch die im Call Context vorhandene "ArticleList" zu ordnen war nicht möglich, weil sie die Artikel automatisch nach der ID sortiert. Deshalb wird der Vektor an die Webseite übergeben. Die Dateien der Artikel sind in zwei Spalten eingeordnet. Die erste Spalte enthält alle druckbaren, die zweite Spalte alle Quelldateien eines Artikels. Um dies zu realisieren wurden die Endungen der "printable" und "source" Dateien aus der "Setup.properties" Datei ausgelesen und die Dateien entsprechend eingeordnet. Dazu wurden die beiden SIBs "GetSetupPropertySourceFiles" und "GetSetupPropertyPrintableFiles" implementiert. Durch die Benutzung von Checkboxen ist das Auswählen einer Datei aus jeder Spalte gesichert. Somit wird zu jedem Artikel eine druckbare Version im späteren Zip-Archiv vorhanden sein. Sollte der Benutzer der Ansicht sein, alle Dateien eines Artikels zu benötigen, wurde der Button "Download All" zur Webseite hinzugefügt. Durch seine Nutzung werden alle Dateien eines Artikels in das Zip-Archiv gepackt. Ausserdem stehen zwei Möglichkeiten zur Verfügung, Dateien zu in ein Archiv zu packen. Entweder in ein "multi-zip" oder in ein "plain-zip" Archiv. Diese beiden Möglichkeiten wurden ausführlich im Kapitel 3.3.3 beschrieben.

"showproduction_confirmdownload" wird als letzte Seite des Workflows angezeigt. Diese Seite zeigt alle zum Herunterladen ausgewählten Dateien der Artikel an. Um diese gewählten Artikeldateien auszulesen und sie auf dieser Seite zu präsentieren, mussten einige SIBs vor dieser Website im Vergleich zum Workflow des Kapitels 3.3.3 geringfügig modifiziert werden. Die Modifikationen sind durch das Einsetzen der Checkboxen notwendig geworden. Deshalb wurde der SIB "LoopOverInput" in den SIB "LoopOverMergedInput" modifiziert. Um diesen SIB einzusetzen wurden die Eingaben von dem SIB "MergeInput" aufbereitet. Dieser SIB durchläuft das Array der Parameter der Webseite "showproduction_tableofcontents.wml" und erstellt einen geeigneten Vektor für den "LoopOverMergedInput" SIB. Durch diese Maßnahme konnte fast der gesamte

Workflow für das Erstellen der Zip-Archive übernommen werden.

Nach der Bestätigung des Downloads werden die gewählten Dateien aus dem CVS Repository ausgecheckt und entsprechend der gewählten Zip-Methode gepackt. Wurden alle Dateien der Artikel zum Download angeboten, findet keine Umbenennung der Dateien statt. Sollten jeweils eine "printable" und eine "source" Datei eines Artikels gewählt worden sein, so werden die Dateien umbenannt. Der Name der Dateien entspricht der vierstelligen LNCS-Nummer und der Anfangsseite im Tagungsband. Dadurch ist eine schnelle Einordnung der Dateien im Tagungsband möglich.

Nach dem Erstellen des Zip-Archivs muss die Datei ins CVS Repository eingechekkt werden. Zuerst wird überprüft, ob eine gleichnamige Datei sich bereits im CVS Repository befindet. In diesem Fall muss das File lediglich "comitted" werden. Ansonsten muss entsprechend den CVS Mechanismen die Datei "added" und anschliessend "comitted" werden. Erst nach diesen Schritten wird das Zip-Archiv, das den Dateinamen der LNCS-Nummer trägt, an den Browser zum Download geschickt. Durch dieses Vorgehen kann die Arbeit eines Benutzers, sollte der Download scheitern, gesichert werden.

Existieren Dateien im CVS Repository des Tagungsbandes, so können diese auf der ersten Webseite "showproduction.wml" des Workflows heruntergeladen werden. In diesem Fall existiert ein zusätzlicher Button "Download ZipFile" auf der ersten Seite. Dieser zeigt die Webseite "showzipfiledownload.wml" an. Auf dieser Seite werden alle existierenden Dateien des CVS Repositories für das Tagungsband angezeigt. Der Benutzer kann eine Datei zum Herunterladen auswählen und muss den gesamten Workflow nicht nochmals durchlaufen. Dort wird jeweils die aktuellste Datei des CVS Repositories zum Herunterladen angeboten.

Testphase

Wegen der Testphase konnten zahlreiche Verbesserungen am Workflow vorgenommen werden und viele Designfehler beseitigt werden. Durch die Testphase wurden die Fehlerseiten des Workflows verbessert. Auf der ersten Webseite "showproduction.wml" können dem Benutzer zwei verschiedene Fehler unterlaufen. Entweder durch die Angabe einer falschen LNCS-Nummer oder durch einen Fehler in der Angabe der Positionen der Artikel. Zu jedem dieser Fehler wurde eine Fehlerseite mit Lösungsvorschlägen programmiert. Somit ist die Fehlerbehebung für den Benutzer komfortabler gestaltet worden.

Die grössten Designänderungen wurden auf der vierten Webseite "showproduction_download.wml" durchgeführt. Zu einem Tagungsband gehört eine Liste aller Teilnehmer, aufgelistet nach ihren Rollen in der Konferenz. Deswegen wurden die Listen mit den PC Chairs, PC Members und Sub-Reviewern hinzugefügt, die nach der ersten Design- und Implementierungsphase nicht vorgesehen waren. Desweiteren fiel die Länge der angezeigten LNCS-Textdatei negativ im Design der vierten Seite auf. Bei Konferenzen mit vielen Artikeln würde die Anzeige der Datei zum langen Scrollen der Seite führen, deswegen ist sie in der Endversion in ein festes, aber scrollbares Fenster eingebettet.

Eine weitere Änderung war die Reihenfolge der Artikel, die auf der vierten Webseite zum Herunterladen angeboten werden. Zuerst waren die Artikel nach ihrer ID geordnet. Dieses Design hätte zur Verwirrung des Benutzers führen können, deshalb wurden die Artikel in der Reihenfolge des Inhaltsverzeichnisses geordnet.

Ein weiterer, schwerwiegender Fehler war beim Einchecken der Dateien aufgetreten. In der Testphase wurden zufällig gleichnamige Dateien eingecheckt, was zum Absturz des Workflows führte. In der Endversion wird zuerst überprüft, ob eine gleichnamige Datei im Repository existiert. In diesem Fall wird die Datei "committed", ansonsten vorher ins Repository eingefügt.

Weitere Kleinigkeiten, wie die Anpassung aller Tabellenstrukturen auf den Webseiten und die Anpassung der Reihenfolge der Buttons fielen ebenfalls erst in der Testphase negativ auf. Desweiteren wird die "plain" Zip-Methode der "multi" Zip-Methode vorgezogen. Diese Methode wurde als vorgewählt festgelegt, weil durch die "multi" Zip-Methode ungewollte Effekte entstehen können. Sollte es sich bei den "source" Dateien der Artikel um "Zip"-Dateien handeln, so können die heruntergeladenen Hauptarchive durch die entpackten Quelldateien überschrieben werden.

Nach der Testphase und den gefundenen Bugs wurden keine weiteren Fehler oder Designschwächen gefunden, womit ein komfortabler Workflow für das Erstellen eines Tagungsbandes entstand.

4.4 Service Status Report Feature

Beschreibung der Aufgabe

Aufgabe war es, dem OCS ein neues Feature hinzuzufügen, das eine Übersicht über den momentanen Status der Konferenz gibt. Der *Service Status Report* sollte Auskunft über die eingereichten Artikel sowie ihren jeweiligen Status geben. Darüber hinaus sollte der Zustand des Delegationsvorgangs deutlich gemacht werden und bereits eingegangene Reports und die Bewertungen der Reviewer in einer Tabelle aufgeschlüsselt werden. Der Report sollte als HTML Datei und als CSV¹ Datei zur Verfügung stehen und von dem Benutzer herunter geladen werden können. Wahlweise sollte der Benutzer die Möglichkeit haben, die CSV Datei in ein CVS Repository zu speichern.

Eine weitere Anforderung war, dass dieser Status Report automatisch durch einen Thread in einem im Setup einstellbaren Zeitintervall generiert und an die Mitglieder des Programm Komitees verschickt wird, sofern diese es wünschen. Der automatisch erstellte Report sollte immer in das CVS Archiv gespeichert werden, auf das auch die Benutzer Zugriff haben. Die einzelnen Funktionen (generieren, speichern, gespeicherte Reports einsehen, einen Report per Email bekommen) sollten über Features den Rollen gestattet, bzw. verboten werden können.

¹comma separated values

Modellierungsphase

Zu Beginn der Designphase wurden die Use Case Diagramme erstellt. Abbildung 4.24 zeigt die Anwendungsfälle, die mit der interaktiven Benutzung des Status Reports in Verbindung stehen. Als legitimer Benutzer wird der PC Chair dargestellt. Dies könnte

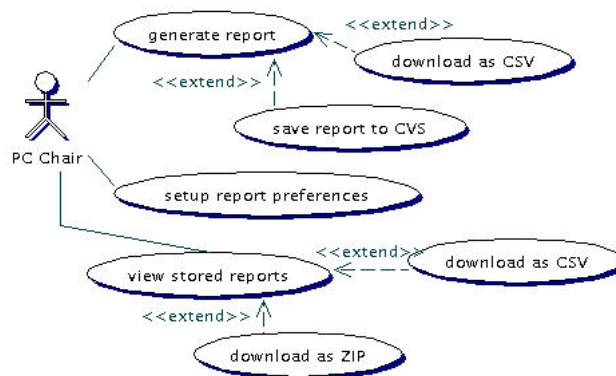


Abbildung 4.24: Use Case des interaktiven Arbeitsablaufs

aber genauso gut jeder andere Benutzer sein, dem die entsprechenden Features zugeteilt wurden. Ebenso wäre es denkbar, dass bestimmte User nur Teile dieser Anwendungsfälle ausführen dürfen. Von dieser Tatsache wird in den Abbildungen abstrahiert. Abbildung 4.25 stellt die Operationen, die von dem Thread ohne Eingreifen eines Benutzers vollzogen werden, dar.

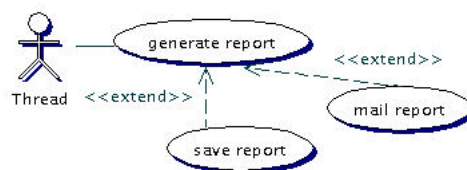


Abbildung 4.25: Use Case des automatischen Arbeitsablaufs

Nach den Use Case Diagrammen wurden die Aktivitätsdiagramme erstellt, die den genaueren Ablauf modellieren. Hier gilt es wieder den interaktiven Arbeitslauf vom automatischen Ablauf zu unterscheiden. Unsere Idee war es, beim Generieren des Status Reports direkt die CSV Datei in den Call Context zu schreiben, weil dies der einfachste Weg war, die Daten in diese Form zu bekommen.

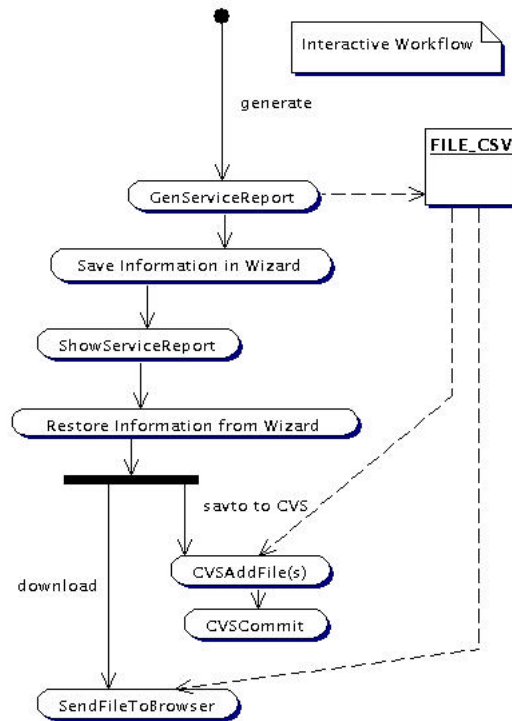


Abbildung 4.26: Activity Chart des interaktiven Arbeitsablaufs

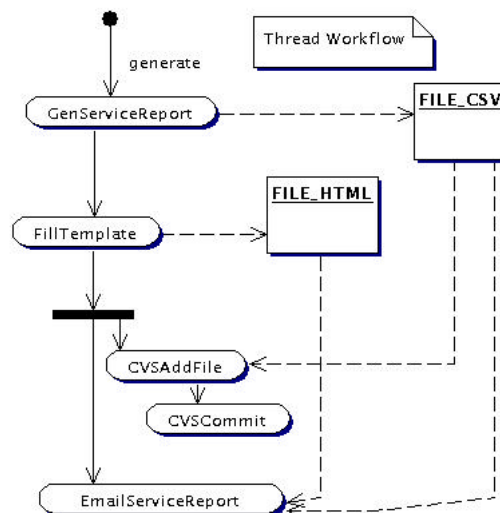


Abbildung 4.27: Activity Chart des automatischen Arbeitsablaufs

Im interaktiven Arbeitsablauf (Abbildung 4.26) kann mit Hilfe eines Wizards die Information zur CSV Datei auch noch nach dem Anzeigen des Reports abgerufen werden und somit ist es möglich, die Datei direkt herunterzuladen oder in das CVS zu speichern.

Der automatische Arbeitsablauf (Abbildung 4.27) benötigt auf Grund der fehlenden Anzeigeseite keinen Wizard und kann direkt alle nötigen Informationen durch den Call Context zwischen den SIBs austauschen.

Implementierungsphase

Als ersten Schritt der Implementierung legten wir den Grundstein, indem wir das Navigationsmenü um einen neuen Unterpunkt für den Status Report erweiterten. Darauf folgte das Definieren der Features, die dem Report zugeordnet werden sollten. Wie in Abbildung 4.28 zu sehen, können die einzelnen Aspekte des Features granular spezifiziert werden. Die Voraussetzung zur Nutzung des Reports stellt das Feature *F-REPORT-01* dar. Ohne dieses Feature bekäme der Benutzer den Menüpunkt für den Status Report nicht zu Gesicht. Desweiteren kann bestimmt werden, welche Rolle den Report generieren darf (*F-REPORT-02*), was Voraussetzung für das Speichern des Reports (*F-REPORT-02.save*) ist. Ferner sind das Abonnieren der Report Email (*F-REPORT-03*) und das Anzeigen der bereits gespeicherten Reports (*F-REPORT-04*) wählbar.

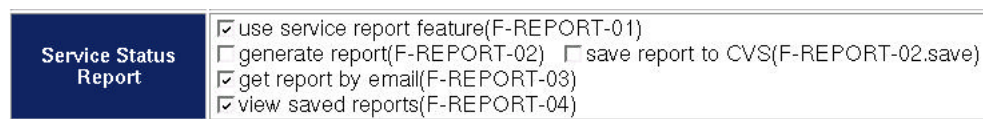


Abbildung 4.28: Zum Report gehörende Features

Der nächste Schritt war, den Report zu erstellen und anzuzeigen. Um an die benötigten Informationen zu gelangen, stellte das OCS bereits einige Administratoren Objekte (etwa den Artikel Administrator oder den User Administrator) zur Verfügung. Durch entsprechende Anfragen konnten die benötigten Informationen gesammelt werden. Diese wurden in Vektoren gegliedert und dann der anzeigenden WML Seite bereitgestellt, welche Velocity Variablen und Anweisungen zur Darstellung der Inhalte der Vektoren enthielt. Wie schon im Design Abschnitt beschrieben, wurden die Vektoren zum Zeitpunkt der Generierung durch Kommata getrennt in eine CSV Datei geschrieben, welche im Call Context hinterlegt wurde (vgl. Abbildung 4.26) und über die anzeigende HTML Seite verlinkt ist. Mit Hilfe des Wizards, der Call Context Daten über eine anzeigende Seite rettet, konnte die Referenz auf die hinterlegte Datei wieder hergestellt werden. Je nach Aktion, die der Benutzer wählt, wird diese Referenz benutzt, um die CSV Datei an den Browser zu senden, oder in das CVS zu speichern.

Um das Speichern im CVS, was bisher in erster Linie für Artikel vorgesehen war, möglich zu machen, wurden einige der CVS-SIBs angepasst, so dass sie auf einem bestimmten Verzeichnis für die Reports innerhalb des CVS Repositories arbeiten konnten.

Da die einzelnen Komponenten des Reports wählbar sein sollten und ein Thread mit einstellbarem Intervall mit dem Arbeitsablauf verbunden werden sollte, wurde das Setup des Dienstes um einen Unterpunkt für den Status Report erweitert. Der Benutzer (mit Setup

Configuration of the Reporting Feature

This page lets you set up the automatic report generation.

You can specify how often a report should be generated automatically by the service. Furthermore, you can select the information, which should be part of the report.

Make sure to select **generate report** if you want automatic generation of reports.

General Settings	
<input checked="" type="checkbox"/> generate report	every 8 hours
Information included in Report	
<input checked="" type="checkbox"/> article information	
<input checked="" type="checkbox"/> reports information	
<input checked="" type="checkbox"/> delegation status	
<input checked="" type="checkbox"/> responsibles information	
Style of Report in Emails	
<input checked="" type="checkbox"/> attach as HTML	
<input checked="" type="checkbox"/> attach as CSV	
save	cancel

Abbildung 4.29: Setup des Reports

Berechtigung) hat dort die Möglichkeit, das Intervall der automatischen Generierung, sowie die Generierung selbst, einzustellen und die einzelnen Informationen, die im Report enthalten sein sollen, zu wählen (Abbildung 4.29). Er kann auch entscheiden, ob der Report als HTML Datei oder als CSV Datei (oder auch beides) an die zu verschickenden Emails angehängt werden soll. Die einzelnen Optionen werden auf *Setup Properties* des Dienstes abgebildet, so dass die Einstellungen global und dauerhaft gelten.

Jeder Benutzer, der das Report Feature benutzen darf, bekommt Zugang zu einer Service Report Seite (Abbildung 4.30), auf der er den Report als Email abonnieren, gespeicherte Reports einsehen und den Status Report erzeugen kann, sofern die entsprechenden Features dem Benutzer zugewiesen sind. Um das Abonnement der Report Email für einen Benutzer speichern zu können, mussten wir der Datenbanktabelle *user* ein neues Feld hinzufügen, das den Status des Abonnements speichert. Zusätzlich wurde der User Administrator um eine set- und eine get-Methode für dieses Feld erweitert. Das Generieren des Status Reports erzeugt eine HTML Seite, die entsprechend den Einstellungen des Setups (Abbildung 4.29) die gewählten Komponenten enthält. Die Informationen zu verantwortlichen Personen des Dienstes umfasst eine Auflistung der Administratoren, sowie der PC Chairs (Abbildung 4.31). Bei eingeschalteten Artikelinformationen wird eine Tabelle mit den Artikeln und ihren Autoren eingebunden (Abbildung 4.32), die für andere Tabellen als Legende dient, da diese sich nur über die Artikel ID auf die Artikel beziehen. Die

Service Report

Here you can specify your personal settings for the Service Report.

Personal Service Report Settings
<input checked="" type="checkbox"/> Get Service Report by email
<input type="button" value="SAVE"/>

Click the button below to generate a Service Report displaying the current state of the Service.

Click the button below to view all saved Service Reports.

Abbildung 4.30: User Seite zum Report

Responsibles

Administrators
Holger Willebrandt (holger.willebrandt@postamt.cs.uni-dortmund.de)

PC Chairs
Nobody Isperfect (holger.willebrandt@uni-dortmund.de)

Abbildung 4.31: Administratoren und PC Chairs

Tabelle, welche die Artikelanzahl pro Status anzeigt (Abbildung 4.33) wird ebenfalls zu den Artikelinformationen gezählt. Die Tabelle *Delegation Information* gibt Aufschluss

Article Information

ID	Title	Corresponding Author	Further Authors
001	Developing OCS with ABC	Zufall, Rainer (holger.willebrandt@cs.uni-dortmund.de)	Samantha Sample, Peter Smith
002	ABC Userguide	Zufall, Rainer (holger.willebrandt@cs.uni-dortmund.de)	Samantha Sample, Peter Smith
003	Worst Summer ever	Zufall, Rainer (holger.willebrandt@cs.uni-dortmund.de)	
004	I'm the BEST	Zufall, Rainer (holger.willebrandt@cs.uni-dortmund.de)	
005	L?ngste Theke der Welt	Theke, Anna (holger.willebrandt@ls5.cs.uni-dortmund.de)	
006	New Article Submission	Theke, Anna (holger.willebrandt@ls5.cs.uni-dortmund.de)	Holger Willebrandt, Rainer Zufall, Peter Smith
007	EUREKA OCS Conference Service	Isperfect, Nobody (holger.willebrandt@uni-dortmund.de)	

Abbildung 4.32: Legende der Artikel IDs

über den Status der Delegierungen (Abbildung 4.34). In der ersten Spalte werden die Benutzer aufgelistet, die die Rolle des Reviewers besitzen. Dahinter folgen Spalten mit der Anzahl der dem Benutzer delegierten Artikel, der Anzahl der Artikel, die der Benutzer für ein Review akzeptiert hat, sowie die Anzahl der abgelehnten Reviews. Die letzten drei Spalten listen die jeweils entsprechenden Artikel IDs der akzeptierten, abgelehnten und delegierten Reviews auf. Zusätzlich werden einige Artikel IDs farblich kodiert. Die Farbe hellgrün, in der Spalte *accepted IDs* besagt, dass dieser Artikel für ein Review akzeptiert wurde, aber der Report über das Review noch aussteht. Alle IDs dieser Spalte, die nicht farblich markiert sind, kennzeichnen bereits abgeschlossene Reviews, zu denen ein Report eingereicht wurde. In der Spalte *rejected IDs* werden alle Artikel IDs rot hervorgehoben, um abgelehnte Reviews auf den ersten Blick erkennen zu können. Alle IDs der Spalte *delegated IDs*, die in gelb hervorgehoben sind, repräsentieren Artikel, deren Review dem Benutzer delegiert wurde, welches er aber bis zum momentanen Zeitpunkt weder angenommen, noch abgelehnt hat. Diese Farbcodes entsprechen den Farbcodes, die bereits in der Delegation Matrix benutzt wurden. Somit finden sich Benutzer, die mit der Delegation Matrix vertraut sind, schneller in der Tabelle zurecht.

Die letzte noch fehlende Informationseinheit des Status Reports ist die Tabelle der Reports (Abbildung 4.35). Sie gibt Aufschluss über bereits eingereichte Reportes und die damit abgegebenen Bewertungen eines Artikels durch die Reviewer. Hier schlägt sich ein Teil der bereits in der Tabelle der Delegierungsübersicht (Abbildung 4.34) wiedergegebenen Informationen nieder. Es wird (mit der selben Farbcodierung) dargestellt, ob ein Benutzer sein Review abgelehnt hat (*rejected*), noch unentschieden ist (*undecided*), oder ob es akzeptiert wurde und nur noch der abschließende Report fehlt (*pending*). Liegt bereits

Number of Articles per State

State	Number of Articles
accepted	0
created	0
submitted	1
delegated	3
under_review	2
some_reports	0
all_reports	1
proposed_accepted	0
rejected	0
proposed_rejected	0
discussion	0
final_version	0
abstract	0
Total	7

Abbildung 4.33: Anzahl von Artikeln pro Status

Delegation Information

User	delegated	accepted	rejected	accepted (IDs)	rejected (IDs)	delegated (IDs)
Gotno I. Name (willebra@ls5.cs.uni-dortmund.de)	3	2	0	001 003	none	001 003 006
Holger Willebrandt (holger.willebrandt@postamt.cs.uni-dortmund.de)	1	1	0	004	none	004
Mr X (holger.willebrandt@udo.edu)	2	0	1	none	002	002 007

The red IDs in the column **delegated (IDs)** correspond to articles, that are neither accepted nor rejected from the contractor. For this IDs, a decision is awaited.

The yellow IDs in the column **accepted (IDs)** correspond to articles, whose report from the user is awaited.

The column **accepted** shows the number of articles that the contractor has accepted.

The column **rejected** shows the number of articles that the contractor has rejected.

Abbildung 4.34: Informationen zu delegierten Artikeln

ein Report über das Review eines Artikels vor, wird in der entsprechenden Tabellenspalte die vom Reviewer abgegebene Punktzahl (*score*) und sein Wissen über das Thema des begutachteten Artikels (*confidence*) angezeigt.

Scores and Confidences

ID	reviewers		
	Gotno I. Name	Holger Willebrandt	Mr X
001	0/3		
002			rejected
003	pending		
004		pending	
005			
006	undecided		
007			undecided

This table shows the Scores and the Confidences the Reviewers gave with their Reports.
(s/c): Score / Confidence

Abbildung 4.35: Informationen zu Reports

Auf der Anzeigeseite des Status Reports hat der Benutzer die Möglichkeit, diesen direkt als CSV Datei herunterzuladen, oder ihn in das CVS Repository zu speichern. Der Dateiname der erzeugten CSV Datei wird durch das Datum und die Uhrzeit bestimmt, so dass kein bereits gespeicherter Report überschrieben werden kann und sich die Reports im Dateisystem in chronologischer Reihenfolge auflisten lassen. Das Format des Dateinames ist YYYYMMDDhhmm, also vierstellige Jahreszahl, zweistelliger Monat, zweistelliger Tag, zweistellige Stunde und zweistellige Minute.

Von der Benutzerseite des Status Reports (Abbildung 4.30) aus lassen sich über den Link *View Saved Service Reports* die im CVS gespeicherten Dateien auflisten (Abbildung 4.36). Auf dieser Seite können die archivierten Reports einzeln oder eine Auswahl bestimmter Reports als ZIP Datei herunter geladen werden.

Implementierung der SIBs

Um das oben beschriebene Verhalten umzusetzen, wurden eine Reihe von SIBs geschrieben. Der zentrale SIB, *GenServiceReport* bereitet die notwendigen Daten für den Inhalt des Status Reports vor und schreibt sie in den Call Context. Darüberhinaus erstellt er die CSV Datei, die ebenfalls im Call Context gespeichert wird. Der SIB besitzt einen erforderlichen Parameter *csv_key*, der den Namen der Call Context Variablen beinhalten muss, unter der die CSV Datei abgelegt werden soll und vier optionale Parameter *include_articles*, *include_delegation*, *include_reports* und *include_userinfo*. Die optionalen Parameter dienen zum Überschreiben der Standardkonfiguration, dem Lesen der Einstellungen aus den *Setup Properties*, und bewirken, dass der SIB die über die Parameter angegebenen Call

List of Service Status Reports

This is a list of all available Service Status Reports

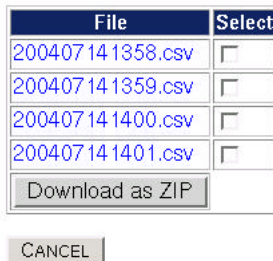


Abbildung 4.36: Im CVS gespeicherte Reports

Context Variablen liest, um zu erfahren, welche Informationen im Report enthalten sein sollen. Wenn keiner der optionalen Parameter gesetzt ist, wird standardmässig aus den *Setup Properties* gelesen. Ist einer der optionalen Parameter gesetzt, so werden alle überprüft und leere Parameter als *false* angesehen.

Der SIB speichert die anzuzeigenden Informationen in Vektoren im Call Context, die im folgenden beschrieben werden sollen:

- *Req_names* beinhaltet die Namen der möglichen Stati, die ein Artikel annehmen kann (submitted, created, rejected uvm.).
- *Req_numbers* beinhaltet die Anzahlen der Artikel pro Status, Die Reihenfolge entspricht der in *Req_names*.
- *total_of_all_article* beinhaltet die Gesamtanzahl aller Artikel.
- *resp_admin* beinhaltet alle Benutzer, die die Rolle *Administrator* haben.
- *resp_pcchair* beinhaltet alle Benutzer, die die Rolle *PC Chair* haben.
- *allarticle* enthält alle Artikel.
- *authors* enthält die Benutzer, die mindestens einen Artikel geschrieben und eingereicht haben.
- *users_with_del*: enthält die Benutzer, die mindestens an einer Delegation als Contractor beteiligt sind.
- *del_art_user*: Vektor, der die IDs von Artikeln, die einem Benutzer aus *users_with_del* delegiert sind, enthält.

- *del_id* enthält für jeden Benutzer aus *users_with_del* einen Vektor mit delegierten Artikeln. Die Reihenfolge der Vektoren entspricht der Reihenfolge in *users_with_del*.
- *ac_art_user*: Vektor, der die IDs von Artikeln, die einem Benutzer (aus *users_with_del*) delegiert sind und von diesem Benutzer akzeptiert wurden, enthält.
- *ac_art* enthält für jeden Benutzer aus *users_with_del* einen Vektor mit IDs von Artikeln, die der Benutzer akzeptiert hat.
- *user_rej_id*: Vektor, der die IDs von Artikeln, die den Benutzern (aus *users_with_del*) delegiert sind und von diesem abgelehnt wurden, enthält.
- *rej_art* enthält für jeden Benutzer aus *users_with_del* einen Vektor mit IDs von abgelehnten Artikeln.
- *user_prep_id*: Vektor, der die IDs von Artikeln, die einem Benutzer (aus *users_with_del*) delegiert sind, der diese akzeptiert aber den Report noch nicht geschrieben hat, enthält.
- *pending_rep* enthält für jeden Benutzer aus *users_with_del* einen Vektor mit IDs von Artikeln, zu denen der Report des Reviewers noch nicht eingereicht wurde.
- *reviewer_for_art*: Vektor, der als erstes Element eine Artikel ID enthält und als folgende Elemente die Bewertung des Reviewers (score/confidence), die zu dem begutachteten Artikel abgegeben wurde. Bei nicht vorhandenen Bewertungen enthält der Vektor an entsprechender Stelle `null`. Die Reihenfolge der Speicherung dieser Elemente im Vektor entspricht der Reihenfolge von *users_with_del*.
- *ranking_inf* enthält für jeden Artikel aus *allarticle* einen Vektor mit den Bewertungen der Reviewer (in der Reihenfolge von *users_with_del*). Wurde noch keine Bewertung abgegeben, enthält der jeweilige Vektor an dieser Stelle `null`.

Die durch *GenServiceReport* erzeugten Daten werden durch die WML Datei *reportingsdetails-cont.wml* angezeigt. Die angezeigten Daten wurden bereits beschrieben. Es handelt sich um die für den Dienst Verantwortlichen (Abbildung 4.31), den Informationen zu Artikeln (Abbildungen 4.33, 4.32), den Informationen zu den Reports (Abbildung 4.35), sowie den Delegierungs Informationen (Abbildung 4.34).

Das Sicherstellen der Korrektheit und die Speicherung der Einstellungen zum Service Report (Abbildung 4.29) übernimmt der SIB *CheckReportingConfiguration*. Es wird sichergestellt, dass das Zeitintervall für die Generierung den Wert von einer Stunde nicht unterschreitet und dass für den Report mindestens eine der möglichen Informationsgruppen selektiert wurde. Bei einer ungültigen Konfiguration gelangt der Benutzer wieder zu der Einstellungsseite und wird durch eine Fehlermeldung darauf hingewiesen. Nach erfolgreicher Überprüfung der Einstellungen, werden sie in die *Setup Properties* gespeichert (Abbildung 4.37).

Für die Umsetzung des Threads musste der vorhandene SIB *InitNewThread* angepasst werden, da dieser sein Zeitintervall immer aus den *Service Properties*, nicht aber aus den *Setup Properties* gelesen hat. Zusätzlich zu dieser Funktionalität wurde der neue SIB *InitNewThreadFromSP* um einen Parameter erweitert, der es ermöglicht, die Referenz auf den Thread in einer Servlet Context Variablen zu speichern, so dass er während der kompletten Laufzeit ansprechbar bleibt. Dieses machten wir uns für den SIB *StopThread* (Abbildung 4.37) zu nutze, der über den selben Parameter verfügt und somit einen zuvor gestarteten Thread anhalten kann. Notwendig wurde dies, um eine Änderung am Zeitintervall im laufenden Dienst wirksam zu machen. Das erneute Starten eines Threads nach dem Verändern des Intervalls konnte allerdings nicht durch die bereits erwähnten Thread-SIBs geschehen, da diese nur beim Initialisieren des Dienstes, innerhalb des Init Contextes, ausführbar waren. Die Lösung war der SIB *InitNewCallContextThread*, der an beliebiger Stelle aus dem Call Context heraus ausgeführt werden kann. Auch dieser SIB hinterlegt die Referenz auf den Thread unter einer einstellbaren Servlet Context Variablen, so dass er ebenfalls mit *StopThread* kompatibel ist.

Da normalerweise beim Starten des Threads (also auch beim Starten des Dienstes) die komplette Sequenz von SIBs, die zum Thread gehören, ausgeführt wird, wurde der SIB *Counter* implementiert (Abbildung 4.38). Dieser kann anhand der Anzahl der Aufrufe entscheiden, welchen Zweig im Service Graph er benutzen soll. Die Anzahl der Aufrufe, bis der *pass* Zweig angesprungen wird, kann über den Parameter *block* angegeben werden. Die Anzahl der bereits erfolgten Ausführungen speichert der SIB unter einer Servlet Context Variablen, die über den Parameter *counter_key* angegeben wird. Der Counter wurde in den Service Graph direkt nach dem SIB zum Starten des Threads eingefügt und der *skip* Zweig an der Ausführung des Threads vorbei geleitet (Abbildung 4.38). Beim Ändern des Intervalls wird der Counter durch *ResetCounter* (Abbildung 4.37) zurück gesetzt. Dadurch war es möglich zu verhindern, dass bei jedem Neustart des Dienstes, der Status Report erstellt und an die Benutzer verschickt wird. Auch beim Ändern des Intervalls zahlt sich dies aus. Der Thread lässt somit erst das komplette Intervall verstreichen, bevor er ausgeführt wird, anstatt bei einer Änderung des Intervalls sofort neu zu starten.

Testphase

Zum Testen des Workflows wurden neue Benutzer mit unterschiedlichen Rollen angelegt und diesen unterschiedliche Features (vgl. Abbildung 4.28) zugewiesen. Danach wurde eine Konferenz simuliert, in der Artikel eingereicht und an Reviewer delegiert wurden. Diese akzeptierten die Artikel oder lehnten einige ab. Zu einem Teil der akzeptierten Artikel wurden Reports eingereicht und wieder andere blieben ohne Report. Damit hatten wir eine Basis, auf der der Status Report getestet werden konnte.

Die Umsetzung der Features verlief ohne Probleme. Nur Benutzer, denen das entsprechen-

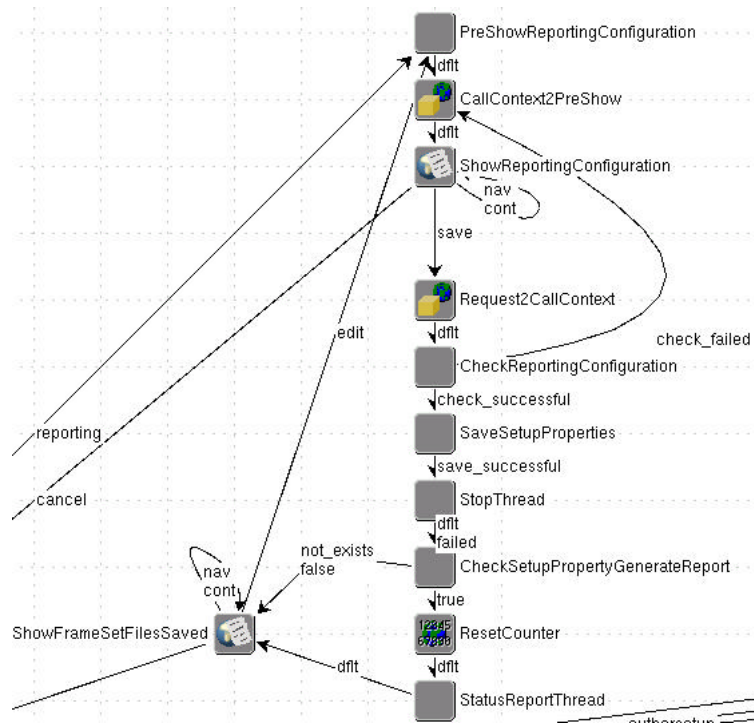


Abbildung 4.37: Service Graph: Setup des Status Reports

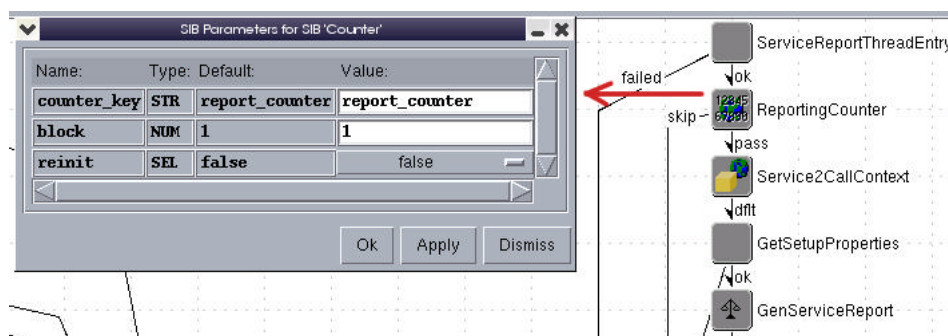


Abbildung 4.38: Service Graph: Der SIB Counter

de Feature zugeschrieben war, konnten auf den Workflow zugreifen und die Funktionen benutzen. Das Versenden des Reports an alle Benutzer, die ihr Abonnement eingeschaltet hatten, funktionierte sowohl mit der CSV Datei, als auch der HTML Version. Weiter war das Generieren und Herunterladen, sowie das Speichern des Reports erfolgreich. Kleinere Änderungen mussten lediglich am Aussehen des Reports gemacht werden.

Das Setup der Reportkonfiguration musste nur hinsichtlich der unteren Intervallgrenze und der Bedingungen für die Mindestanzahl der Inhalte verändert werden.

Kapitel 5

Ausblick

Die in der Projektgruppe Eureka implementierten Features und Dienste sind nach Abschluß der Implementation voll funktionsfähig und entsprechen den Vorgaben der Aufgabenstellung. Während der Implementations- und Testphase wurde deutlich, dass in vielen Fällen weitere Funktionalitäten implementiert werden könnten, um die Funktionalität der Software abzurunden. In den einzelnen Gruppen wurden folgende Verbesserungsvorschläge für eine Weiterentwicklung der jeweiligen Software gesammelt:

Management Overview System

Zum Ende der Arbeiten am Management Overview System ist eines besonders klar geworden: Die Notwendigkeit, die Funktionalitäten des Systems weiter auszubauen und zu verfeinern. Bisher erfüllt der Dienst alle geforderten Grundeigenschaften, es gibt allerdings noch viel Raum zur Verbesserungen. Insbesondere auf OCS-Seite fehlen viele Optionen und Möglichkeiten den Zugriff von MOS-Diensten oder MOS-Usern zu steuern, zu kontrollieren und zu erleichtern. Momentan ist davon auszugehen, dass die im Anschluss dieser PG startende Projekt-Gruppe Eureka 2 die Arbeit an diesem Projekt fortsetzen wird. Es ist also zu erwarten, dass einiges aus der nun folgenden Liste von Verbesserungsvorschlägen umgesetzt werden kann:

Verbindungs-Check Um beim Hinzufügen einer neuen Konferenz in die Konferenz-Liste prüfen zu können ob alle Angaben korrekt sind und sich an der angegebenen Internet-Adresse tatsächlich ein kompatibler Dienst befindet, wäre es wünschenswert, während dieses Workflows die Verbindung testen zu lassen. Es könnte zum einen geprüft werden, ob ein kompatibler Dienst existiert oder zum anderen, ob das angegebene MOS-Passwort vom OCS-Dienst akzeptiert wird.

Rollen-Anfrage Wenn ein MOS-User in einer OCS-Instanz vom MOS-Dienst registriert wurde, so hat der User in der OCS-Instanz nur die standardmäßig festgelegte Rolle zu-

gewiesen bekommen. Es sollte möglich sein, dass der User im OCS-Dienst eine Anfrage stellen kann um eine andere Rolle zu erhalten, zum Beispiel die "PC Chair"-Rolle.

MOS-Einstellungen Bisher können alle Parameter im OCS-Dienst, die mit dem Management Overview System zu tun haben, wie zum Beispiel MOS-Passwörter und Internet-Adressen nur durch die Bearbeitung der Web.xml geändert werden. Es wäre hilfreich, wenn diese Einstellungen über den Dienst zu steuern wären.

MOS-User Sperrung Wenn einem MOS-User die Zugangs-Berechtigung zu einer OCS-Instanz entzogen wird, wird dies lediglich auf MOS-Seite vollzogen. Der User wird in der OCS-Instanz weder gelöscht noch gesperrt. Besonders im Falle, dass der User durch den MOS-Dienst in der OCS-Instanz registriert wurde, könnte es gewünscht sein, dass dieser User im OCS-Dienst ebenfalls gesperrt wird oder ihm die Rolle "MOS-User" automatisch entzogen wird. Eines ist allerdings bereits sichergestellt: Ein durch den MOS-Dienst registrierter User kann sich ohne den MOS-Dienst nicht direkt in die OCS-Instanz einloggen, indem er die Login-Seite benutzt, die Sperrung des Zugangs ist also in jedem Fall sichergestellt.

Verstecken des DefaultMosUser Der im OCS-Dienst registrierte DefaultMosUser, der zum Anzeigen des Status-Reports nötig ist, sollte in einigen Bereichen versteckt werden, da er kein regulärer User ist.

Support und Feedback Dienst

In der derzeitig implementierten Version des Support- und Feedbackdienstes werden die XML Templates, welche für die Generierung der Supportanfrage- und Umfrageseiten verwendet werden, als Text direkt in der XML Datei editiert. Denkbar wäre es, hierfür eine webseitenbasierte Oberfläche zu schaffen, die es ermöglicht, die standardisierten Eingabefelder "grafisch" zu verändern. Darüber hinaus muss das Laden und Speichern von Reports neu implementiert werden, da nach Änderungen an der XML Datei alte Reports nicht mehr geladen werden können, sofern die Reihenfolge oder Anzahl der Fragen verändert wurde. Die zugrunde liegende Implementation im Paket "ewis survey" müsste mehr von der Gestaltung der XML Datei und der Abhängigkeit von Fragetiteln zum Auslesen der einzelnen Datenfelder entkoppelt werden.

Production Feature

Um ein komplettes Tagungsband zu erstellen, muss das Production Feature erweitert werden. In einen Tagungsband existiert zuerst ein Vorwort. Um diese Möglichkeit zu realisieren müsste dem Benutzer die Möglichkeit gegeben werden, ein neues Vorwort zu schreiben oder eine Datei, die das Vorwort enthält, hochzuladen. Desweiteren bietet

das Production Feature die Vergabe einer Überschrift für eine Einleitung an. Um eine Einleitung verfassen zu können, müsste die gleiche Möglichkeit wie für das Vorwort realisiert werden. Am Ende eines Tagungsbandes befindet sich ein Index aller Autoren. Dieser ist alphabetisch nach Namen sortiert und enthält zu jedem Autor die Anfangsseite seines Artikels. Nach Implementierung dieser Möglichkeiten wäre das Production Feature komplett.

Service Status Report Feature

Der Status Report zeigt alle wesentlichen Informationen zur Konferenz an, ohne jedoch Links zu den eigentlichen Inhalten zur Verfügung zu stellen. Durch Einbindung von Links zu Artikeln könnte es dem Benutzer einfacher gemacht werden, die aufgelisteten Informationen im Detail zu betrachten. Darüber hinaus könnte die Seite unterteilt werden, damit bei einer großen Anzahl von Benutzern und Artikeln die Übersicht nicht verloren geht.

Die Tabelle "Ranking Informations" unterscheidet momentan nicht, ob ein Artikel von einem PC Member abgelehnt oder diesem nicht delegiert wurde.

Allgemeines Fazit der Projektgruppe Eureka

Die Zusammenarbeit der Projektgruppe verlief weitgehend unproblematisch. Das Aufteilen der großen Gruppe in vier kleinere Gruppen zur besseren Bearbeitung von Teilaufgaben war sinnvoll. Da die vier Gruppen an einem Projekt arbeiteten, musste sichergestellt sein, dass am Ende alle durchgeführten Änderungen problemlos zusammengeführt werden konnten. Hierbei konnte durch ein Versionskontrollsystem weitgehend automatisiert vorgegangen werden, dennoch musste zuletzt noch manuell dafür gesorgt werden, dass die Änderungen der jeweiligen Gruppen am Hauptgraphen korrekt zusammengefügt wurden. Hierbei traten keine nennenswerten Probleme auf.

Die Arbeit innerhalb der Kleingruppen wurde sinnvoll aufgeteilt, so dass jeder eine Teilaufgabe zu lösen hatte und die Gruppe so effektiver arbeiten konnte.

Sofern Überschneidungen in der Implementierung zwischen den Gruppen auftauchten, konnten diese relativ schnell beseitigt werden. Gravierende Behinderungen traten nicht auf.

Die Betreuer waren zu Beginn der Projektgruppe stets zur Stelle, wenn es Schwierigkeiten mit dem relativ komplexen System gab. Probleme konnten so schnell beseitigt werden und die Teilnehmer lernten das System dabei besser kennen. Später konnten so die meisten kleineren Probleme gruppenintern gelöst werden.

Der Zeitplan der Projektgruppe wurde eingehalten, nennenswerte Verzögerungen bei der Implementierung gab es nicht.

Als größeres Problem entwickelte sich jedoch der nach dem ersten Semester anzuferti-

gende Zwischenbericht, in dem der Stand der Projektgruppe nach den Einzel- und den Gruppenkurzthemen abgebildet werden sollte. Durch nicht Einhalten von Deadlines zur Abgabe der Dokumentationen einiger Teilnehmer konnte der Zwischenbericht nicht zum vorgegebenen Zeitpunkt fertig gestellt werden. Zusätzlich musste er mehrfach Korrektur gelesen werden, bis er in einer vernünftig lesbaren Form vorlag.

Literaturverzeichnis

- [1] Y. Belkhadir, H. Willebrandt 2003 *Comparison of ANT and Makefiles*, Seminar Eureka PG440, Schloß Dagstuhl
- [2] M. Dimov 2003 *General OO-Software-Design*, Seminar Eureka PG440, Schloß Dagstuhl
- [3] M. Drazek 2003 *Webdesign and Dynamic Content*, Seminar Eureka PG440, Schloß Dagstuhl
- [4] M. Dong 2003 *Technology of Tomcat and Servlet*, Seminar Eureka PG440, Schloß Dagstuhl
- [5] C. Kuete-Tsatedem 2003 *Evaluation of Projectmanagement-Tools*, Seminar Eureka PG440, Schloß Dagstuhl
- [6] L. Lamport 2002 *LATEX*, Addison-Wesley
- [7] METAFrame Technologies GmbH 1999 *The Agent Building Center User's Guide*, METAFrame Technologies GmbH
- [8] D. Pierlings 2003 *Tools for Software Development*, Seminar Eureka PG440, Schloß Dagstuhl
- [9] M. Said, J. El Jerroud 2003 *Development with ABC and EWIS*, Seminar Eureka PG440, Schloß Dagstuhl
- [10] D. Saßmannshausen 2003 *Technology of XML*, Seminar Eureka PG440, Schloß Dagstuhl
- [11] B. Steffen, T. Margaria 1999 *METAFrame in Practice: Design of Intelligent Network Services. Correct System Design - Recent Insights and Advances.*, Lecture Notes in Computer Science (LNCS) State-of-the-Art Survey, Vol. 1710, pp. 390–415. Springer-Verlag, Heidelberg, Germany
- [12] J. Zhou 2003 *Concepts of the Postgres DB*, Seminar Eureka PG440, Schloß Dagstuhl

- [13] B. Steffen, T. Margaria, M. Karusseit, M. Bajohr 2003 *Projektgruppe 440 Antrag*, Lehrstuhl 5, Universität Dortmund