

# Endbericht



**Entscheidungs-Unterstützung:**  
**Rollengerecht, Effizient,**  
**Kooperativ, Allgegenwärtig 2**

Abdelazziz Ait Ben Haddou, Dominik Brink, Taieb  
Elfakiri, Omar Zaouaoui Elidrissi, Mikhail  
Farnassov, Holger Hopf, Markus Korte, Philipp  
Langer, Ganna Ostapchuk, Alexander Powolozki,  
Abdelmounaim Ramadane, Marc von Renteln.

PG 465  
am Fachbereich Informatik  
der Universität Dortmund  
Lehrstuhl V

22. Juli 2005

**Betreuer:**

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



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>OJS</b>	<b>3</b>
2.1	Beschreibung des Begutachtungsprozesses . . . . .	3
2.2	Das Entscheidungssystem OJS . . . . .	5
2.3	User- / Rollenkonzept des OJS . . . . .	11
2.3.1	User . . . . .	11
2.3.2	Features und Permissions . . . . .	11
2.3.3	Rollen . . . . .	11
2.4	Funktionalität des OJS . . . . .	14
2.4.1	Article Feature . . . . .	14
2.4.2	Conflicts Feature . . . . .	14
2.4.3	Delegation Feature . . . . .	15
2.4.4	Email Feature . . . . .	15
2.4.5	Evaluation Feature . . . . .	15
2.4.6	My Staff Feature . . . . .	15
2.4.7	Preferences Feature . . . . .	16
2.4.8	Profile Feature . . . . .	16
2.4.9	Reports Feature . . . . .	16
2.4.10	Roles Feature . . . . .	16
2.4.11	Setup Feature . . . . .	17
2.4.12	Tasklist Feature . . . . .	19
2.4.13	Users Feature . . . . .	19
2.5	Technische Realisierung . . . . .	20
2.5.1	Serverstruktur . . . . .	20
2.5.2	verwendete Komponenten . . . . .	22

<b>3</b>	<b>Entwicklungsumgebung ABC</b>	<b>23</b>
3.1	Grundlagen und Konzepte des ABC . . . . .	23
3.2	Makrofizieren . . . . .	27
<b>4</b>	<b>Gruppenkurzthemen</b>	<b>29</b>
4.1	Report Feature . . . . .	29
4.1.1	Beschreibung der Aufgabe . . . . .	29
4.1.2	Modellierungsphase . . . . .	29
4.1.3	Implementierungsphase . . . . .	34
4.1.4	Testphase . . . . .	35
4.2	Article Feature . . . . .	38
4.2.1	Beschreibung der Aufgabe . . . . .	38
4.2.2	Modellierungsphase . . . . .	38
4.2.3	Implementierungsphase . . . . .	42
4.2.4	Testphase . . . . .	42
4.3	Delegation Feature . . . . .	44
4.3.1	Beschreibung der Aufgabe . . . . .	44
4.3.2	Modellierungsphase . . . . .	44
4.3.3	Implementierungsphase . . . . .	46
4.3.4	Testphase . . . . .	51
4.4	Delegation Feature 2 . . . . .	53
4.4.1	Beschreibung der Aufgabe . . . . .	53
4.4.2	Modellierungsphase . . . . .	53
4.4.3	Implementierungsphase . . . . .	55
4.4.4	Testphase . . . . .	57
4.5	Report Feature 2 . . . . .	59
4.5.1	Beschreibung der Aufgabe . . . . .	59
4.5.2	Modellierungsphase . . . . .	59
4.5.3	Implementierungsphase . . . . .	61
4.5.4	Testphase . . . . .	64
4.6	Evaluation Feature . . . . .	65
4.6.1	Beschreibung der Aufgabe . . . . .	65
4.6.2	Modellierungsphase . . . . .	65

4.6.3	Implementierungspahse . . . . .	66
4.6.4	Testphase . . . . .	69
<b>5</b>	<b>Gruppenthemen 1. Semester</b>	<b>75</b>
5.1	Verwaltung der Co-Autoren . . . . .	75
5.1.1	Beschreibung der Aufgabe . . . . .	75
5.1.2	Modellierungsphase . . . . .	75
5.1.3	Implementierungsphase . . . . .	76
5.1.4	Testphase . . . . .	83
5.2	Article Makro - Manage Versions . . . . .	86
5.2.1	Beschreibung der Aufgabe . . . . .	86
5.2.2	Modellierungsphase . . . . .	87
5.2.3	Implementierungsphase . . . . .	93
5.2.4	Testphase . . . . .	99
5.3	Role Feature . . . . .	102
5.3.1	Beschreibung der Aufgabe . . . . .	102
5.3.2	Modellierungsphase . . . . .	102
5.3.3	Implementierungsphase . . . . .	105
5.3.4	Testphase . . . . .	109
5.4	Erweiterung des User Features . . . . .	112
5.4.1	Beschreibung der Aufgabe . . . . .	112
5.4.2	Modellierungsphase . . . . .	112
5.4.3	Implementierungsphase . . . . .	115
5.4.4	Testphase . . . . .	118
<b>6</b>	<b>Gruppenthemen 2. Semester</b>	<b>123</b>
6.1	Delegation Stack . . . . .	123
6.1.1	Beschreibung der Aufgabe . . . . .	123
6.1.2	Modellierungsphase . . . . .	124
6.1.3	Klassendiagramm . . . . .	126
6.1.4	Implementierungsphase . . . . .	127
6.1.5	Testphase . . . . .	130
6.2	Reviewer Allocation Feature . . . . .	133

6.2.1	Beschreibung der Aufgabe . . . . .	133
6.2.2	Modellierungsphase . . . . .	134
6.2.3	Implementierungsphase . . . . .	140
6.2.4	Testphase . . . . .	145
6.3	Distribution Feature . . . . .	154
6.3.1	Beschreibung der Aufgabe . . . . .	154
6.3.2	Modellierungsphase . . . . .	154
6.3.3	Implementierungsphase . . . . .	155
6.3.4	Testphase . . . . .	159
6.4	XML-Import/Export für Reports . . . . .	162
6.4.1	Beschreibung der Aufgabe . . . . .	162
6.4.2	Modellierungsphase . . . . .	162
6.4.3	Implementierungsphase . . . . .	166
6.4.4	Testphase . . . . .	172
<b>7</b>	<b>Fazit</b>	<b>177</b>
	<b>Literaturverzeichnis</b>	<b>179</b>



# Abbildungsverzeichnis

2.1	Login-Seite . . . . .	5
2.2	Registrierungsformular . . . . .	6
2.3	Delegation Matrix . . . . .	7
2.4	Evaluation Matrix . . . . .	8
2.5	OJS - Geschäftsprozess . . . . .	10
2.6	Bearbeiten einer Rolle . . . . .	13
2.7	Dienstestruktur . . . . .	21
3.1	Lifecycle des ABC . . . . .	25
3.2	Ein Service Logic Graph . . . . .	26
4.1	Use Case: Editorial Office . . . . .	30
4.2	Use Case: Reviewer . . . . .	31
4.3	Activity Chart: Rolle modifizieren . . . . .	32
4.4	Activity Chart: Liste der Reports . . . . .	33
4.5	Ausschnitt aus dem ABC . . . . .	34
4.6	Rollen Management - Report Feature . . . . .	36
4.7	Liste verbundener Reports . . . . .	37
4.8	Liste aller Reports . . . . .	37
4.9	Use case Diagramm: Ist-Zustand. . . . .	38
4.10	Aktivitätsdiagramm: Ist-Zustand. . . . .	39
4.11	Use case Diagramm: Soll-Zustand mit Feature view metadata anonym (F-ART-15) . . . . .	40
4.12	Aktivitätsdiagramm: Soll-Zustand mit Feature view metadata anonym (F-ART-15) . . . . .	41
4.13	Email ohne Autor . . . . .	42

4.14	Email mit Autor . . . . .	43
4.15	Use Case „Extend Decision/Task Deadline“ – Anfangszustand . . . . .	45
4.16	Activity Chart „Extend Decision/Task Deadline“ – Anfangszustand . . . . .	47
4.17	Activity Chart „Extend Decision/Task Deadline“ – Zielzustand . . . . .	48
4.18	Workflow „Extend Decision/Task Deadline“ – Anfangszustand . . . . .	49
4.19	Workflow „Extend Decision/Task Deadline“ – Zielzustand . . . . .	50
4.20	Delegation Matrix . . . . .	51
4.21	Eingabeformular – Extend Decision Deadline . . . . .	52
4.22	Extend Decision Deadline – Eingabe der inkorrekten Testdaten . . . . .	52
4.23	Extend Decision Deadline – Fehlermeldung . . . . .	52
4.24	Alter Use Case . . . . .	54
4.25	Altes Activity Chart . . . . .	54
4.26	Neuer Use Case . . . . .	55
4.27	Neuer Activity Chart . . . . .	56
4.28	Sortieren nach Artikelkategorien . . . . .	58
4.29	Das alte Activity Chart . . . . .	60
4.30	Das neue Use Case . . . . .	61
4.31	Das neue Activity Chart . . . . .	62
4.32	Der modifizierte Workflow . . . . .	63
4.33	ReportSubmissionForm mit dem Link Information to old report . . . . .	64
4.34	Old report Liste . . . . .	64
4.35	Use Case Auswertung/Modifikation der Evaluation Matrix . . . . .	66
4.36	Activity Chart: Änderung von Scores / Confidences . . . . .	67
4.37	Activity Chart: Artikelstatus auf angenommen setzen . . . . .	68
4.38	Vorbereitungsphase des Workflows . . . . .	70
4.39	Durchführungsphase des Workflows . . . . .	71
4.40	Benachrichtigungsphase des Workflows . . . . .	72
4.41	Selektion der Artikel und Eingabe der zu modifizierender Daten . . . . .	73
4.42	Bestätigung der gewünschten Änderungen . . . . .	73
4.43	Benachrichtigung über durchgeführte Änderungen . . . . .	74
5.1	Use Case . . . . .	77
5.2	Activity Chart . . . . .	78

5.3	Klassendiagramm . . . . .	79
5.4	Macro CoAuthorList . . . . .	81
5.5	Macro CoAuthorRegistration . . . . .	82
5.6	Liste der Co-Autoren . . . . .	84
5.7	Eingabeformular - Co-Autoren . . . . .	84
5.8	Use Case „Article Feature“ – Anfangszustand . . . . .	88
5.9	Use Case „Article Feature“ – Zielzustand . . . . .	88
5.10	Activity Chart „Delete Article Version“ – Zielzustand . . . . .	89
5.11	Activity Chart „Restore Article Version“ – Zielzustand . . . . .	90
5.12	Activity Chart „Entfernen des Artikels“ – Zielzustand . . . . .	91
5.13	Klassendiagramm . . . . .	92
5.14	ArticleVersion Makro . . . . .	95
5.15	RemoveArticleVersionFile Makro . . . . .	96
5.16	RestoreArticleVersionFile Makro . . . . .	97
5.17	SIB RemoveArticleVersionData . . . . .	98
5.18	Liste der Artikel . . . . .	99
5.19	Dateiliste der Artikel . . . . .	100
5.20	Ausführen – Löschen . . . . .	100
5.21	Benachrichtigung über Löschenvorgang . . . . .	100
5.22	Das Löschen von der Liste von zugänglichen Files des Artikels . . . . .	101
5.23	Bestätigung des Wiederherstellens . . . . .	101
5.24	Benachrichtigung über Dateiwiederherstellung . . . . .	101
5.25	Use case Diagramm: Ist-Zustand . . . . .	103
5.26	Aktivitätsdiagramm: Ist-Zustand . . . . .	104
5.27	Use case Diagramm: Soll-Zustand. . . . .	105
5.28	Aktivitätsdiagramm . . . . .	106
5.29	Klassendiagramm . . . . .	107
5.30	Makro . . . . .	108
5.31	Benutzerliste . . . . .	110
5.32	Liste der Rollen . . . . .	110
5.33	Maske zum Modifizieren der einzelnen Rolle . . . . .	111
5.34	Das neue Use Case . . . . .	113

5.35	Das neue Activity Chart . . . . .	114
5.36	Das eingefügte useroverview Sub-Macro in Users Macro . . . . .	115
5.37	Das Macro useroverview . . . . .	116
5.38	Der Link user overview und die Möglichkeit die Benutzerliste nach Rollen zu filtern . . . . .	118
5.39	Informationen über einen Benutzer . . . . .	119
5.40	Informationen über einen Benutzer (Ergänzung) . . . . .	120
5.41	CSV-Ansicht der Benutzerinformation . . . . .	121
6.1	Use Case Diagramm . . . . .	124
6.2	Aktivitätsdiagramm . . . . .	125
6.3	Zuweisung von Objekten . . . . .	125
6.4	Architektur der Delegation im OJS . . . . .	126
6.5	Die an den OJS angepassten Klassen . . . . .	128
6.6	Allgemeine und die an den OJS angepassten Klassen . . . . .	129
6.7	Liste der Artikel . . . . .	131
6.8	Delegation an die Staff-Member . . . . .	131
6.9	Delegierte Reviews . . . . .	132
6.10	Use Case „Reviewer Allocation Feature“ – Zielzustand . . . . .	135
6.11	Activity Chart „Assign Editors“ – Zielzustand . . . . .	136
6.12	Activity Chart „Propose Reviewers“ – Zielzustand . . . . .	137
6.13	Activity Chart „Delegate Article“ – Zielzustand . . . . .	138
6.14	Klassendiagramm . . . . .	139
6.15	Makro ReviewerAllocation . . . . .	140
6.16	Makro RAFA2E . . . . .	141
6.17	Makro RAFE2D . . . . .	142
6.18	Makro RAF2Delegation . . . . .	143
6.19	Reviewer Allocation – Editorial Office – Startseite . . . . .	145
6.20	Article-Editor Assignment Matrix – „not delegated“ Filter . . . . .	146
6.21	Article-Editor Assignment Matrix – „all“ Filter . . . . .	146
6.22	Assignment of Editors – Bestätigungsseite . . . . .	147
6.23	Assignment of Editors – Acknowledgementseite . . . . .	147
6.24	Reviewer Allocation – Editor – Startseite . . . . .	148

6.25	Reviewer Allocation – Vorschlag von Reviewers . . . . .	148
6.26	Reviewer Allocation – das Vorschlagsformular . . . . .	149
6.27	Suggestion of Reviewers – fehlende Email-Adresse . . . . .	149
6.28	Suggestion of Reviewers – falsches Email-Format . . . . .	150
6.29	Suggestion of Reviewers – das Fehlen von Vor- und Nachname . . . . .	150
6.30	Suggestion of Reviewers – Bestätigungsseite . . . . .	151
6.31	Delegate article – Liste der delegierten Reviews . . . . .	151
6.32	Delegate article – Liste der aufgetretenen Fehler . . . . .	151
6.33	Delegate article – neue Delegation . . . . .	152
6.34	Delegat article – Acknowledgementseite . . . . .	152
6.35	Delegation Feature – Ausschnitt aus der Delegation Matrix . . . . .	153
6.36	Use Case . . . . .	155
6.37	Activity Chart . . . . .	156
6.38	Übersicht über die neu implementierten SIBs im Makro Bidding Matrix .	157
6.39	Bidding matrix vor der Verwendung des Solvers . . . . .	160
6.40	Bidding matrix nach der Verwendung des solvers . . . . .	161
6.41	Use Case Import, Export XML Reports . . . . .	163
6.42	Activity Chart Import XML Reports . . . . .	164
6.43	Activity Chart Modify XML Reports . . . . .	165
6.44	Download XML Template . . . . .	167
6.45	Export XML Reports . . . . .	169
6.46	Import XML Reports . . . . .	171
6.47	Der Link Download XML Template . . . . .	173
6.48	Die Import XML Report –Seite . . . . .	173
6.49	Bestätigungsseite der eingereichten Reports in der XML Datei . . . . .	174
6.50	Der Link Export Reports in der ReportList–Seite . . . . .	175
6.51	Die Export–ReportList . . . . .	175
6.52	Import des modifizierten XML Reports . . . . .	176



# Kapitel 1

## Einleitung

In diesem Endbericht der Projektgruppe EUREKA II (**E**ntscheidungs-**U**nterstützung: **R**ollengerecht, **E**ffizient, **K**ooperativ, **A**llgegenwärtig) des Lehrstuhls V geht es um die Beschreibung der Arbeit der Projektgruppe hinsichtlich des Online Journal Service (OJS) in den vergangenen zwei Semestern.

Die Projektgruppe beschäftigt sich mit der Weiterentwicklung des OJS. Der OJS ist ein personalisiertes und internetbasiertes Entscheidungssystem, das wissenschaftliche Begutachtungsprozesse koordiniert und Nutzer weltweit, rund um die Uhr, mit aktueller, auf sie zugeschnittener Information versieht. Die Benutzer werden durch das System bei ihren Aktivitäten unterstützt. Autoren können im OJS ihre Artikel einreichen, die von so genannten Reviewern begutachtet und bewertet werden. Des Weiteren schreiben Reviewer Gutachten (Reports) zu den ihnen zugewiesenen Artikeln. Anhand dieser Gutachten wird entschieden welche Artikel veröffentlicht werden. Der gesamte Kommunikationsprozess wird vom OJS unterstützt.

Derartige Kooperations- und Entscheidungssysteme werden durch mehrstufige Web-Applikationen als Online-Dienst realisiert, die auf einer Client-Server-Architektur basieren. Der Client (Frontend) ist ein Browser, der Transaktionen generiert und an einen Webserver (Backend) zur Verarbeitung übergibt. Der Webserver leitet diese Anfrage an einen Applikationsserver weiter, der unter Verwendung von verteilten Systemen (z.B. Datenbank, Versionsverwaltungssystem, ...) die benötigten Daten zusammenstellt, um anschließend dynamisch die angeforderte Web-Seite zu generieren.

Das System soll hinsichtlich einer kooperativen Entscheidungsfindung unabhängig von Zeit und Ort an moderne Maßstäbe angepasst werden. Hierbei stand bei der Weiterentwicklung des Systems die Modularität, objektorientiertes Softwaredesign, Rollen- und Feature-basierte Personalisierung und die flexible Anpassung der Ablaufsteuerung im Vordergrund.

Die Gruppe hatte die Aufgabe dieses System flexibler (neue anwendungsspezifische Systeme) zu gestalten, des Weiteren ist die Architektur feingranularer aufzubauen und der OJS wird zudem durch weitere Komponenten modular erweitert werden. Die Vorgehens-

weise der Gruppe ist Pattern-basiert, sowohl auf Architektur als auch auf Prozessebene. Es soll von der Gruppe außerdem eine konzeptuelle Architektur/ Prozess-Strukturen und Vorgehensweisen für das betrachtete Anforderungsprofil herausgearbeitet werden. Das System soll dabei Transparent, Effizient und Flexibel sein. Hier ist ein Dienst- und Featureorientierte Vorgehensweise vorgesehen. Sie soll die Spezifikation des „Was“ auf der Anwendungsebene von dem „Wie“ auf der Realisierungsebene trennen.

Bevor sich die Gruppe mit den Aufgaben zum OJS vertraut gemacht hat, wurde gemeinsam mit den Betreuern ein Seminar im Hause Bommerholz veranstaltet. Im Zuge dieses Seminars musste jedes Gruppenmitglied eine Seminararbeit ausarbeiten, die sich jeweils mit einem Themengebiet befasste, das zur Weiterentwicklung des OJS benötigt wird. Im Rahmen eines Vortrages hat dann jedes Mitglied der Gruppe den anderen Teilnehmern seine Arbeit vorgestellt. Sinn dieses Seminars war es, alle Gruppenmitglieder auf denselben Wissenstand zu bringen. Die einzelnen Ausarbeitungen sind im Anhang [3], [4], [5], [6], [7], [8], [9],[10], [11], [12], [13] und [14] zu finden.

In Kapitel 2 wird eingehend auf den OJS eingegangen. Die Begutachtungsprozesse des Entscheidungssystems sowie das User-/ Rollenkonzept werden detailliert vorgestellt und erläutert. Es wird zudem auf die Erklärung und Umsetzung einzelner Features innerhalb des Systems eingegangen und deren technische Realisierung wird ausführlich beschrieben. Das folgende dritte Kapitel widmet sich dem ABC (Application Building Center), das der Gruppe als Grundlage zur Umsetzung ihrer Ziele dient. Das ABC ist eine graphische Entwicklungsumgebung für prozessorientiertes, komponentenbasiertes Softwaredesign, welches am Lehrstuhl V entwickelt wurde.

Kapitel 4 behandelt die ersten Themen, die von den Gruppen innerhalb der Projektgruppe zu lösen waren und bildet die ersten Schritte. Es waren kleinere Lösungen gefordert, die zur Einarbeitung in den OJS dienten.

Kapitel 5 ist der eigentlichen Arbeit des ersten Semesters der Gruppe gewidmet, die Erweiterungen des OJS, die von der Gruppe vorgenommen wurden, werden dort vorgestellt. Die einzelnen Gruppen beschreiben hierbei ihre Aufgabe und die dazugehörige Modellierungsphase (UML). Weiterhin wird von den jeweiligen Gruppen die Implementierungsphase und die Testphase erläutert und beschrieben.

Kapitel 6 beschreibt schließlich in gleicher Weise die Arbeit des zweiten Semesters. Diese (Haupt-)Themen, die auch auf dem PG-Antrag vermerkt waren, erfordern einige Kenntnis des OJS und sind der krönende Abschluss der PG.

Im weiteren Verlauf dieser Arbeit wird noch ein kurzer Ausblick folgen, in dem die Zukunft des OJS kurz angesprochen wird. Ein Literaturverzeichnis vervollständigt den Endbericht, der Leser kann dort tiefergehende Literatur und Beschreibungen zu den Entwicklungswerkzeugen und technischen Realisierungen finden.

# Kapitel 2

## OJS

Im folgenden Kapitel wird nun der Online Journal Service (OJS) vorgestellt. Hierzu werden nach einer kurzen Beschreibung von Begutachtungsprozessen im Allgemeinen deren Umsetzung im Rahmen des OJS erläutert. Danach wird das User- / Rollenkonzept näher betrachtet und die bereitgestellten Funktionalitäten sowie die technische Realisierung vorgestellt.

### 2.1 Beschreibung des Begutachtungsprozesses

Bei einer Konferenz handelt es sich um das Zusammentreffen von Personen zum gemeinsamen Austausch von Wissen. Zum Ausrichten einer Konferenz gehört dabei nicht nur das Ausschreiben der Konferenz mit den Themen und das Bereitstellen von Seminarräumen oder Hörsälen. Vielmehr ist es eine organisatorische Herausforderung aus einer Vielzahl von Papers (Beiträgen) diejenigen auszuwählen, die den Ansprüchen der Konferenz genügen, welche von den Einreichern auf der Konferenz vorgestellt werden sollen. Die Auswahl der Redner für eine Konferenz erfolgt auf der Grundlage der von ihnen eingereichten Papers. Somit ist das Begutachtungsverfahren der Schlüsselprozess.

Der Begutachtungsprozess beginnt mit der Einreichung (per Post oder per Email) von Artikeln durch die Autoren. Dann nach Ablauf der Einreichungsfrist werden diese Artikel an die Komiteemitglieder zur Begutachtung delegiert. Dabei sollte Expertenwissen des Gutachters berücksichtigt werden und Interessenskonflikte (der Begutachter kennt den Autor persönlich oder ist der Mitautor des ihm zugeteilten Artikels) vermieden werden. Je nach Anzahl der zugeteilten Beiträge leiten die Komiteemitglieder diese an andere Personen (Reviewer) weiter. Nach der Beurteilung eines Artikels wird vom Reviewer bzw. dem Komiteemitglied ein Report verfasst und an den Komiteevorsitz zurückgesendet. Anhand der Bewertungskriterien in den Gutachten und einer darauf angewandten Auswertungsfunktion wird eine Gradierung erstellt, so dass vom Komiteevorsitz festgelegt wird, welche Paper zur Veröffentlichung akzeptiert werden. Ein Durchlauf des Begutachtungsprozesses

endet mit dem Versenden von abschließenden anonymisierten Gesamtgutachten an die Autoren. Die Autoren der akzeptierten Artikel fertigen abschließend unter Berücksichtigung der Gutachten eine Endversion an und schicken diese zurück.

Die Koordination des Begutachtungsprozesses und die Kommunikation mit allen Autoren und Komiteemitgliedern erfordern viel Zeit. Erschwerend kommt hinzu, dass in den meisten Fällen beteiligte Personen über die ganze Welt verteilt sind, d.h. die Papers müssen verschickt, bewertet, zurückgeschickt werden. Die verschiedenen festgesetzten Fristen stellen zusätzliche Herausforderungen dar.

Um all diese Schwierigkeiten zu umgehen, wurde eine Plattform im Internet geschaffen, der „Online Journal Service“ [1]. Dieser Service zentralisiert und erleichtert alle Kommunikations- und Verwaltungsprozesse während des Begutachtungsverfahrens.

## 2.2 Das Entscheidungssystem OJS

Online Journal Service (OJS) ist ein personalisiertes und internetbasiertes Entscheidungssystem. Es bietet eine vollständig einstellbare, moderne, rollenbasierte Unterstützung des wissenschaftlichen Begutachtungsprozesses. Es erleichtert und koordiniert das Management des Begutachtungsprozesses von der Einreichung bis zur Endversion. Während der Konferenzvorbereitung versorgt dieser Service die beteiligten Personen mit allen relevanten Daten und gibt jedem Benutzer eine maßgeschneiderte Sicht auf Dienstfunktionalitäten und Daten. Der OJS ist rund um die Uhr über eine Standard-Benutzerschnittstelle (Browser) weltweit verfügbar. Um diesen Dienst benutzen zu können, muss der Benutzer sich erst registrieren, bzw. registrieren lassen. Die Abbildung 2.1 zeigt die Login-Seite und die Abbildung 2.2 – das Registrierungsformular:

**Login**

Please enter your email address and your password into the following fields.

Email Address

Password  (If you do not know your password anymore please use the "Forgot my Password" feature on the left side of this page!)

Then click the "Login" button to enter the service.

---

**Required Configuration**

- Browser: Netscape 4.0 or higher, or Internet Explorer 5.0 or higher
- (for more comfort enabled JavaScript, but not necessary)

---

**Note:** After 120 minutes of inactivity you will be logged out automatically and your session will be terminated. Then you have to login again.

---

**Demonstration Logins**

*Note:* These demo logins only work with the password "secret1". Please do not change the passwords. Otherwise you have to enter both email address and password into the login form above.

login as a, a (UID=1) Assigned Roles: **Administrator Author Editorial Office**

login as b, b (UID=2) Assigned Roles: **Reviewer**

Abbildung 2.1: Login-Seite

Please register to the Journal Online Service by filling in the following form. Your login identifier will be your email address and it will be authenticated via a password of your choice.

## Registration Form

Please fill in the following form carefully to get a new login identifier for the Service. Optional fields are marked in *emphasized* font.

Email Address (Login)	<input type="text"/>
Email Address (Verification)	<input type="text"/>

Please choose a password matching the following conditions:

- length at least 5 characters
- includes a digit or a special character
- is different from your first-, lastname and email address

The registration will reject passwords not matching these conditions!

Password	<input type="password"/>
Password(Verification)	<input type="password"/>

**Important: Be sure to remember your password!**  
You will need it to access the reviewers' reports and to submit the final article version!

If you forget your password use the [Forgot my Password feature](#). This feature requires the input of the following two text fields.

Forgot my password question	<input type="text"/>
Forgot my password answer	<input type="text"/>

Forms of Address  Mr.  Ms.

<i>Title</i>	<input type="text"/>
First name	<input type="text"/>
<i>Prefix</i>	<input type="text"/>
Last name	<input type="text"/>
<i>Institute</i>	<input type="text"/>
<i>Department</i>	<input type="text"/>
Street	<input type="text"/>
Zipcode	<input type="text"/>
City	<input type="text"/>
Country	<input type="text"/>
<i>Phone</i>	<input type="text"/>
<i>Fax</i>	<input type="text"/>

Abbildung 2.2: Registrierungsformular

Der Begutachtungsprozess mittels des OJS kann so skizziert werden: Die Autoren reichen Artikel ein. Es ist möglich, die Artikel in verschiedenen Formaten, wie pdf/ps/html mit Hilfe des Web-Browsers einzureichen. Der Dienst bietet Vorlagen und Leitfäden für Einreichung, Modifizierung und Upload der Artikel. Es ist nicht nötig, FTP-Befehle oder Email-Anhänge zu verwenden. Der Dienst benachrichtigt den Nutzer sofort über eine erfolgreiche Einreichung. Die zu den Artikeln gehörige Dateien werden im Datei-Repository des Dienstes gespeichert. Das Repository erlaubt eine Versionisierung der Artikeldateien. Diese Artikel werden dann an die Reviewer delegiert. Die Verteilung der Begutachtungsaufgaben (Reporttasks) wird vom Editorial Office, Leiter des Begutachtungsprozesses, anhand von den Editoren abgegebener oder eigener Reviewerlisten durchgeführt. Der Gutachter kann den Reporttask entweder akzeptieren oder ablehnen. Im Fall, dass der Reviewer den Reporttask ablehnt, kann das Editorial Office den Artikel an einen anderen Gutachter delegieren. Nach der Beurteilung eines Artikels wird vom Reviewer ein Report verfasst und an das Editorial Office zurückgesendet. Anhand der Delegation Matrix (Abbildung 2.3) kann dieser den aktuellen Stand der einzelnen Reporttasks kontrollieren.

**Overview of Delegated Reviews**

This pages gives an overview of the delegated reviews, i.e. when and to whom you have delegated a review of an article. So you can get the information about the "position" of articles in the reviewing process.

The legend at the bottom of this page explains the content of each column.

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

[Refresh](#) (last refresh: *Wed Feb 16 13:09:11 CET 2005*)

ID	Delegated from	Delegated to	Status(Since)	Decision Deadline	Task Deadline	Action	Comments	Type	ID
003	aa	C C	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	003
003	aa	D D	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	003
003	aa	b b	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	003
004	aa	D D	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	004
004	aa	b b	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	004
005	aa	C C	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	005
005	aa	D D	done, report submitted(on 2005-02-16)				Please review this paper and submit a report about it.	Regular Paper	005

Abbildung 2.3: Delegation Matrix

Der OJS unterstützt den gesamten erforderlichen Kommunikationsprozess, inklusive Deadlinemanagement, phasenabhängiger Geheimhaltungsvorschriften (z.B. „Chinese Walls“-Kriterium), und Konfliktregelungen. Anhand der Bewertungskriterien in den Gutachten und einer darauf angewandten Auswertungsfunktion wird eine Gradierung erstellt, so dass vom Editorial Office festgelegt wird, welche Paper zur Veröffentlichung akzeptiert werden. Nachdem alle Gutachten eingegangen sind, entscheidet das Editorial Office anhand der Gutachten, welche Artikel akzeptiert werden sollen. In der Evaluation Matrix sind alle Auswertungen zu jedem Artikel zusammengefasst. Abbildung 2.4 stellt die Evaluation Matrix dar.

Die akzeptierten Artikel werden in *direkt akzeptierte* (Accepted) Artikel und in *vielleicht akzeptierte* (Propose Accepted) Artikel unterschieden. Im Falle eines akzeptierten Artikels

### 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 Reviewers show the corresponding **score/confidence** value of the report written by the particular member.

**Refresh** (last refresh: *Wed Feb 16 13:06:25 CET 2005*)

**Caution, operating with the Select All or the Unselect All actions will have the loss of all score / confidence changes not yet committed as consequence. First select the articles, than update the scores / confidences you want to change.**

Sel.	Art.ID	R.	Ø	Scores			Status	Type	Art.ID
<input type="checkbox"/>	004   R	1	5.0	DD: 5/2	bb: 5/2		accepted	Regular Paper	004   R
<input type="checkbox"/>	005   R	2	4.55	CC: 4/2	DD: 5/3		accepted	Regular Paper	005   R
<input type="checkbox"/>	003   R	3	4.15	CC: 5/3	DD: 3/1	bb: 4/3	rejected	Regular Paper	003   R

- modifies the score/confidence values
- set the new article state to **Accepted**
- set the new article state to **Propose Accepted**
- set the new article state to **Rejected**

modify

**Caution, operating with the Select All or the Unselect All actions will have the loss of all score / confidence changes not yet committed as consequence. First select the articles, than update the scores / confidences you want to change.**

Abbildung 2.4: Evaluation Matrix

muss der Autor noch seine Endversion des Artikels (final article version) einreichen. Im Falle eines vielleicht akzeptierten Artikels reicht er seine überarbeitete, verbesserte Version des Artikels ein, die den Begutachtungsprozess erneut durchläuft. Ein Durchlauf des Begutachtungsprozesses endet mit dem Versenden von abschließenden Gesamtgutachten an die Autoren [2].

Die Abbildung 2.5 spiegelt das gesamte Begutachtungsverfahren wieder.

Legende:

- A. - Autor
- E. - Editor
- E.O. - Editorial Office
- R. - Reviewer

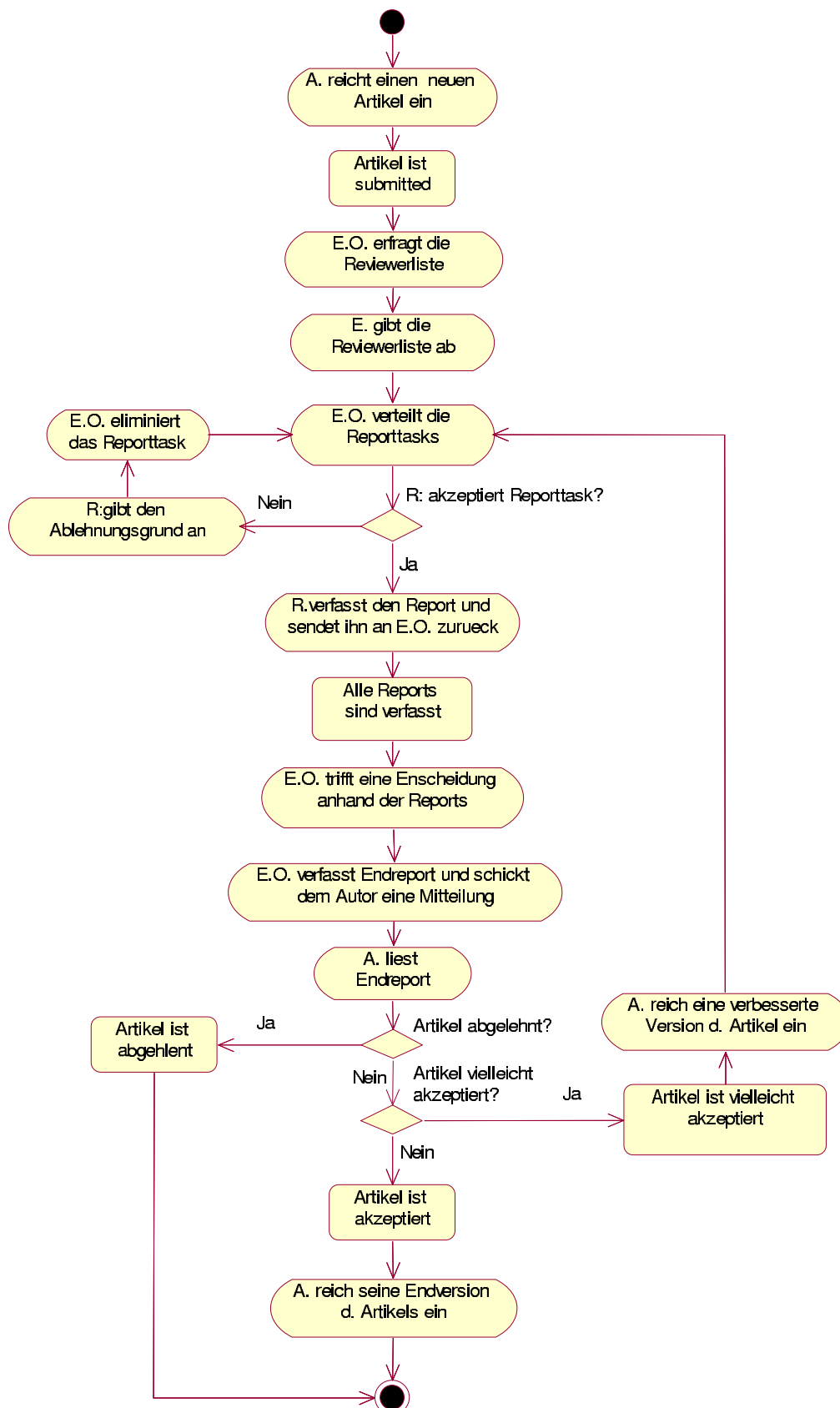


Abbildung 2.5: OJS - Geschäftsprozess

## 2.3 User- / Rollenkonzept des OJS

### 2.3.1 User

Benutzer (*User*) sind Konferenzteilnehmer. Dabei kann es sich um Autoren, Reviewer oder Aufsichtspersonen handeln, die sich um den Workflow kümmern. Am gängigsten ist wohl die Anmeldung eines Autors:

Wenn jemand einen Artikel geschrieben hat und diesen im OJS einreichen möchte, muss er zuerst einen *Benutzeraccount* anlegen. Dazu füllt er ein Formular aus, in dem Daten wie Name, Adresse, Telefonnummer und Email-Adresse eingetragen werden. Letztere nimmt im System einen besonderen Stellenwert ein, denn hierdurch wird die Validität und Eindeutigkeit des Accounts überprüft. Dem Benutzer wird eine Bestätigungs-Email geschickt. Es ist nicht möglich, zwei verschiedene Accounts auf die selbe Email-Adresse anzumelden.

Nachdem der Benutzer nun angemeldet ist, kann er sich einloggen. Dazu legt er sich ein Passwort fest, das seinen Account schützt.

Dem Benutzer können mehrere *Rollen* zugewiesen werden. Je nachdem, welche Rolle er hat, stehen ihm verschiedene Funktionen zur Verfügung. Im Folgenden soll auf dieses Verhalten des Systems weiter eingegangen werden.

### 2.3.2 Features und Permissions

Das System gliedert sich in mehrere, funktional von einander getrennte Einheiten. Da wären z.B. das User Management, der Bereich zum Einreichen von Artikeln oder der zur Bewertung ebensolcher. Diese Bereiche nennt man *Features*.

Jedes Feature erlaubt spezifische Aktionen, sei es an den Benutzern, am System selbst, an Artikeln oder Reviews. Um diese Aktionen ausführen zu können, braucht man die entsprechende *Berechtigung (Permission)*.

Permissions gliedern sich in verschiedene Bereiche, so beginnen alle Permissions, die sich mit dem Rollen-Management beschäftigen, mit „F-ROLE-“ gefolgt von einer zweistelligen Nummer zur Identifikation. Sie können auch in Unter-Permissions aufgeteilt sein, um noch genauer anzeigen zu können, was der Benutzer darf und was nicht. So gibt „F-REV-06“ an, dass ein Benutzer die Liste der Reviews anzeigen darf. In den Unter-Permissions wird festgehalten, welche Art von Reviews er angezeigt bekommt. „F-REV-06.all“ zeigt alle Reviews an, wohingegen „F-REV-06.own“ nur die Reviews des Benutzers selbst anzeigt. Meist können auch mehrere Unter-Permissions miteinander kombiniert werden, was das Verhalten noch feiner einstellbar macht.

### 2.3.3 Rollen

Jedem Benutzer wird eine oder mehrere Rollen zugewiesen. Sie legen fest, was dem Benutzer innerhalb des Systems erlaubt ist. Jeder Rolle werden Berechtigungen zuge-

wiesen, d.h. für jede Rolle kann eingestellt werden, welche Funktionen des Systems sie benutzen oder anzeigen darf. Dazu gibt es das Rollenmanagement, wo jedes Feature einzeln hinzugefügt oder entzogen werden kann (siehe Abbildung 2.6).

So kann jede Rolle für bestimmte Aufgaben optimiert werden. Außerdem wird verhindert, dass ein Benutzer Bereiche sieht oder gar bearbeiten kann, die nicht für ihn bestimmt sind. So kann in der Standard-Einstellung ein Autor nicht in den Bereich der Reviews und sehen, was zu seinem Artikel geschrieben wurde. (Standard-Einstellung deswegen, weil das System den individuellen Bedingungen angepasst werden kann und auch diese Funktion für einen Autor freigeschaltet werden könnte - ob das nun Sinn ergibt, ist eine andere Sache...)

Es gibt 5 voreingestellte Rollen:

1. **Administrator**

Der Administrator kümmert sich um die Einrichtung des Systems und die erste Vergabe der Berechtigungen an die übrigen Rollen. Er darf so ziemlich alles.

2. **Editorial Office**

Leitet die Konferenz und verwaltet den Workflow zwischen den verschiedenen Benutzern. Er bestimmt, wer einen Review zu einem Artikel schreibt. Vor allem ist er aber auch für die Verwaltung von Rollen und Benutzern zuständig.

3. **Author**

Reicht Artikel ein.

4. **Reviewer**

Bewertet die ihm zugewiesenen Artikel.

5. **Editor**

Schlägt Reviewer vor.

Wenn sich ein Benutzer einloggt, dem mehrere Rollen zugeordnet wurden, sieht er in der Navigationsleiste, welche Rolle er gerade verwendet. Seine verschiedenen Aufgabenbereiche sind von einander getrennt und er muss mit einem Klick auf die entsprechende Rolle selbige wechseln, bevor er die entsprechenden Funktionen nutzen kann.

Name: Author

Description: This is the default role for authors of new articles.

## Permissions

Feature Category	Allowed Actions
Role Management	<input type="checkbox"/> define a new role (F-ROLE-01) <input type="checkbox"/> modify an existing role (F-ROLE-02) <input type="checkbox"/> remove an existing role (F-ROLE-03) <input type="checkbox"/> assign a role to user (F-ROLE-04) <input type="checkbox"/> revoke role from user (F-ROLE-05) <input checked="" type="checkbox"/> switch role(F-ROLE-06) <input type="checkbox"/> get a list of all existing roles(F-ROLE-07) <input checked="" type="checkbox"/> set user specific permissions(F-ROLE-08)
User Management	<input type="checkbox"/> create new user(F-USER-04) <input type="checkbox"/> remove user(F-USER-05) <input checked="" type="checkbox"/> modify user attributes (F-USER-06) <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> own</li> <li><input type="checkbox"/> all</li> </ul> <input type="checkbox"/> register new user(full profile)(F-USER-08) <input type="checkbox"/> register new user(empty profile)(F-USER-09) <input checked="" type="checkbox"/> rlogin(F-USER-10) <input type="checkbox"/> use preferences(F-USER-11) <input type="checkbox"/> get a list of all existing users (F-USER-12) <ul style="list-style-type: none"> <li><input type="checkbox"/> this journal</li> <li><input type="checkbox"/> all</li> </ul> <input type="checkbox"/> lock/unlock user account(F-USER-13) <input type="checkbox"/> user overview(F-USER-14)
	<input type="checkbox"/> submit a new article in representation of a specified author(F-ART-02) <input checked="" type="checkbox"/> submit a new own article (F-ART-03) <input checked="" type="checkbox"/> submit final version (F-ART-04) <input checked="" type="checkbox"/> read article (F-ART-05) <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> own</li> <li><input type="checkbox"/> delegated</li> <li><input type="checkbox"/> contributed</li> <li><input type="checkbox"/> all</li> </ul>

Abbildung 2.6: Bearbeiten einer Rolle

## 2.4 Funktionalität des OJS

### 2.4.1 Article Feature

Der Benutzer hat im "Article Feature" zwei Möglichkeiten:

- Er kann neue Artikel einreichen.
- Er kann die eingereichten Artikel einsehen. Sie sind in einer Tabelle dargestellt, die Informationen über die Artikel enthält, wie z.B: ArtikelID, Zustand des Artikels, Titel, Autor, Typ des Artikels und Aktionen. Die Aktionen hängen von den gesetzten Berechtigungen für die aktuelle aktive Rolle ab. Als Editorial Office kann der Benutzer einen Artikel lesen, entfernen, delegieren und eine Email an den Autor schreiben. Er kann auch die neuesten oder einige ältere Versionen zu einem bestimmten Artikel finden. Wenn der Artikel begutachtet wurde, können die Reports von den Autoren angesehen werden. Die Artikeltabelle kann nach ArtikelID, Titel, Autor und Typ sortiert werden. Durch das Anklicken des "ArticleAbstracts" werden alle Artikel kurz beschrieben. Die Artikeltabelle kann als ASCII Text exportiert werden.

### 2.4.2 Conflicts Feature

Mit Hilfe des `Conflicts Feature` werden Konflikte zwischen Artikel und Begutachter festgelegt. Ein Konflikt zwischen einem Begutachter und einem Artikel bedeutet, dass der Begutachter keine Information über die Auswertung des Artikels bekommen wird, d.h.:

- Er ist nicht in der Lage, den Status dieses Artikels einzusehen.
- Der Begutachter kann keinen Report über diesen Artikel lesen bis die Entscheidung vom Editorial Office getroffen wird.
- Der Begutachter kann nicht den Rang dieses Artikels innerhalb der Auswertungsmatrix sehen.
- Er darf keinen Bericht über diesen Artikel schreiben.

Als Beispiel: Wir nehmen an, ein Benutzer hat zwei Rollen, sowohl "Author" als auch "Reviewer" und hat einen Artikel eingereicht. In diesem Fall besteht eine persönliche Beziehung zwischen dem Benutzer als "Reviewer" und dem Artikel. Das Editorial Office muss dieses bei der Delegation des Artikels berücksichtigen.

### 2.4.3 Delegation Feature

Das Delegation Feature erlaubt dem Editorial Office die Begutachtungsaufgaben an den Reviewer zu delegieren. Das Editorial Office kann dem Reviewer zur Abgabe des Berichtes über den delegierten Artikel eine Frist geben. Diese Frist kann bei Bedarf verlängert werden, bzw. gelöscht werden. Mittels der Delegationliste kann der Begutachtungsprozess verfolgt und nachvollzogen werden.

### 2.4.4 Email Feature

Das Email Feature erlaubt dem Benutzer das Versenden von Emails komfortabler zu verwalten. Es besteht aus drei Unterfeatures:

- **Email Templates:** Der Benutzer kann sowohl die vorhandenen Email Templates editieren als auch neue Templates definieren. Der Service speichert sie für den Benutzer in der Datenbank.
- **Text Macros:** Dieses Unterfeature erlaubt dem Benutzer seine eigenen Textmakros zu definieren und die vordefinierten Textmakros zu benutzen.
- **Compose New Email:** Es gibt zwei Möglichkeiten eine neue Email zu schreiben: Entweder wird ein leeres Formular oder ein existierendes Email Template verwendet. Die Empfänger können auf einfacher Weise aus Mailingliste ausgewählt oder direkt eingegeben werden.

### 2.4.5 Evaluation Feature

Die Evaluationmatrix zeigt eine Übersicht über alle Reports, die von dem Reviewer-Komitee geschrieben wurden. Die Evaluationmatrix besteht aus mehreren Spalten. In der Spalte Scores werden die Bewertungsdaten der Reviewer, die an der Bewertung des betroffenen Artikels teilgenommen haben aufgelistet. In der Spalte Status wird der Zustand eines Artikels gezeigt, der mittels einer farblichen Markierung gekennzeichnet wird. Diese Farbmarkierung zeigt, in welchem Diskussionstatus sich der Artikel befindet. Das Editorial Office kann den Score sowie den Zustand für einen oder mehrere Artikel ändern, während die Reviewer nur das Leserecht haben. Die Evaluationmatrix bietet die Möglichkeit der Mehrfachselection und Modifikation der Artikel, sowie das Sortieren nach ArtikelID, Average und Type.

### 2.4.6 My Staff Feature

Das Staff Feature ermöglicht dem Benutzer seinen Mitgliederstab zu gestalten. Der Benutzer kann in der Liste, der zur Verfügung stehenden User so viele Benutzer auswählen

wie er möchte und zu seinem Mitgliederstab hinzufügen. Es können sowohl bereits angemeldete als auch neue Benutzer ausgewählt werden und zum Mitgliederstab hinzugefügt werden. Das `Staff Feature` erleichtert dem Benutzer die Delegation eines Artikels. Er kann eine Person oder mehrere Personen seines Mitgliederstabs auswählen, er muss nicht aus der Gesamtheit aller zugelassenen Dienstbenutzer auswählen. Die Mitgliederstabta-  
belle besteht aus den Namen der Mitglieder, ihrer Rolle, dem Datum, an dem der Benutzer diese Person zu seinem Mitgliederstab hinzufügte, einer kurzen persönlichen Anmerkung über den Mitarbeiter, seinen Kategorien und Aktionen. Die bereits existierenden Mitarbeiter des Stabes können entfernt oder es können ihnen neue Rollen zugewiesen werden.

### **2.4.7 Preferences Feature**

Mit Hilfe des `Preferences Feature` kann ein Benutzer, der über mehrere Rollen im OJS verfügt, eine bestimmte Rolle festlegen, die nach dem Einloggen im OJS aktiviert sein soll.

### **2.4.8 Profile Feature**

Das `Profile Feature` dient dem Benutzer zur Modifizierung seiner persönlichen Daten. Es gibt einige Daten, die eingegeben werden müssen : Email Adresse, Anrede, Vorname, Nachname, Straße, Postleitzahl, Stadt, Land, die Kennwortfrage und die Kennwortantwort. Andere Daten sind optional wie z.B: Titel, Präfix, Institut, Abteilung, Telefon- und Faxnummer. Der Benutzer kann sein Passwort ändern, indem er das alte Kennwort und das neue zweimal eingibt. Das Kennwort muss mindestens 5-stellig sein. Der Benutzer sollte seinen Namen als Passwort vermeiden. Der Service weist diese Art der Kennwörter aus Sicherheitsgründen zurück.

### **2.4.9 Reports Feature**

Das `Report Feature` zeigt dem Benutzer eine Liste von Final Reports und (oder) eine Liste von Reports, die er eingereicht hat und (oder) die ihm zugeteilt wurden. Bei der Verwaltung dieses Feature sind die folgenden Aktionen möglich: das Einreichen und das Löschen von Reports, das Lesen der eigenen bzw. aller (Final-) Reports, das Modifizieren der eigenen bzw. aller (Final-) Reports.

### **2.4.10 Roles Feature**

Das `Roles Feature` dient dazu, neue Rollen zu erstellen und existierende zu modifizieren. Die jetzigen Rollen im OJS sind Administrator, Author, Co-Author, Editor, Editorial Office und Reviewer. Jeder Benutzer kann eine oder mehrere Rollen besitzen. Jede Rolle ist durch verschiedene Rechte gekennzeichnet. Nur das Editorial Office

und der Administrator können eine Rolle erstellen, entfernen und ihre Permission modifizieren.

### 2.4.11 Setup Feature

Das Setup Feature besteht aus sechs Unterfeatures, mit denen der OJS konfiguriert werden kann.

#### 1. Look and Feel of the Journal Service

- **The Navigationbar:** Hier kann der Benutzer die Navigationsleiste konfigurieren.
- **The Content Page:** Es wird der Inhalt und das Aussehen der Seite spezifiziert.
- **The Sponsors:** Informationen über Sponsoren können eingetragen werden.
- **The Contact Addresses:** Dieses Unterfeature dient dazu, Kontaktadressen bekannt zu machen.
- **Preview of the Look & Feel:** Es wird die Konfiguration der Navigationsleiste, des Inhalts der Seite und der Sponsoren gezeigt.

#### 2. Papers and Reports

- **The article specific pages:** Mit Hilfe dieses Unterfeatures werden die unterschiedlichen Kategorien der Papiere, die im OJS existieren, spezifiziert.
- **The report specific pages:** Dieses Unterfeature bietet an, das Bewertungssystem zu spezifizieren. Es wird benutzt, um einen Artikel auszuwerten und um die Zahl der Begutachter, die mit dem Begutachtungsprozess jedes Artikels beschäftigt sind, festzulegen.

#### 3. Deadlines of the Journal

- **Setup the journal deadlines:** Es werden die Deadlines des OJS festgelegt. Zusätzlich kann die Reaktion des Services spezifiziert werden, wenn eine Deadline abgelaufen ist.

#### 4. Configuration of Roles

- **Reviewer:** Hier wird die Begutachterrolle konfiguriert, wie z.B.: Ob er den entsprechenden Autor und den Co-Autor eines Artikels sehen darf oder nicht. Ob er alle eingereichten Artikel oder nur die Artikel, die ihm zugewiesen wurden, lesen darf oder nicht.
- **Editorial Office:** Es kann konfiguriert werden, ob das Editorial Office auch Gutachten schreiben soll.

## 5. Setup of the Committee

- **Setup the Program Committee:** In der Tabelle wird das Programmkomitee des OJS spezifiziert.

## 6. Finish the Setup

Es wird die Konfiguration des Konferenzservices beschrieben.

- **Setup Modus:** Nach der Installation des Konferenzservices muss zuerst der Einstellungsmodus gestartet werden. Danach kann das Editorial Office die Konfiguration des Konferenzservices (z.B. Stichtage der Konferenz, Konfiguration von Rollen, Definition des Ausschusses etc.) konfigurieren. Der Einstellungsmodus wird beendet und es kann dann begonnen werden damit zu arbeiten. Um den Workflow zu prüfen sollte der Demo-Service verwendet werden. Der Original-Service enthält einen Link zum Demo-Service. Nur das Editorial Office hat die Erlaubnis sich in den Original-Service einzuloggen und die Einstellungskonfiguration zu ändern.
- **Test Modus:** Der Test-Modus enthält einen Link zum Original-Service. Der Demo-Service erlaubt dem Editorial Office und dem Programm-Ausschuß die Konferenz zu prüfen ohne den Original-Service zu beeinflussen, weil er auf eigenen unabhängigen Datenbanken arbeitet. Die Funktionalität und das Aussehen des Demo-Services sind im Vergleich zum Original-Service, bis auf die Einstellungseigenschaften, identisch.
- **Produktive Modus:** Wenn der Original-Konferenzservice im produktiven Modus begonnen wird, bekommen alle Mitglieder des Ausschusses automatisch E-mails, die die URL des Services und ein Kennwort für den ersten Zugang zum Service enthalten.

Man kann die Datenbanken des Original-Service in den Demo-Service kopieren um die Störungen zu lokalisieren. Es ist möglich die Datenbanken zu kopieren, ohne den Original-Service zu stoppen.

- **Stop and Resume the Conference Service:** Beim Stoppen des Journals muss mindestens eine der gezeigten Rollen ausgewählt werden. Dieses führt dazu, dass nur die Benutzer, die mindestens eine der vorgeählten Rollen besitzen, sich im OJS einloggen können. Ein Grund für das Stoppen des Services muss im entsprechenden Eingabefeld eingegeben werden. Diese Meldung wird auf der LOGIN-Seite angezeigt.
- **Copy the Database:** Die Einstellungseigenschaften des Original-Services hat eine zusätzliche Funktionalität um die Datenbanken vom Original-Service zum Demo-Service zu kopieren.

### 2.4.12 Tasklist Feature

Das `Tasklist Feature` zeigt dem User die Liste aller zu erledigenden Aufgaben, für die der eingeloggte User verantwortlich ist. Es stellt verschiedene Tasks zur Verfügung wie z.B. "accept/reject review task" , "delegate review" und "submit report"

Die Tasklistseite stellt die persönlichen Aufgaben eines OJS–Benutzers im `Journal Workflow` dar und bietet viele Informationen, die dem OJS–Benutzer beim Bewertungsprozess helfen.

### 2.4.13 Users Feature

Das Editorial Office ist für die Verwaltung aller OJS–Benutzer verantwortlich. Nach dem Anklicken des "Users" Button ist die "Services User" Seite einzusehen. Dort findet man einen Überblick über alle zugelassenen Dienstbenutzer. Die Benutzerliste ist alphabetisch sortiert. Das Anklicken von "All" oder eines Buchstabens des Nachnamens führt zu den entsprechenden Benutzern. Das Editorial Office kann auf zwei Arten neue Benutzer erstellen:

- User als Gast, d.h. ohne Profil.
- Eine komplette User Registration, d.h. mit einem kompletten Profil.

Das Editorial Office kann die Benutzerliste nach existierenden Rollen filtern sowie zu der ursprünglichen Liste zurückkehren. Das Editorial Office kann mehrere Aktionen durchführen. Es kann einem ausgewählten Benutzer neue Rollen zuweisen sowie bereits zugewiesene Rollen entziehen. Außerdem kann es einen Benutzer sperren, ein Benutzerprofil modifizieren, eine Benutzerkategorie modifizieren, einen Benutzer löschen, sich einen "useroverview" anschauen und die Permissions für einen Benutzer bewilligen oder widerrufen.

## 2.5 Technische Realisierung

Der OJS ist als eine komplexe, verteilte, mehrschichtige Webapplikation realisiert.

### 2.5.1 Serverstruktur

Bei der Realisierung des OJS wurde ein 3 Schichten Modell verwendet. Das System unterteilt sich in folgende Schichten oder layer (engl.):

- Benutzerschicht (frontend)
- Applikationsserverschicht (backend)
- Persistenzschicht (storagelayer)

Die **Benutzerschicht** dient im OJS zur Datenein-/ und ausgabe. Die Rolle der Benutzerschicht kann von jedem Webbrowser übernommen werden.

Die **Applikationsserverschicht** nimmt die Eingaben aus der **Benutzerschicht** entgegen und verarbeitet diese. Gegebenenfalls werden die Daten in der **Persistenzschicht** manipuliert und anschliessend wird eine Ausgabe generiert, die an die **Benutzerschicht** zurückgegeben wird.

Die **Persistenzschicht** ist ausschliesslich dafür da, um von den Benutzern getätigte Eingaben und Änderungen dauerhaft zu speichern. Da die von den Benutzern eingegebene Daten äusserst unterschiedlich sind, so ist auch die **Persistenzschicht** heterogen. Diese ist aus den folgenden Diensten zusammengesetzt, die im folgenden Absatz 2.5.2 erläutert werden:

- Datenbankserver
- Dateiversionierungssystem
- Newsserver
- Dateisystem

Die nachstehende Abbildung 2.7 zeigt die Zusammenarbeit der einzelnen Dienste untereinander, so wie den Zugriffsablauf von Client aus gesehen.

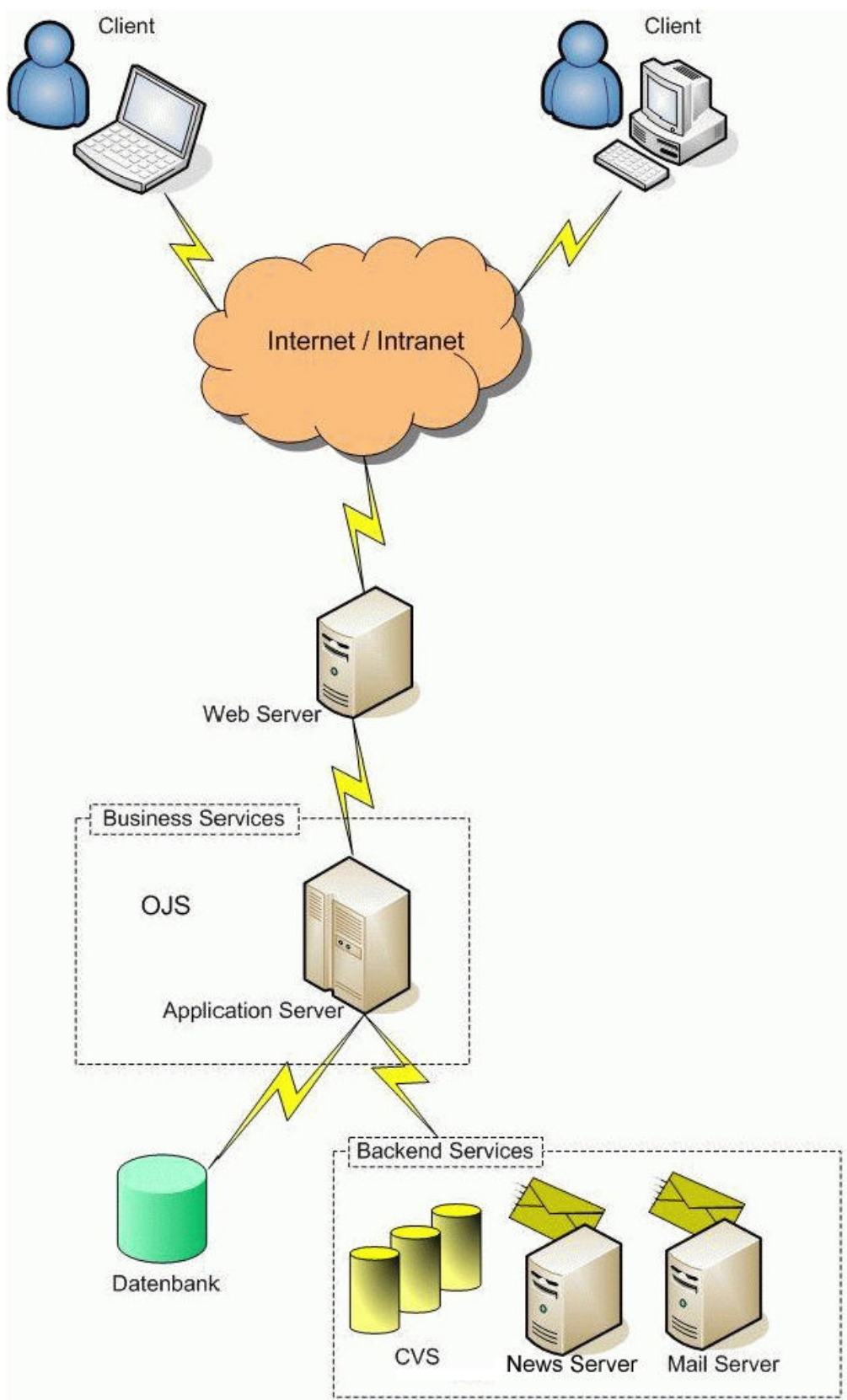


Abbildung 2.7: Dienstestruktur

## 2.5.2 verwendete Komponenten

Der OJS verwendet bei seiner Ausführung folgende Dienste bzw. Komponenten:

- Datenbankserver
- Dateiversionierungssystem (CVS)
- Newsserver
- E-Mail Server
- Dateisystem

Der *Datenbankserver* wird benutzt um alle Benutzer-, Rollen-, und Artikelmetadaten effizient abzuspeichern und dem Dienst schnell verfügbar zu machen. Dieses beinhaltet folgende Operationen, die abhängig von Datenart und Berechtigung des Benutzers möglich sind:

- Daten eintragen
- Daten ansehen
- Daten ändern
- Daten löschen

Das *Dateiversionierungssystem* wird verwendet um die Artikeldateien zu speichern und zu verwalten. Das benutzte Versionierungssystem CVS erlaubt folgende Operationen mit Dateien, abhängig vom Status des Benutzers:

- Dateien einchecken
- Dateien herunterladen
- Dateien löschen
- Dateien nach dem löschen wiederherstellen

Der *Newsserver* wird dazu benutzt um für einen Artikel eine Forumfunktionalität zu gewährleisten, damit ein Artikel zwischen mehreren Teilnehmern diskutiert werden kann.

Der *E-Mail Server* wird verwendet um eine Benachrichtigung von Benutzern über Vorgänge, die im System ablaufen, und die Benutzer betreffen, per E-Mail zu informieren.

Das *Dateisystem* wird benutzt um die Konfigurationsdateien für die **Applikationsserverschicht** abzuspeichern.

## Kapitel 3

# Entwicklungsumgebung ABC

Im nun folgenden Kapitel wird die Entwicklungsumgebung, das Agent Building Center (ABC), mit der der OJS entwickelt wurde, vorgestellt. Die Projektgruppe arbeitete ebenfalls mit dem ABC.

### 3.1 Grundlagen und Konzepte des ABC

Das Agent Building Center (ABC) ist eine Umgebung zum applikationsspezifischem Softwaredesign. Es unterstützt dabei eine programmierfreie Entwicklung von Applikationen. Applikationsentwickler sind also von Programmieraufgaben beim Entwickeln befreit. Die Applikationsentwicklung erfolgt über eine verhaltensorientierte Kombination von sogenannten Service Independent Building Blocks (SIBs). Diese SIBs sind Softwarekomponenten mit besonders einfachem Interface (Schnittstelle). Semantisch gesehen sind SIBs Einheiten die eine einzelne Aktion oder Aufgabe ausführen. SIBs sind leicht verständlich für Applikationsentwickler und bestehen gewöhnlich aus einer Zahl von klassischen Programmereinheiten (Prozeduren, Klassen, Module). Das ABC konzentriert sich auf das dynamische Verhalten von Applikationen. Komplexe Funktionalitäten werden graphisch aus den einzelnen SIBs zusammengesetzt. So entsteht ein Graph, in dem die SIBs die Knoten darstellen und die gerichteten Kanten den Kontrollfluß. Der Graph stellt also das Verhalten(Kontrollfluß) der Applikation dar. Dieser Graph ist unabhängig von Programmierparadigmen der zugrundeliegenden Programmiersprache. Die einzelnen SIBs werden implementiert unter Benutzung aller objektorientierten Features und nur ihre Kombination ist operational organisiert. Die graphische Kombination der SIBs passiert auf einer abstrakten Ebene und kann implementiert werden durch Techniken wie RMI, CORBA oder (D)COM. Die richtige Benutzung von SIBs wird durch Constraints kontrolliert. Sie sind eine Art Regelwerk zur richtigen Kombination der SIBs. Die Datenbank der Constraints wächst dynamisch mit der gewonnenen Erfahrung beim Benutzen der Umgebung (z. B. durch gefundene Errors). Sie sind intuitiv zu verstehen und wiederverwendbar. Die Arbeit der Applikationsentwicklung kann auf folgende Experten aufgeteilt werden.

- Programmierexperten sind verantwortlich für die Softwareinfrastruktur, die runtime-Umgebung, die kompilierten Services und die programmierten SIBs.
- Constraint Modelling Experten klassifizieren die SIBs. Gewöhnlich geschieht das nach technischen Kriterien wie ihrer Version, spezifischen Hardware und Software Anforderungen oder gemäß ihrem Zwecks bzgl. einem gegebenen Applikationsbereichs.
- Applikationsexperten entwickeln konkrete Applikationen durch die graphische Kombination von SIBs zu grob-granularen Flußgraphen (Service Logic Graph genannt). Diese Graphen können sofort durch eine Tracer ausgeführt werden, um das beabsichtigte Verhalten zu prüfen.
- End User können eine gegebene Applikation modifizieren, indem sie Forderungen durch Verhaltenspfade hinzufügen. Diese Forderungen werden dann mit dem gegebenen Programm verschmolzen, um so eine neue Applikation zu erhalten, die sich wie gefordert entlang der neu definierten Pfade verhält und überall sonst wie das ursprüngliche Programm.

Basierend auf Bibliotheken von SIBs und Constraints wird ein erstes Applikationsprogramm (der Prototyp) graphisch konstruiert. Dieser Prototyp wird durch Model Checking kontrolliert und darauf modifiziert. Der Applikationsexperte wählt einen Aspekt, generiert die entsprechende Sicht (view) die von allen irrelevanten Details abstrahiert und modifiziert die nötigen Stellen. Der Effekt der Modifikation kann automatisch zum zugrundeliegendem Applikationsprogramm durch einen Konkretisationsschritt transferiert werden. Dieser Schritt wird so lange iteriert bis alle relevanten Aspekte behandelt wurden. Durch den Model Checker werden Verletzungen der Constraints sofort entdeckt. Abbildung 3.1 stellt diesen Ablauf graphisch dar.

Laufende Prototypen können jederzeit getestet, kompiliert, ausgeführt und wenn erfolgreich, in einem Repository gespeichert werden. Das ABC separiert SIB Implementationen von ihren Beschreibungen. Für jede Applikationsbeschreibung gibt es ein gesondertes Meta-Data Repository, das eine abstrakte Beschreibung (logische Sicht) der SIBs enthält. Die SIB Implementation (physikalische Sicht) ist in einem anderen Repository. Applikationsexperten beginnen einen Prototypen zu konstruieren, sobald die abstrakte Beschreibung einer Domain verfügbar ist.

### **Service Logic Graph (SLG)**

Der SLG stellt die graphische Darstellung der Applikation dar. Wie jeder Graph besteht er aus Knoten, die durch Kanten verbunden sind. Die Knoten sind die einzelnen SIBs. Sie zusammen bilden die Funktionalität des Service. Die Kanten, die gerichtet sind, beschreiben den Kontrollfluß innerhalb des Services/der Applikation. Der Kontrollfluß beginnt an

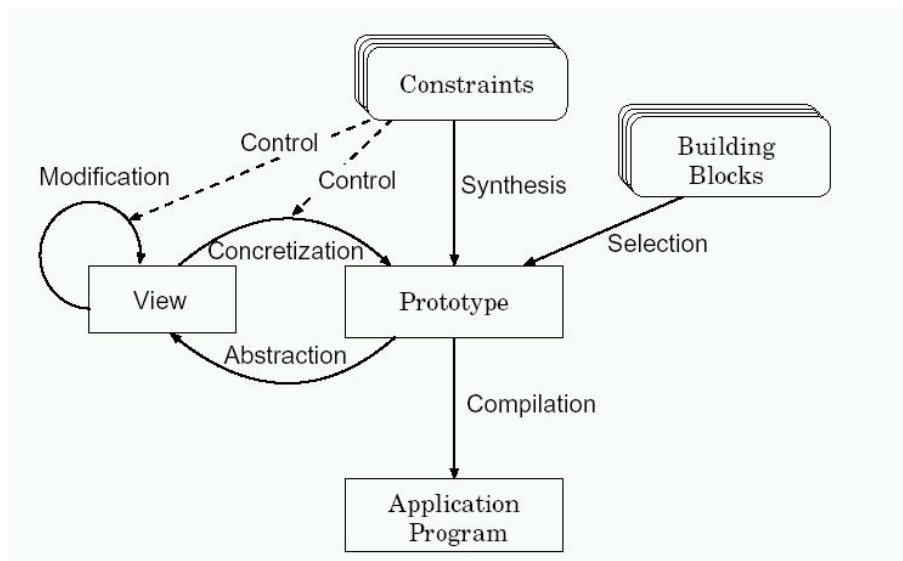


Abbildung 3.1: Lifecycle des ABC

einem ausgezeichneten Startknoten und durchläuft den Graphen bis hin zu einem Endknoten unter der Voraussetzung, dass kein Fehler auftritt. Abbildung 3.2 zeigt einen Service Logic Graphen.

### Service Independent Building Blocks (SIB)

SIBs stellen einzelne Aktionen dar. Dies bedeutet, dass sie einfach zu verstehen sind. Jedes SIB hat zwei Seiten. Eine logische und eine physikalische. Die logische Seite wird durch ihre Parameter (Eigenschaften), den Branches sowie ihrem Trace Code und Local Check Code repräsentiert. Die physikalische Seite beinhaltet den aktuell laufenden Code des SIB. Dieser kann in Java, C++ oder einer anderen Sprache geschrieben sein. Beim Konstruieren von Services (Service Logic Graph) ist die physikalische Seite verdeckt. Sie ist nur sichtbar, wenn eine laufende Applikation vom Service konstruiert wird. Innerhalb eines SIB sind vier Stücke Code, die die Interaktion mit anderen SIBs definieren, um den Service zu realisieren. Dies ist die SIB-Spezifikation, der red-line Code, der local check Code und die Implementation. Die SIB-Spezifikation ist die höchste Ebene der SIB Definition. Sie definiert vier Basisstücke von Informationen.

- Der Name des SIB identifiziert es in seiner graphischen Darstellung als Knoten und ebenso bei einem local oder global check.
- Die Klasse zu dem das SIB gehört. SIBs, die ähnliche Funktionen ausüben oder auf ähnlichen Daten operieren, werden in einer Klasse zusammengefaßt. SIBs einer Klasse werden durch das gleiche Icon repräsentiert.
- Die Parameter eines SIB. Sie sind der Schlüssel zur serviceunabhängigen Natur der SIBs. Sie werden benutzt um eine den Anforderungen des Service gerechten Instanz

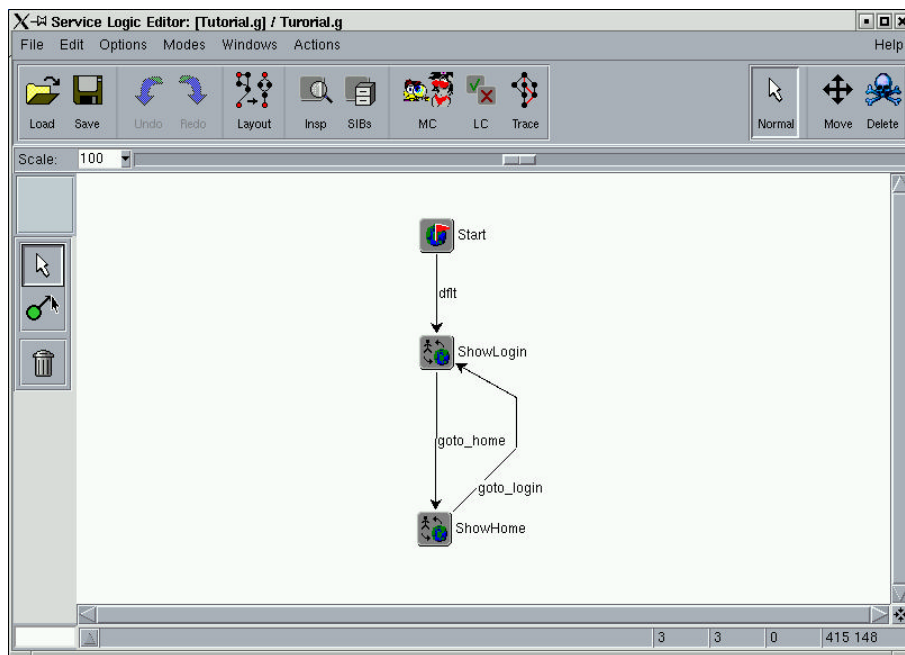


Abbildung 3.2: Ein Service Logic Graph

des SIBs innerhalb des SLGs zu erschaffen.

- Branches eines SIB repräsentieren verschiedene mögliche Ausgänge aus einem SIB. Ein SIB kann jede mögliche Anzahl von Branches haben.

Der red-line Code ist die logische Sicht auf das, was ein SIB aktuell tut. Er wird ausgeführt durch den Tracer beim Eintritt in den jeweiligen Knoten.

Der local check Code kontrolliert, ob das SIB richtig verwendet wird. Er wird ausgeführt, wenn der Service lokal gecheckt wird.

Die Implementation ist die physikalische Seite des SIB und besteht aus einer einzelnen Java oder C++ Klasse. Sie wird benutzt vom Compiler um den Service in eine laufende Applikation zu konvertieren.

Durch die Verbindung solcher SIBs wird der Workflow einer Applikation modelliert. Diese Verbindung geschieht durch gerichtete Kanten, die den Kontrollfluß durch den Graphen beschreiben. Die Branches eines SIB, die die möglichen Ausgänge aus einem SIB definieren, können den Kanten zugewiesen werden, so daß sie auf das logisch folgende SIB zeigen.

---

## 3.2 Makrofizieren

Ein Makro kapselt einen Teilgraphen. Dadurch wird die Wiederverwendung von SIB-Sequenzen sehr vereinfacht. Makros verleihen grossen Diensten eine bessere Übersicht und erleichtern damit Programmierern die Erweiterung, sowie das Suchen nach Fehlern. Im Graphen erscheint das Makro als Knoten mit eingehenden und ausgehenden Branches. Dieser Knoten läßt sich öffnen. Es erscheint der Teilgraph der als Makro zusammengefaßt wurde. In der Einzelkurzthemenphase machten wir von diesem Konzept Gebrauch um uns den Graphen des OJS übersichtlicher zu gestalten. Dieser besteht aus gut voneinander separierbaren Workflows. Die Projektgruppenleiter ordneten jedem Projektgruppenteilnehmer einen zusammenhängenden Teilgraphen zu. Dies erleichterte die weitere Arbeit am Graphen. So war es möglich parallel in kleinen Gruppen am OJS zu arbeiten. Jede Gruppe arbeitete separat an einem Makro ohne sich bei Ihrer Arbeit zu behindern.



# Kapitel 4

## Gruppenkurzthemen

Um sich in die verschiedenen Dienste und Programme einzuarbeiten, wurde die Gruppe in 6 Gruppen mit jeweils 2 Personen unterteilt. Für die Ausarbeitung dieser Kurzthemen hatte die PG 4 Wochen Zeit zur Verfügung.

### 4.1 Report Feature

#### 4.1.1 Beschreibung der Aufgabe

In der Aufgabe ging es darum, das `report feature` des Rollenmanagements um ein neues Auswahlkriterium `show list of reports` zu erweitern. Bisher gab es die Möglichkeit, einem Reviewer entweder eine Liste aller `reports` anzubieten (`all`), oder nur diejenigen `reports` die der Reviewer selber eingereicht hat.

Nun sollte das Feature dahingehend erweitert werden, dass es dem Reviewer erlaubt ist, eine Liste von Reports zu denjenigen Artikeln zu sehen zu denen er selbst einen Report verfasst hat.

Der zweite Teil der Aufgabe bestand darin, die Permission `read report` in gleicher Weise zu erweitern. Hierbei geht es um die Leserechte der Reports. Wieder sollte der Fall ergänzt werden, dass ein Reviewer alle Reports über die Artikel lesen darf zu denen er auch einen Report verfasst hat.

#### 4.1.2 Modellierungsphase

##### Use Case Diagramme

Abbildung 4.1 zeigt ein Use Case Diagramm aus der Sicht eines Editorial Office. Ein Benutzer dem die Rolle des Editorial Office zugewiesen ist, besitzt die Berechtigung das Rollenmanagement zu verändern.

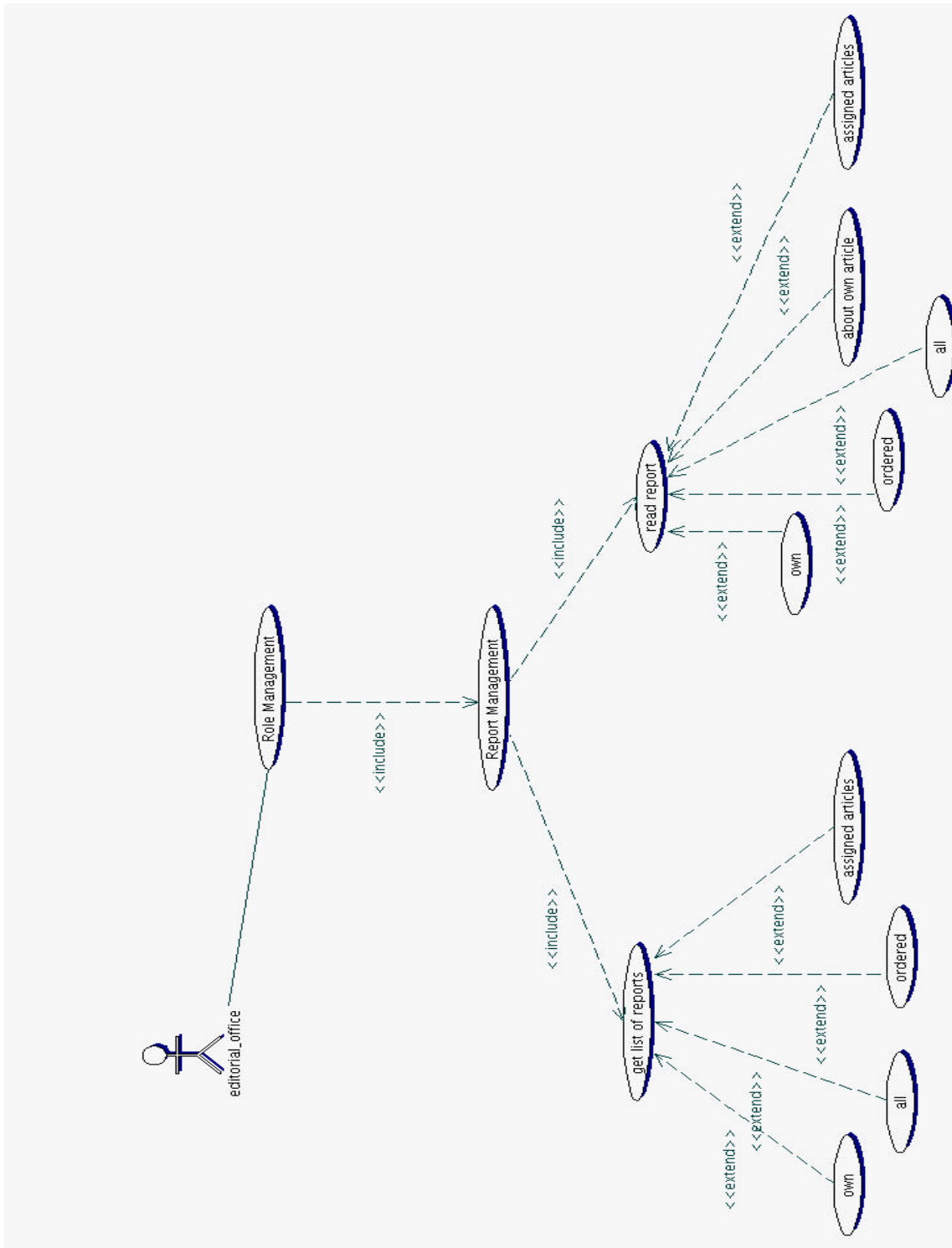


Abbildung 4.1: Use Case: Editorial Office

In diesem Fall interessiert nur der Bereich des report managements. Es sind 2 Auswahlmöglichkeiten gegeben: „get list of reports“ und „read reports“. Entscheidet man sich für das „get list of reports“ Feature, folgten ursprünglich 3 weitere Konfigurationsmöglichkeiten: „all“, „own“ und „ordered“. Im zweiten Fall kam noch eine vierte Möglichkeit dazu: „about own article“. Durch die Erweiterung hinzugekommen ist nun die neue Auswahl: „assigned articles“, die in beiden Fällen realisiert wurde.

Die Sicht eines Reviewers hängt von den Berechtigungen ab, welche das Editorial Office für einen Reviewer vorgesehen hat. In jedem Fall bekommt der Reviewer eine Liste von Reports angezeigt. Welche Einträge sich in dieser Liste befinden, regeln die Einstellungen im Rollenmanagement wie oben besprochen. Wenn ein Reviewer nun eine Liste von Reports sieht, hat er die Auswahl, ob er entweder einen Report lesen möchte, einen Artikel angezeigt bekommen möchte oder einen eigenen Report verändern will. Ob er die Erlaubnis hat, einen bestimmten Report zu lesen, hängt wiederum von den Einstellungen im Rollenmanagement ab. Abbildung 4.2 dient zur Verdeutlichung. Hinzugekommen ist hier wieder der Fall assigned articles.

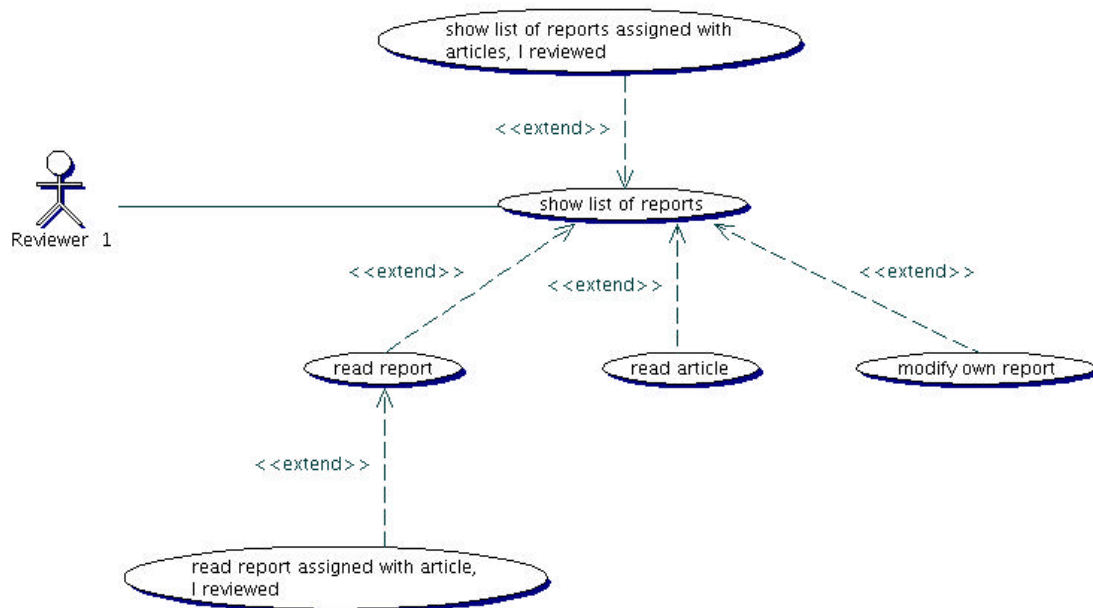


Abbildung 4.2: Use Case: Reviewer

### Aktivitätsdiagramme

Um die Eigenschaften einer Rolle ändern zu können, muss das Editorial Office die Seite Show Role betreten. Hier kann es dann zwischen den beiden (für unseren Fall relevanten) Features read reports und show list of reports wählen. Im ersten Fall hat es nun die Wahl ob es die folgenden Checkboxes selektiert oder nicht: „own“, „ordered“, „all“. Im zweiten Fall steht eine weitere Checkbox zur Auswahl: „about own article“. Im Soll

Zustand findet sich hier in jedem der beiden Fällen eine neue Checkbox: „assigned article“. Zur Verdeutlichung dient die Abbildung 4.3

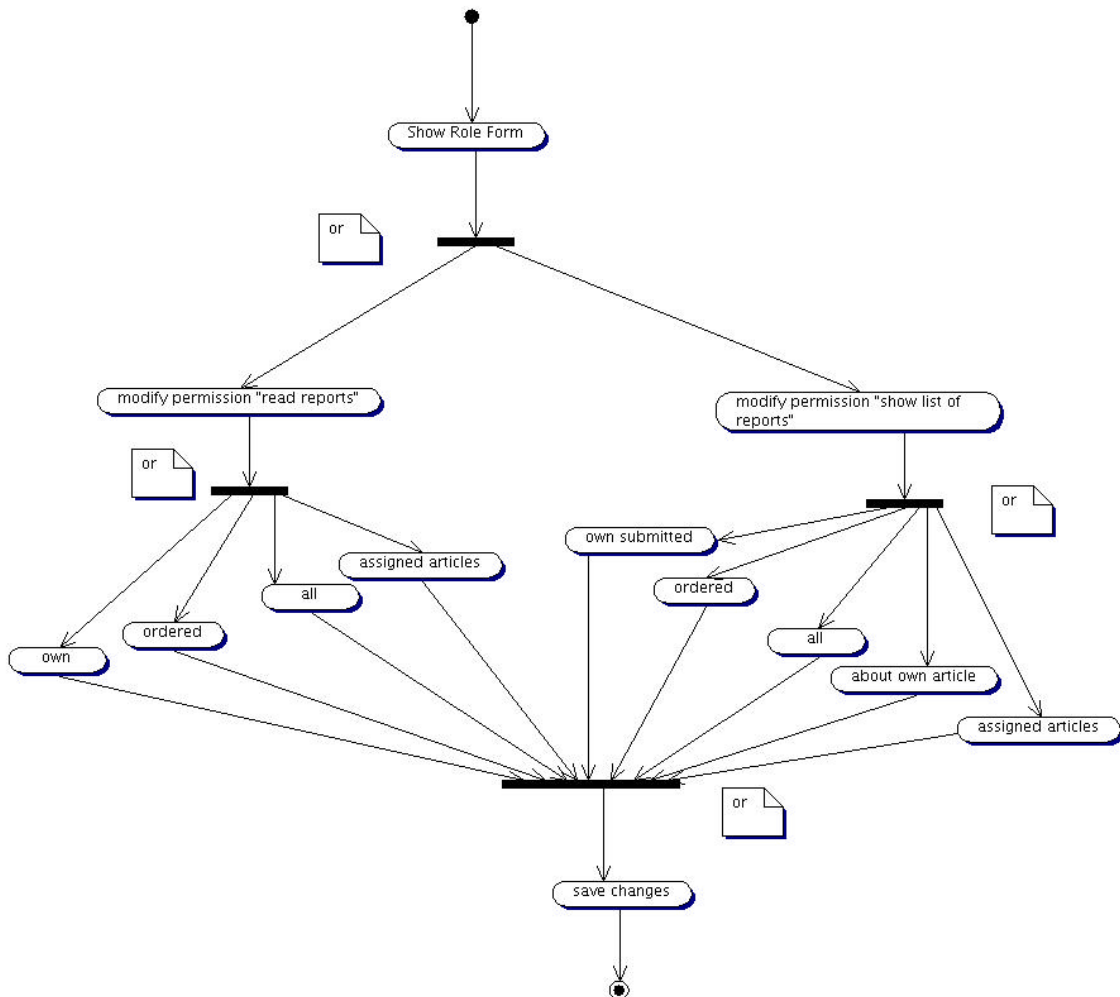


Abbildung 4.3: Activity Chart: Rolle modifizieren

Abbildung 4.4 zeigt ein weiteres Aktivitätsdiagramm. Wählt ein Benutzer das report feature aus der Navigationsleiste, so hängt das folgende Systemverhalten von der Rolle ab, die dem Benutzer zugeordnet ist. Handelt es sich bei der Rolle um die eines Editorial Office, so wird diesem eine Liste aller Reports angezeigt und es kann daraufhin entscheiden, wie sein weiteres Vorgehen aussehen wird:

- read report
- read article
- modify report

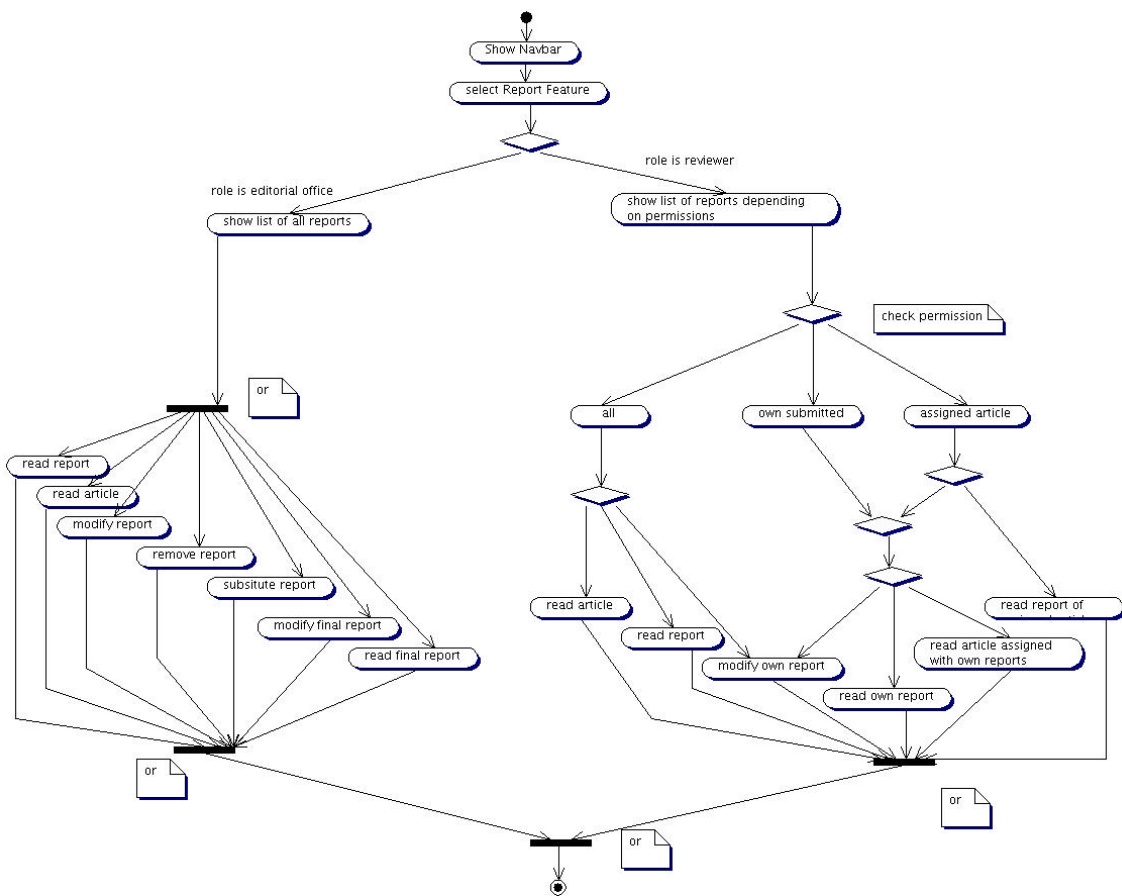


Abbildung 4.4: Activity Chart: Liste der Reports

- remove report
- substitute report
- modify final report
- read final report

Die Reportliste wird aus den Berechtigungen erzeugt die für die jeweilige Rolle eingestellt sind.

### 4.1.3 Implementierungsphase

Die Arbeitsgrundlage war der SIB ShowReportList im Report Makro. Zuerst wurde dieser den Konventionen entsprechend in drei SIBs aufgeteilt: GenerateReportList, ein CallContext2PreShow- und ein Show-SIB. Der neue SIB GenerateReportList enthielt die wesentliche Funktionalität des alten ShowReportList. (Siehe Abbildung 4.5)

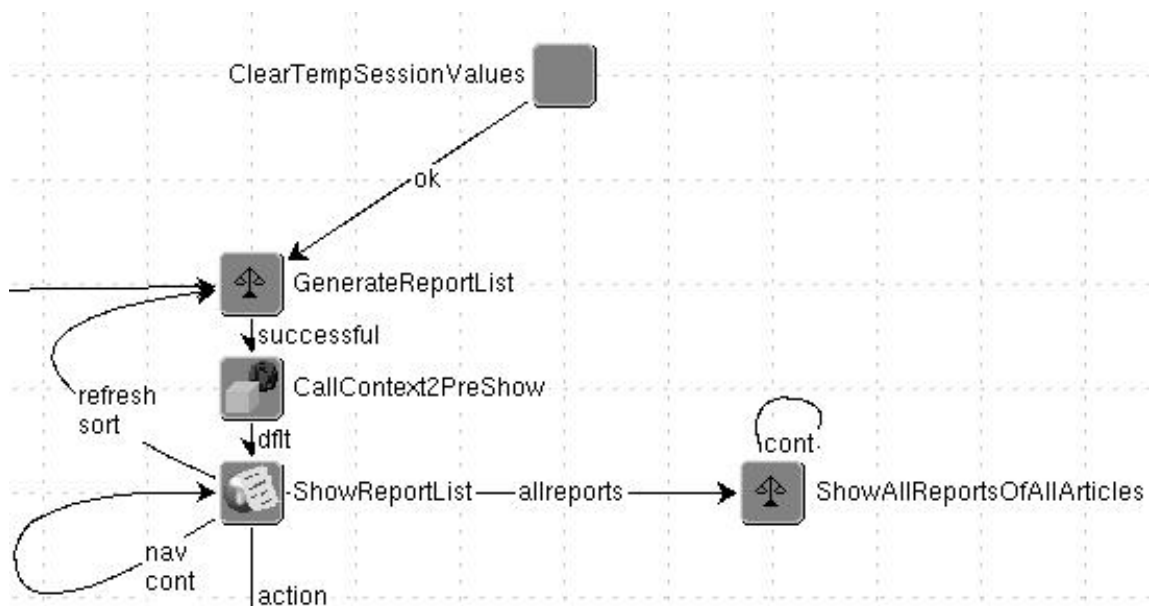


Abbildung 4.5: Ausschnitt aus dem ABC

Danach wurde er um den neuen Filter erweitert.

### Der Workflow

GenerateReportList schreibt eine Liste von Reports in den Call Context, abhängig von den Rechten des aktuellen Benutzers. Diese Liste enthält dann alle Reports, die der

Benutzer sehen darf. Der `CallContext2PreShow-SIB` kopiert diese Liste in den `PreShow`, von wo sie zum Füllen der Seite `ShowReportList` genutzt wird.

`GenerateReportList` besitzt eine Reihe von Parametern, die angeben, unter welchem Key die Daten im Call Context abgelegt werden sollen. Die interessantesten sind dabei `reportList` und `finalReportList`, die angeben, wo die Liste der anzuzeigenden Reports bzw. Final Reports abgelegt werden soll. Außerdem besitzt `GenerateReportList` zwei Branches: „successful“ und „failed“. „successful“ führt normalerweise zu einem `CallContext2PreShow-SIB`, um die Daten aufzubereiten und anzuzeigen. „failed“ kann genutzt werden, um eine Fehlerseite anzuzeigen.

#### 4.1.4 Testphase

Zuerst wurde der Dienst mit Demodaten gefüllt, also mit verschiedenen Reviewern und Artikeln. Dann wurden die Artikel an verschiedene Reviewer delegiert, wobei darauf geachtet wurde, dass nicht jeder zu jedem Artikel einen Report schrieb - denn ein solcher Reviewer darf auch im neuen Modus „of assigned articles“ alle Artikel sehen.

Danach änderte das Editorial Office für die Rolle Reviewer die Rechte „get list of reports“ und „read report“ auf „of assigned articles“. (Siehe Abbildung 4.6)

Als Reviewer eingeloggt wurde folgende Liste der Reports ausgegeben(siehe Abbildung 4.7).

Als das Editorial Office ein zweites Mal die Rechte änderte und „get list of reports“ auf „all“ setzte, fiel die Liste wesentlich länger aus (siehe Abbildung 4.8).

Da „read report“ jedoch noch auf „of assigned articles“ eingestellt war, durfte nicht jeder Artikel angezeigt werden, wenn auf „read article“ geklickt wurde. Wie erwartet wurde der Zugriff auf alle Artikel verwehrt, für die der Reviewer nicht selbst einen Review geschrieben hatte. Es erscheint:

```
You are not permitted to read this Report!
```

<b>Report Management</b>	<input checked="" type="checkbox"/> submit report(F-REV-01)
	<input checked="" type="checkbox"/> read report (F-REV-02)
	<input type="checkbox"/> own
	<input type="checkbox"/> ordered
	<input type="checkbox"/> all
	<input checked="" type="checkbox"/> assigned
	<input type="radio"/> modify report (F-REV-03)
	<input checked="" type="radio"/> own
	<input type="radio"/> all
	<input type="checkbox"/> compose final report(F-REV-04)
	<input type="checkbox"/> read final report (F-REV-05)
	<input type="checkbox"/> own
	<input type="checkbox"/> ordered
	<input type="checkbox"/> all
<input type="checkbox"/> about own article	
<input type="checkbox"/> about contributed article	
<input checked="" type="checkbox"/> get list of reports (F-REV-06)	
<input type="checkbox"/> own submitted	
<input type="checkbox"/> ordered	
<input type="checkbox"/> all	
<input type="checkbox"/> about own article	
<input checked="" type="checkbox"/> of assigned articles	
<input type="checkbox"/> modify final report (F-REV-07)	
<input type="checkbox"/> own	
<input type="checkbox"/> all	
<input type="checkbox"/> remove report(F-REV-09)	
<input type="radio"/> read reports (F-REV-10)	
<input checked="" type="radio"/> directly	
<input type="radio"/> if all reports about an article are submitted	
<input type="radio"/> if all reports of all articles are submitted	
<input type="checkbox"/> submit a report in representation of a specified reviewer(F-REV-12)	

Abbildung 4.6: Rollen Management - Report Feature

## List of Reports

To get an overview of all reports click here: [all reports](#)

**Refresh**

(last refresh: *Wed Nov 24 14:12:01 CET 2004*)

ID	Referee	Subreviewer	Score	Confidence	Date	Action
<a href="#">001-R1</a>	The Gray, Gandalf		0	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">001-R2</a>	Brink, Dominik		2	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">001-R3</a>	Simpson, Bart		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">003-R1</a>	The Gray, Gandalf		0	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">003-R2</a>	Brink, Dominik		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">003-R3</a>	Simpson, Bart		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">007-R1</a>	Brink, Dominik		3	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">007-R2</a>	The Gray, Gandalf		0	2	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>

Abbildung 4.7: Liste verbundener Reports

## List of Reports

To get an overview of all reports click here: [all reports](#)

**Refresh**

(last refresh: *Wed Nov 24 14:09:25 CET 2004*)

ID	Referee	Subreviewer	Score	Confidence	Date	Action
<a href="#">001-R1</a>	The Gray, Gandalf		0	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">001-R2</a>	Brink, Dominik		2	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">001-R3</a>	Simpson, Bart		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">003-R1</a>	The Gray, Gandalf		0	3	2004-10-28	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">003-R2</a>	Brink, Dominik		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">003-R3</a>	Simpson, Bart		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">004-R1</a>	Simpson, Bart		0	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">004-R2</a>	The Gray, Gandalf		0	3	2004-11-24	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">006-R1</a>	The Gray, Gandalf		4	2	2004-11-03	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">007-R1</a>	Brink, Dominik		3	3	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a>
<a href="#">007-R2</a>	The Gray, Gandalf		0	2	2004-11-10	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">008-R1</a>	Simpson, Bart		0	3	2004-11-11	<a href="#">read article</a> • <a href="#">read report</a>
<a href="#">008-R2</a>	The Gray, Gandalf		0	3	2004-11-11	<a href="#">read article</a> • <a href="#">read report</a>

Abbildung 4.8: Liste aller Reports

## 4.2 Article Feature

### 4.2.1 Beschreibung der Aufgabe

Der Dienst sendet automatisch generierte Emails beim Auftreten bestimmter Ereignisse. Die Email, die an den Reviewer gesendet wird, hat nicht genug Informationen. Das Ziel der Arbeit war es, diese Email zu erweitern. Sie soll zusätzlich eine Inhaltsangabe des eingereichten Artikels und den Namen des Autors enthalten, wobei das System abhängig davon, ob der Reviewer das Feature `view metadata anonym` (F-ART-15) hat, entscheidet, ob der Name des Autors in der Email angegeben wird oder nicht.

### 4.2.2 Modellierungsphase

In der Modellierungsphase wird der momentane Workflow analysiert, um festzustellen, wie er zur Lösung der Aufgabe modifiziert werden muss. Siehe dazu 4.9 und 4.10

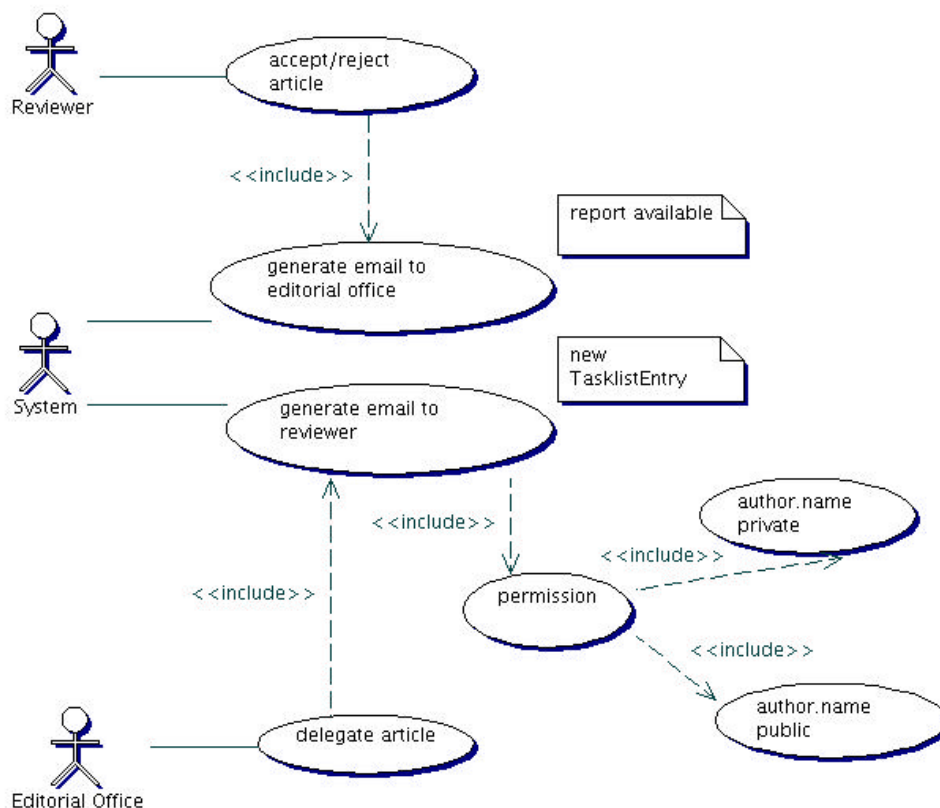


Abbildung 4.9: Use case Diagramm: Ist-Zustand.

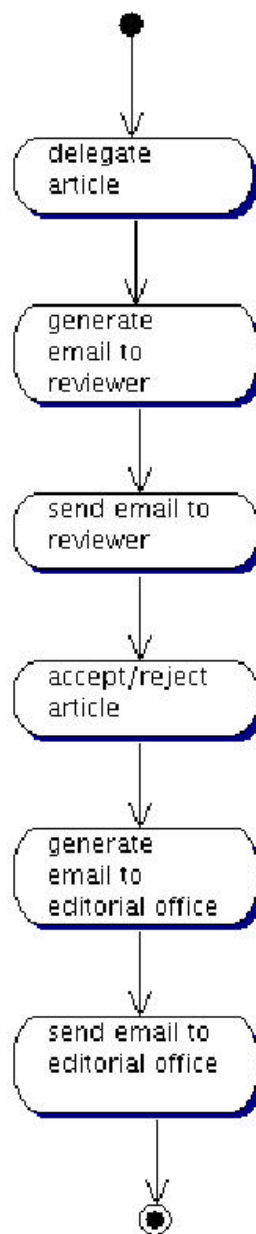


Abbildung 4.10: Aktivitätsdiagramm: Ist-Zustand.

Dann wird der Soll-Zustand modelliert. Anhand dieser Modellierung soll ersichtlich werden was zur Lösung der Aufgabe zu machen ist. Siehe dazu Abbildung 4.11 und 4.12.

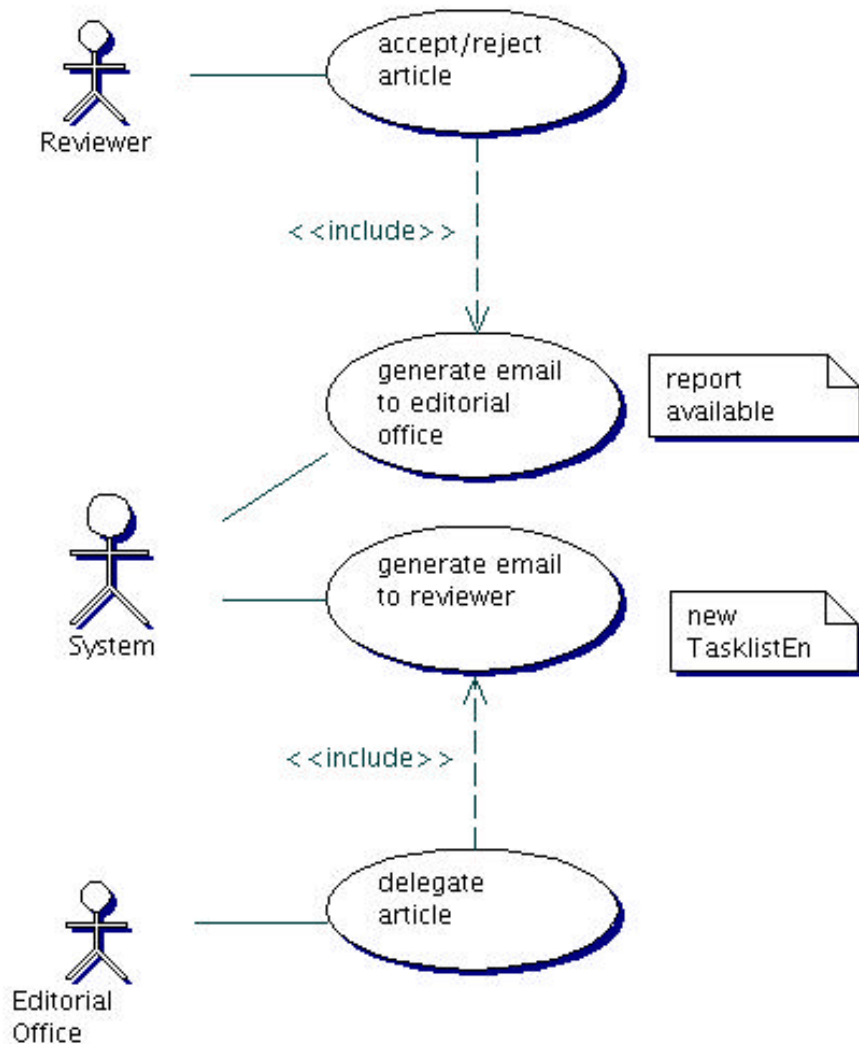


Abbildung 4.11: Use case Diagramm: Soll-Zustand mit Feature view metadata anonym (F-ART-15)

Vor der Lösung der Aufgabe waren zu wenig Informationen in der Email enthalten, die ein Reviewer vom System erhält, sobald ein neuer Artikel eingereicht wurde. Die Email an den Reviewer wurde um den Namen des Autors und einer Inhaltsangabe des Artikels erweitert. Für die Angabe des Namens des Autors gilt, dass dieser nur unter Berücksichtigung des Features view metadata anonym (F-ART-15) in der Email enthalten sein soll. Es war

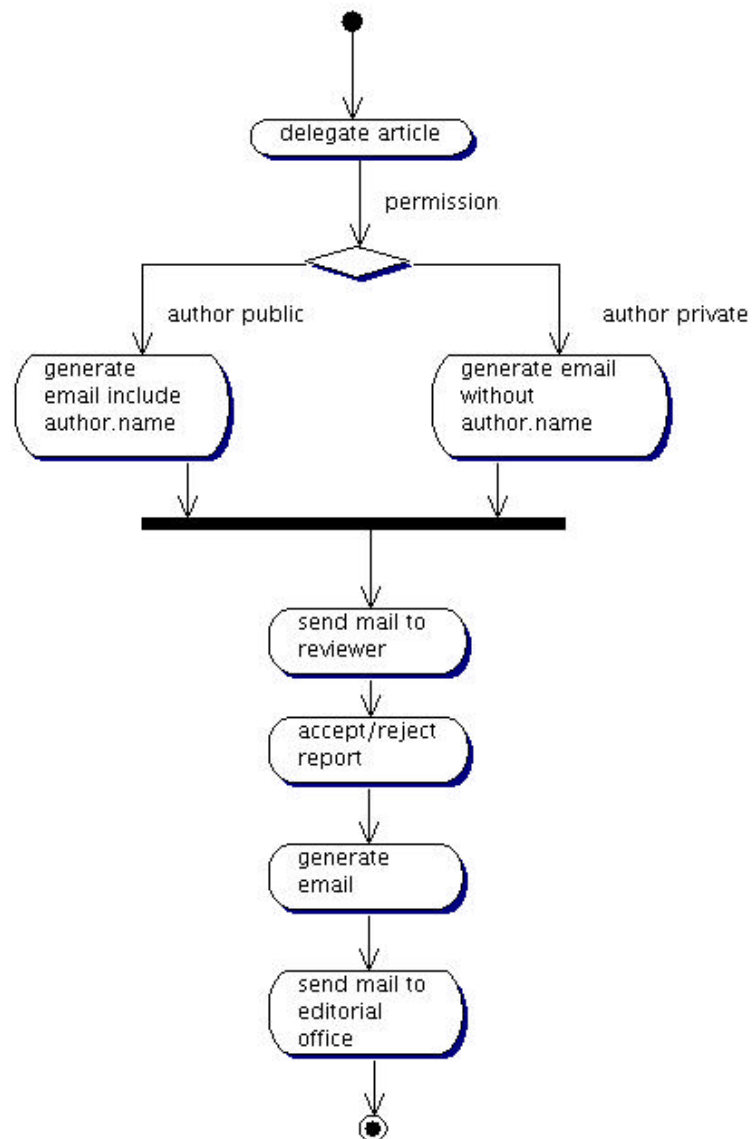


Abbildung 4.12: Aktivitätsdiagramm: Soll-Zustand mit Feature view metadata anonym (F-ART-15)

also zusätzlich die Abfrage des Features in den bestehenden Workflow zu integrieren. Siehe Abbildung 4.12.

### 4.2.3 Implementierungsphase

Zur Lösung der Aufgabe muss das SIB `PrepareTasklistEntryMailParameter` modifiziert werden, sowie das Email Template `new_tasklist_entry.txt`. Das SIB `PrepareTasklistEntryMailParameter` ist zuständig für das Generieren der Email, die der Reviewer erhält. Es stellt die Parameter bereit die die Email an den Reviewer enthalten soll. Das Email-Template `new_tasklist_entry.txt` legt fest, wie die Email aussehen soll. Dieses SIB kontrolliert, ob der Name des Autors angegeben werden darf oder nicht. Das neue Feature `view metadata anonym (F-ART-15)` musste also hier eingepflegt werden.

### 4.2.4 Testphase

Um die Lösung der Aufgabe zu testen, wurde ein Artikel eingereicht und einmal das Feature `view metadata anonym (F-ART-15)` gesetzt und einmal nicht. Abbildung 4.13 und Abbildung 4.14 zeigen, dass die Aufgabe korrekt gelöst wurde, da die dementsprechenden Emails verschickt wurden.



Abbildung 4.13: Email ohne Autor

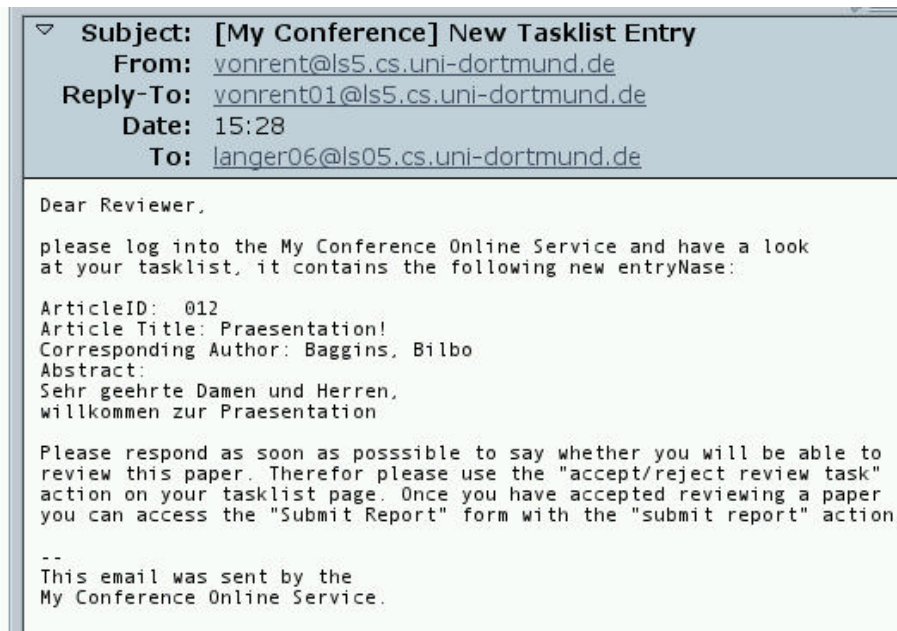


Abbildung 4.14: Email mit Autor

## 4.3 Delegation Feature

### 4.3.1 Beschreibung der Aufgabe

Das Kurzthema beschäftigt sich mit folgenden Aufgaben:

1. Die Delegation Matrix muss um zwei Spalten „Comments“ und „Type“ erweitert werden und die gespeicherten Kommentare der Delegierungen und die Artikelkategorien in den entsprechenden Tabellenzellen anzeigen.
2. Die Funktionalitäten „Extend Decision Deadline“ und „Extend Task Deadline“ müssen so modifiziert werden, dass sie getrennte Eingabefelder für „Tag“, „Monat“ und „Jahr“ der Frist unterstützen. Die Benutzereingabe wird auf ihre Gültigkeit hin überprüft. Nach der erfolgreichen Verifikation werden die Daten gespeichert. Anderenfalls muss die entsprechende Fehlermeldung dem User angezeigt werden.

Um den ersten Teil der Aufgabe zu lösen, waren die Änderungen im Source-Code des ShowDelegatinMatrix SIBs unnötig, weil alle notwendigen Daten für jedes ArticleDelegation Objekt aus der Liste von Delegierungen auf der delegationmatrix-cont.wml Seite schon vorhanden sind.

Die erste Aufgabe verursacht keine Workflow- und SIB-Codeänderungen, nur die Anpassungen auf der WML-Präsentationsebene.

Um das zweite Ziel zu erreichen, musste der Workflow der bestehenden Features „Extend Decision Deadline“ und „Extend Task Deadline“ angepasst werden. Die erforderliche Funktionalität wurde durch die Verwendung von sowohl schon vorhandenen als auch neu implementierten SIB-Blöcken erreicht. Das Design und die Implementierung von oben beschriebener Lösung werden nun näher betrachtet.

### 4.3.2 Modellierungsphase

In diesem Abschnitt werden die Aspekte vom UML – Design der Features „Extend Decision Deadline“ und „Extend Task Deadline“ dargestellt. Weil Beide im System bereits existierten und noch weiterentwickelt wurden, enthalten folgende Kapitel die Beschreibung sowohl vom Anfangszustand als auch vom Zielzustand.

#### Anfangszustand

Die obengenannte Features sind im „Delegation Feature“ eingebaut und als Aktionen in der Delegation Matrix dargestellt.

Das Use Case Diagramm in der Abbildung 4.15 beschreibt unterschiedliche Aktionen, die für jede Delegierung in Abhängigkeit ihres Status durchgeführt werden können.

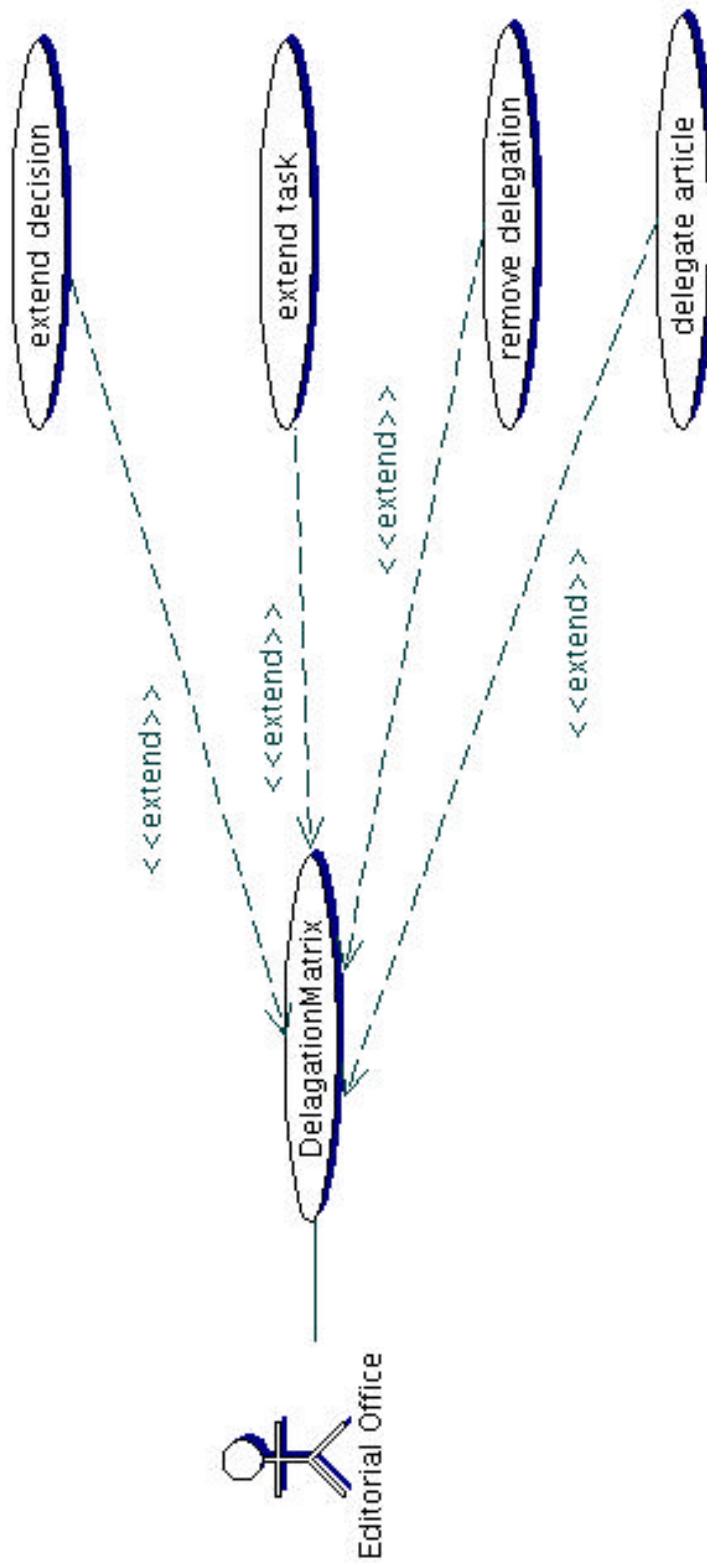


Abbildung 4.15: Use Case „Extend Decision/Task Deadline“ – Anfangszustand

- **Delegation Matrix** – gibt die Übersicht der delegierten Reviews und die Information über „ArticleID“, „Reviewer“, „Decision Deadline“, „Task Deadline“, „Comments“, „Type“ und Bearbeitungsstand der Review;
- **Extend Decision** – das Feature ermöglicht dem Editorial Office die Frist zu ändern, während ein Gutachter den ihm delegierten Reporttask entweder akzeptieren oder ablehnen muss;
- **Extend Task** – dieses Feature gibt dem Editorial Office die Möglichkeit das Abgabedatum des Reports zu ändern;
- **Remove Delegation** – erlaubt den Reporttask aus der Liste zu entfernen;
- **Delegate Article** – ermöglicht den zur Zeit ausgewählten Artikel an die noch nicht beteiligten Reviewers zu delegieren.

Das Aktivitätsdiagramm in der Abbildung 4.16 zeigt den Datendurchlauf nach dem Anklicken des Links „extend task“ bzw. „extend decision“.

### Zielzustand

Hier sind die entsprechend der Aufgabe vorgenommenen Änderungen des Workflows dargestellt.

Das Aktivitätsdiagramm in der Abbildung 4.17 zeigt den modifizierten Datendurchlauf nach dem Anklicken des Links „extend task“ bzw. „extend decision“. Die Änderungen werden weiter im nächsten Unterkapitel detailliert beschrieben.

### 4.3.3 Implementierungsphase

Der Anfangsworkflow enthielt einen SIB, der zum Lesen und Anzeigen der Daten aus der Datenbank diente (ShowDecision-/TaskDeadline). Die SIBs CheckStoreDecision-/TaskDeadline führten nur die einfachste Plausibilitätsprüfungen der Datumseingaben durch (siehe Abbildung 4.18).

Der neue Workflow enthält folgende SIBs (siehe Abbildung 4.19):

- `GenerateDecisionData` und `GenerateTaskData` erhalten den Parameter „DelegationKey“ aus dem SIB `Request2CallContext`. Anhand dieses Schlüssels werden die notwendigen Daten aus der Datenbank gelesen. Die Daten werden mittels des SIBs `CallContext2PreShow` zum Anzeigen vorbereiten;
- `ShowDecisionDeadline` und `ShowTaskDeadline` lesen Daten aus der Map „Content“, welche mittels dem SIB `CallContext2PreShow` hineingeschrieben wurden und zeigen sie in einer HTML-Seite an;

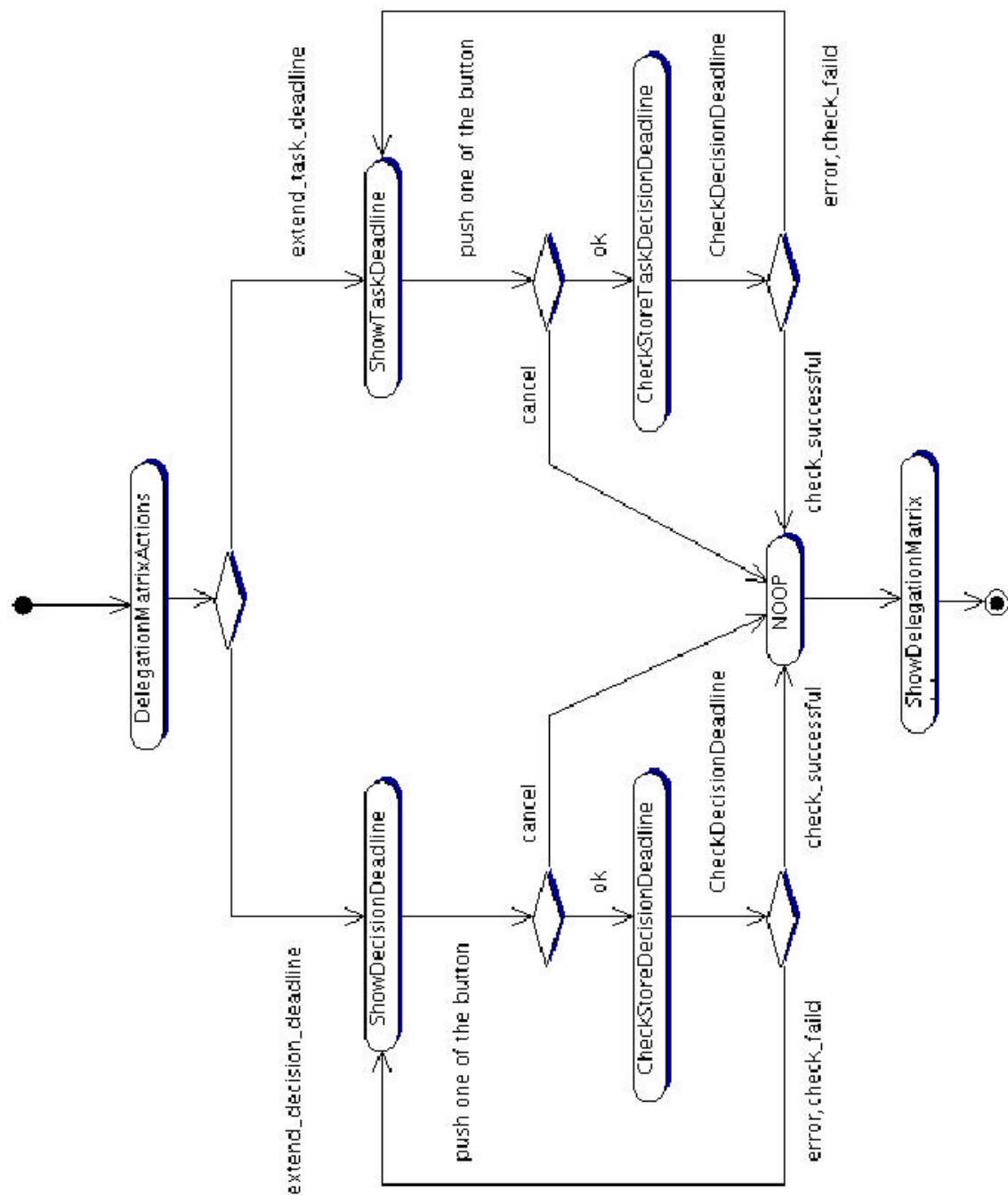


Abbildung 4.16: Activity Chart „Extend Decision/Task Deadline“ – Anfangszustand

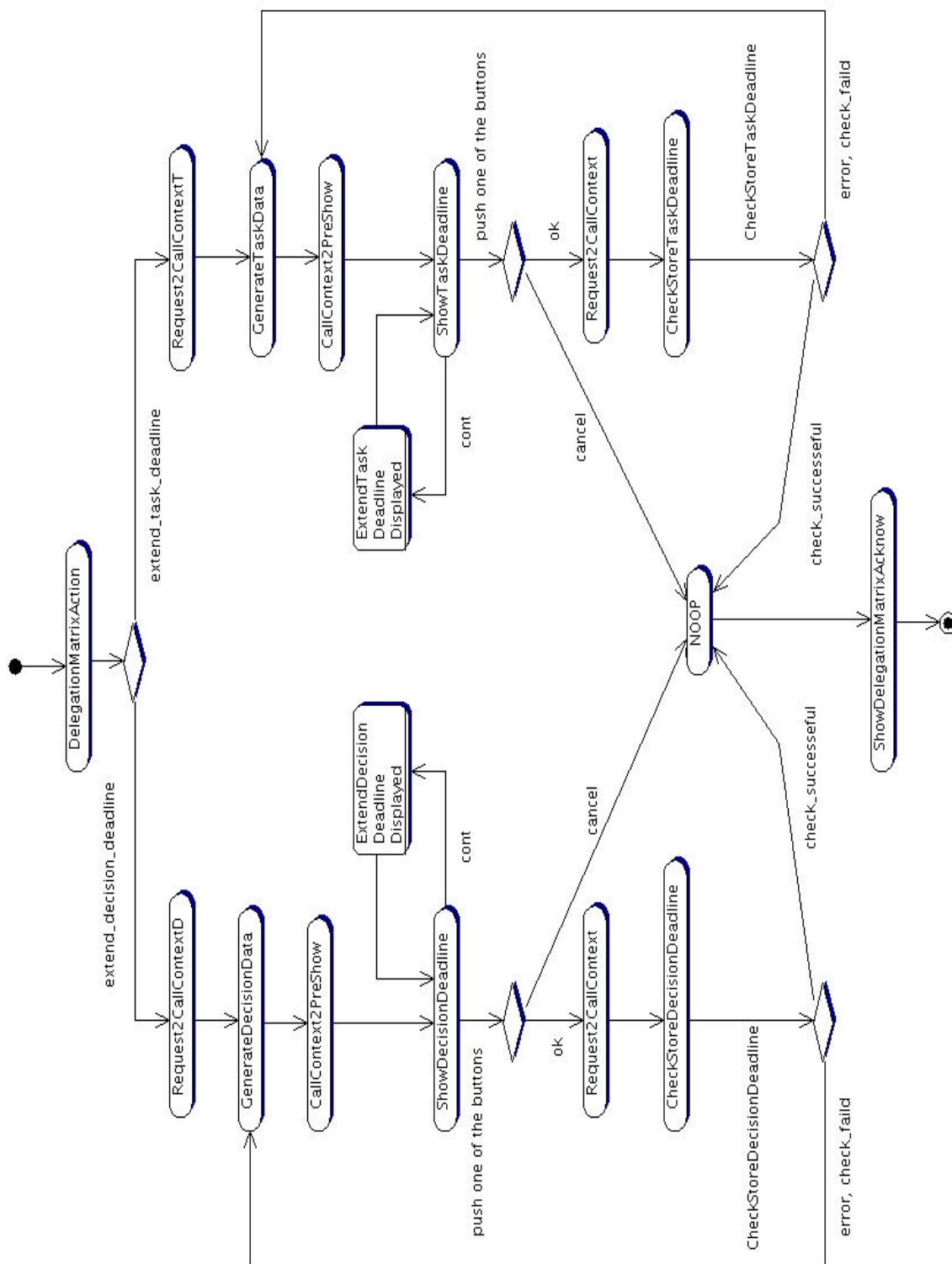


Abbildung 4.17: Activity Chart „Extend Decision/Task Deadline“ – Zielzustand

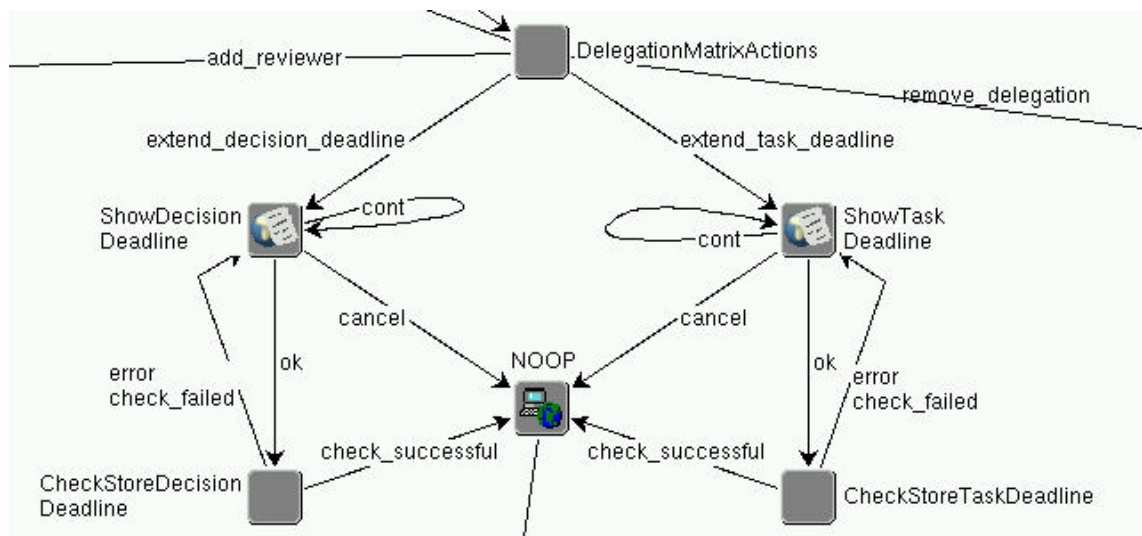


Abbildung 4.18: Workflow „Extend Decision/Task Deadline“ – Anfangszustand

- CheckStoreDecisionDeadline und CheckStoreTaskDeadline lesen eingegangene Daten aus dem Call Context, die vom SIB Request2CallContext hineingeschrieben wurden und überprüfen ihre Gültigkeit. Wenn die Daten korrekt sind, werden sie in der Datenbank gespeichert und es wird zu dem SIB NOOP (check\_successful Kante) gesprungen. Wenn die Daten falsch sind, wird eine Nachricht über den entsprechenden Fehler (error und check\_failed Kanten) angezeigt, indem zurück auf die vorherige HTML-Seite gegangen wird, welche die eingegebenen Daten und eine Fehlermeldung anzeigt.

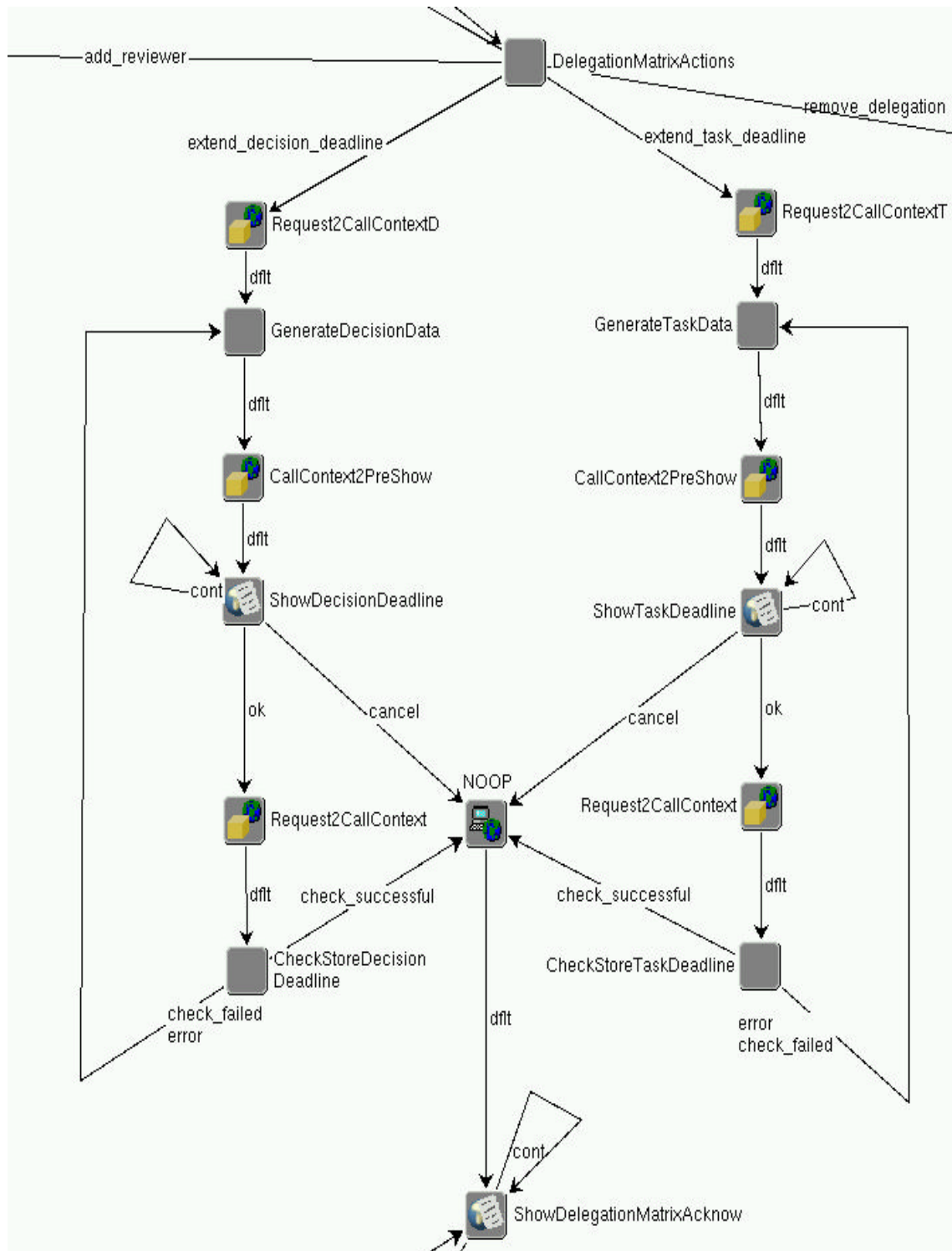


Abbildung 4.19: Workflow „Extend Decision/Task Deadline“ – Zielzustand

### 4.3.4 Testphase

Durch Anklicken des Links „extend task“ bzw. „extend decision“ (Abbildung 4.20) gelangt man in einer Seite (Abbildung 4.21), wo es möglich ist aktuelles Termindatum einzusehen und einen neuen Termin eingeben zu können. Jetzt wird es ausgetestet, ob man wirklich nur zulässiges Datum eingeben kann. Die zum Datum gehörende Eingabefelder werden mit inkorrekten Werten ausgefüllt (siehe Abbildung 4.22) und dann wird auf „Save“ geklickt. Die Abbildung 4.23 zeigt das Ergebnis an.

Mehrfach ausgeführte Testdurchläufe des o.b. Workflows haben gezeigt, dass die Implementierung korrekt ist.

## Overview of Delegated Reviews

This pages gives an overview of the delegated reviews, i.e. when and to whom you have delegated a review of an article. So you can get the in process.

The legend at the bottom of this page explains the content of each column.

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

[Refresh](#) (last refresh: *Tue Apr 12 13:16:48 CEST 2005*)

ID	Delegated		Status(Since)	Decision Deadline	Task Deadline	Action
	from	to				
<a href="#">003</a>	<a href="#">aa</a>	<a href="#">C C</a>	done, report submitted(on 2005-02-16)			
<a href="#">003</a>	<a href="#">aa</a>	<a href="#">D D</a>	done, report submitted(on 2005-02-16)			
<a href="#">003</a>	<a href="#">aa</a>	<a href="#">b b</a>	done, report submitted(on 2005-02-16)			
<a href="#">004</a>	<a href="#">aa</a>	<a href="#">D D</a>	done, report submitted(on 2005-02-16)			
<a href="#">004</a>	<a href="#">aa</a>	<a href="#">b b</a>	done, report submitted(on 2005-02-16)			
<a href="#">005</a>	<a href="#">aa</a>	<a href="#">C C</a>	done, report submitted(on 2005-02-16)			
<a href="#">005</a>	<a href="#">aa</a>	<a href="#">D D</a>	done, report submitted(on 2005-02-16)			
<a href="#">006</a>	<a href="#">aa</a>	<a href="#">D D</a>	waiting for decision(since 2005-03-24)	2005-04-30	2005-05-01	<a href="#">extend decision</a> • <a href="#">extend task</a> • <a href="#">delegate article</a> <a href="#">remove delegation</a>

Abbildung 4.20: Delegation Matrix

Current Decision Deadline (dd.mm.yyyy)	New Decision Deadline (dd.mm.yyyy)
30.04.2005	30 04 2005
<input type="button" value="Save"/>	<input type="button" value="Clear"/> <input type="button" value="Cancel"/>

Abbildung 4.21: Eingabeformular – Extend Decision Deadline

Current Decision Deadline (dd.mm.yyyy)	New Decision Deadline (dd.mm.yyyy)
30.04.2005	35 8a -005
<input type="button" value="Save"/>	<input type="button" value="Clear"/> <input type="button" value="Cancel"/>

Abbildung 4.22: Extend Decision Deadline – Eingabe der inkorrekten Testdaten

The month value is not valid.  
 Please enter a valid month value which is between 1 and 12. The year value is not valid.  
 Please enter a valid 4-digit year value. The following internal error appeared:  
 decision\_deadline\_year\_not\_valid, decision\_deadline\_month\_not\_valid, decision\_deadline\_day\_not\_valid31

Current Decision Deadline (dd.mm.yyyy)	New Decision Deadline (dd.mm.yyyy)
30.04.2005	30 04 2005
<input type="button" value="Save"/>	<input type="button" value="Clear"/> <input type="button" value="Cancel"/>

Abbildung 4.23: Extend Decision Deadline – Fehlermeldung

## 4.4 Delegation Feature 2

### 4.4.1 Beschreibung der Aufgabe

Im Modul Delegation feature kann die Delegationmatrix nicht nach Artikelkategorien sortiert werden. Dies soll in Rahmen von Kurzthema 04 geändert werden, indem ein neuer Workflow, der diese Funktion implementiert, erstellt wird.

### 4.4.2 Modellierungsphase

In diesem Abschnitt werden der alte und der neue Entwurf zum Anzeigen der Delegationmatrix mittels Anwendungs- und Aktivitätsdiagrammen dargestellt. Die Diagramme sind in Together 6.2 erstellt worden.

#### Der alte Zustand

In dem Alten Entwurf konnten die Artikel in der Delegationmatrix nur nach ihrer ArtikelID sortiert werden. Eine Sortierung der Artikel der Delegationmatrix nach den entsprechenden Kategorien war nicht gewährleistet.

Das Anwendungsdiagramm in der Abbildung 4.24 beschreibt die Übersicht der delegierten Reviews und die unterschiedlichen Aktionen, die für jede Delegation durchgeführt werden können, wobei die zur Verfügung stehenden Aktionen von der Rolle des aktuellen Benutzers abhängen.

Während der Benutzer Editor nur die Aktionen `ExtendDecision` und `ExtendTask` durchführen kann, stehen der Benutzer Editorial Office zusätzlich dazu die Aktionen `RemovedDelegation` und `DelegateArticle` zur Verfügung.

Das Aktivitätsdiagramm in der Abbildung 4.25 zeigt die verschiedenen Schritte, die zum Anzeigen der Delegationmatrix notwendig sind. Zuerst werden die delegierten Artikel, die Benutzerkategorien, die erlaubten Benutzeraktionen und die Deadlines aus der Datenbank gelesen. Diese Daten werden dann gefiltert, aufbereitet und in den Content als Delegationmatrix eingefügt. Alle diese Schritte sind in einem einzigen SIB implementiert was zu einer Komplikation des Entwurfes führt.

#### Der neue Zustand

Der neue Workflow soll die Sortierung der anzuzeigenden Delegationmatrix nach den Artikelkategorien gewährleisten. Das neue Anwendungsdiagramm in der Abbildung 4.26 zeigt die Erweiterung des alten Anwendungsdiagramms um die Aktionen: `sortByType` und `sortById`.

Das neue Aktivitätsdiagramm in der Abbildung 4.27 zeigt die Aufteilung des SIBs `ShowDelegationMatrix` in mehrere SIBs, damit das patternorientierte

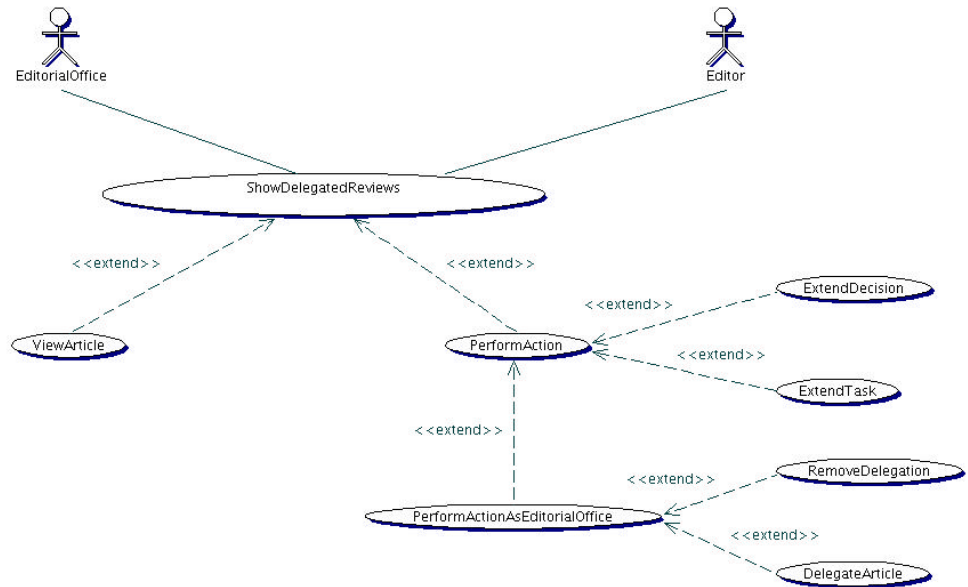


Abbildung 4.24: Alter Use Case

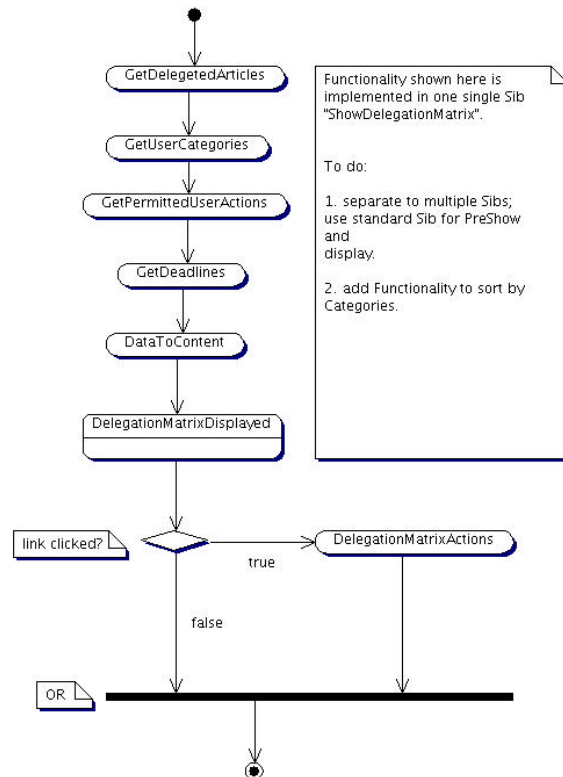


Abbildung 4.25: Altes Activity Chart

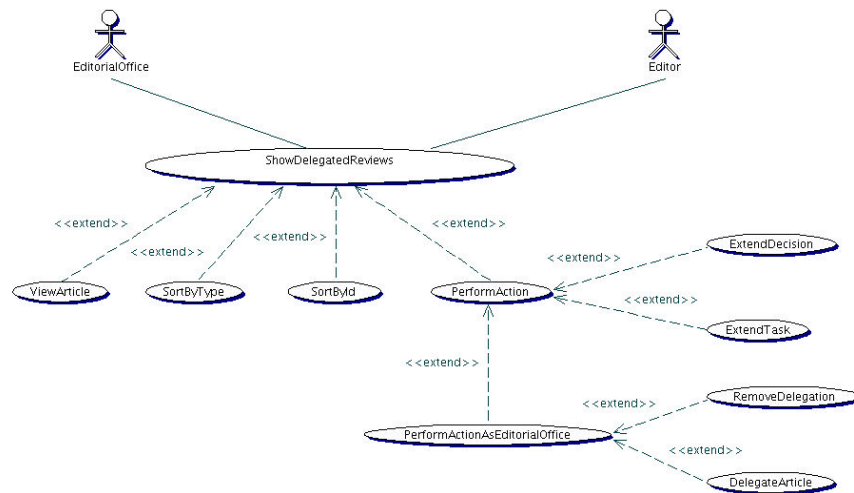


Abbildung 4.26: Neuer Use Case

Konzept eingehalten wird, sowie die Erweiterung des Workflows um den SIB `GenerateDelegationMatrix`, der die Sortierung der Delegationmatrix nach den Artikelkategorien implementiert.

### 4.4.3 Implementierungsphase

In diesem Abschnitt werden die verschiedenen Schritte im Rahmen der Implementierung des neuen Workflows mittels ABC dargestellt. Nach einer kurzen Übersicht wird insbesondere auf die Funktionsweise des Workflows sowie die beteiligten SIBs und deren Spezifikationen eingegangen.

#### Übersicht

Das Generieren und Anzeigen der Delegationmatrix sind in dem alten Workflow in einem einzigen SIB implementiert worden. Dies schränkt erheblich die Modularität und die Wiederverwendbarkeit ein. Von daher wurde der alte SIB `ShowDelegationMatrix` durch drei neue SIBs ersetzt, um die Funktionalität der Generierung der Delegationmatrix von der Funktionalität der Anzeige zu trennen. Die generierten Daten werden zuerst in den Call Context geschrieben dann werden sie mittels des SIB `CallContext2PreShow` für die Anzeige aufbereitet bevor sie schließlich in den Content übernommen und angezeigt werden.

#### Funktionsweise des neuen Workflows

Im SIB `GenerateDelegationMatrix` werden zuerst die nötigen Daten für die Delegationmatrix mittels Datenbankadministratoren geholt und nach den entsprechenden Rechten des aktuellen Benutzers gefiltert. Die generierte Delegationmatrix ist standardmäßig nach ArtikelIDs sortiert und wird in den Call Context geschrieben.

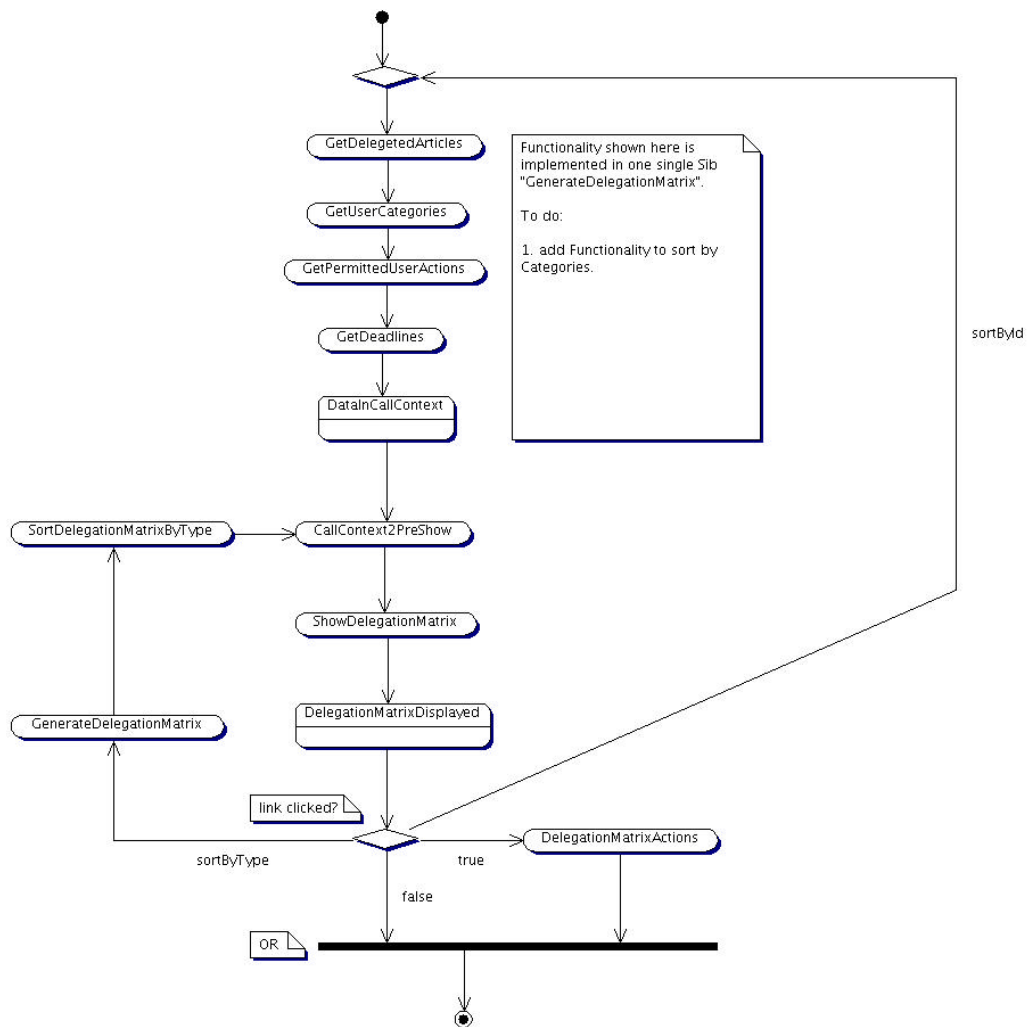


Abbildung 4.27: Neuer Activity Chart

---

In einem zweiten Schritt wird die Matrix in den Content mittels dem Standard-SIB `CallContext2PreShow` geschrieben und dann mittels `ShowFrameSetFiles` angezeigt. Die neue Sortierfunktionalität kann durch den neuen Link in `delegationmatrix-cont.wml` angestoßen werden. Dies führt zu einem weiteren `GenerateDelegationMatrix`. In diesem Branch des Workflows werden die generierten Daten nach den Artikelkategorien in `SortDelegationMartrixByType` sortiert bevor sie an den Content zur Anzeige weitergeleitet werden.

#### Beschreibung der beteiligten SIBs

- `GenerateDelegationMatrix` Dieser SIB generiert eine Matrix der delegierten Artikel, die der aktuelle Benutzer sehen darf. Die Matrix enthält eine Liste der möglichen durchzuführenden Aktionen und der gesetzten Deadlines für jede Delegation.
- `SortDelegationMatrixByType` Dieser SIB sortiert die im Call Context vorhandene Delegationmatrix nach den Artikelkategorien.

#### 4.4.4 Testphase

Das Testen des neuen Workflows wurde manuell durchgeführt. Es wurden mehrere Delegations generiert und in der Delegationmatrix angezeigt. Das Sortieren der Delegationmatrix ist deterministisch, sodass die Matrix zunächst nach den Artikelkategorien und dann nach ihrer ArtikelID als zweiter Schlüssel sortiert wird. Die Abbildung 4.28 zeigt das Sortieren der Delegationsmatrix nach den Artikelkategorien.

markus office

- Administrator
- Autor
- Reviewer
- Editorial Office
- Editor

## Overview of Delegated Reviews

This page gives an overview of the delegated reviews, i.e. when and to whom you have delegated a review of an article. So you can get the information about the "position" of articles in the reviewing process.

The legend at the bottom of this page explains the content of each column.

The following table can be sorted according to the following criteria: **Type**, **ID**. Click on the desired one.

**Refresh** (last refresh: *Thu Nov 25 15:36:11 CET 2004*)

ID	Delegated from	Delegated to	Status(Since)	Decision Deadline	Task Deadline	Action	Comments	Type	ID
003	mo	markus reviewer	waiting for decision(since 2004-10-28)	2004-10-28	2004-11-25	extend decision • delegate article • remove delegation	Please review this paper and submit a report about it	cat 1	003
004	mo	markus editor	done, report submitted on 2004-10-28				Please review this paper and submit a report about it	cat 1	004
004	mo	markus reviewer	done, report submitted on 2004-10-28				Please review this paper and submit a report about it	cat 1	004
005	mo	markus reviewer	review task rejected on 2004-10-28, reason: 2/4			delegate article • remove delegation	Please review this paper and submit a report about it	cat 1	005
005	mo	markus editor	waiting for decision(since 2004-10-28)	2004-10-28	2004-11-25	extend decision • delegate article • remove delegation	Please review this paper and submit a report about it	cat 1	005
006	mo	markus reviewer	done, report submitted on 2004-10-28					cat 1	006
007	mo	markus reviewer	done, report submitted on 2004-10-28				Please review this paper and submit a report about it	cat 1	007
010	mo	markus reviewer	waiting for decision(since 2004-11-10)	2004-11-10	2004-12-08	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 1	010
015	mo	markus reviewer	review task accepted on 2004-11-10		2004-12-08	extend task • remove delegation	Please review this paper and submit a report about it	cat 1	015
015	mo	markus editor	waiting for decision(since 2004-11-25)	2004-11-25	2004-12-23	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 1	015
011	mo	markus reviewer	waiting for decision(since 2004-11-10)	2004-11-10	2004-12-08	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 2	011
014	mo	markus reviewer	waiting for decision(since 2004-11-10)	2004-11-10	2004-12-08	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 2	014
013	mo	markus reviewer	waiting for decision(since 2004-11-10)	2004-11-10	2004-12-08	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 3	013
016	mo	markus reviewer	waiting for decision(since 2004-11-10)	2004-11-10	2004-12-08	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 4	016
016	mo	markus editor	waiting for decision(since 2004-11-25)	2004-11-25	2004-12-23	extend decision • extend task • delegate article • remove delegation	Please review this paper and submit a report about it	cat 4	016
012	mo	markus reviewer	review task rejected on 2004-11-19, reason: 2/4/2/2/2/2/2/2			delegate article • remove delegation	Please review this paper and submit a report about it	cat 5	012
012	mo	markus editor	waiting for decision(since 2004-11-19)	2004-11-19	2004-12-17	extend decision • delegate article • remove delegation	Please review this paper and submit a report about it	cat 5	012

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

Abbildung 4.28: Sortieren nach Artikelkategorien

## 4.5 Report Feature 2

### 4.5.1 Beschreibung der Aufgabe

Die Begutachter müssen das Reportformular erneut komplett ausfüllen, wenn die korrigierte Version eines Artikels erneut an die Begutachter der ersten Artikelversion delegiert wird. Dieses Verfahren ist unpraktisch, komplex und zeitaufwändig.

Der Workflow wurde erweitert, so dass der Begutachter zum Einreichen eines Berichtes über einen Artikel zwei Möglichkeiten hat: Entweder schreibt er direkt einen Bericht oder er kann einen vorherigen Bericht über diesen Artikel öffnen, der von ihm selbst geschrieben worden ist und dann die Eingabe des Berichtes modifizieren und als neuen Bericht einreichen.

Das Design und die Lösung bestehen darin, zwei neue SIBs zu schreiben und zu implementieren. Es handelt sich um **CheckOldReport** SIB und **GenerateReportSubmitData** SIB.

### 4.5.2 Modellierungsphase

In der Modellierungsphase wird der bestehende Workflow analysiert, um zu ergründen, wie er verändert werden muss (siehe Abbildung 4.29).

Um das Ziel der Aufgabe zu erreichen, muss ein neuer Anwendungsfall des Soll-Zustands eingefügt werden (siehe Abbildung 4.30).

Vor der Durchführung der Aufgabe bestand keine Möglichkeit, bei einer `Reportsubmission` direkt auf den vorherigen Bericht zuzugreifen. Die Begutachter sollten das Reportformular erneut komplett ausfüllen. Dieser Workflow muss für die Erfüllung der Aufgabe geändert werden. Bevor der Begutachter seinen Bericht schreibt, kann er zu dem entsprechenden Artikel eine Berichtliste einholen. Dazu wählt er seinen letzten Bericht aus und klickt auf "use" Button, dann wird das Berichtformular mit den Daten des vorherigen Berichtes ausgefüllt. Nach der Änderungen des alten Berichtes kann der Begutachter diesen als neuen Bericht abschicken (siehe Abbildung 4.31).

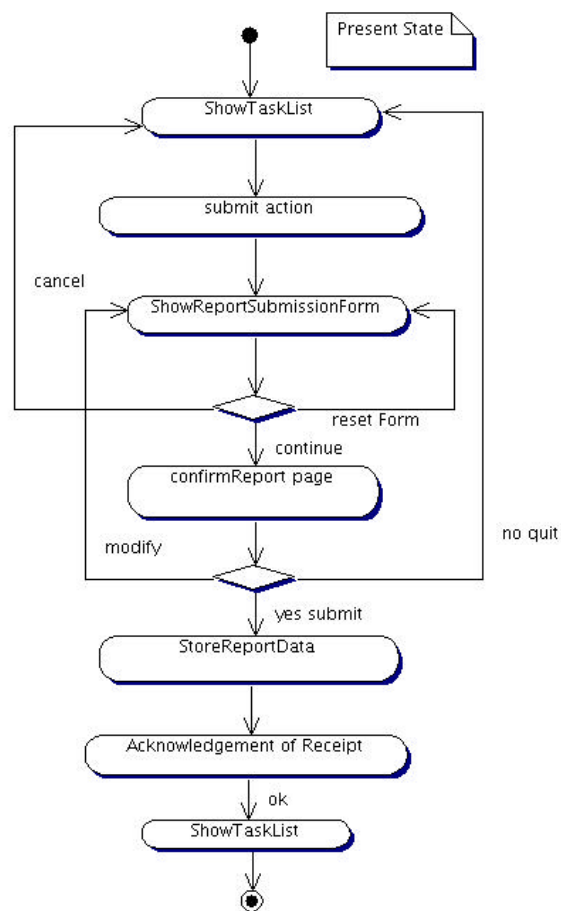


Abbildung 4.29: Das alte Activity Chart

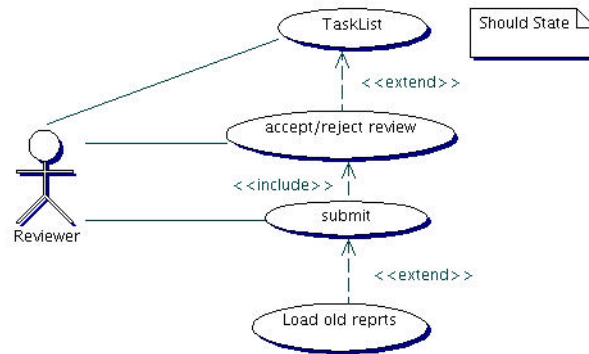


Abbildung 4.30: Das neue Use Case

### 4.5.3 Implementierungsphase

Um das Ziel der Aufgabe zu erfüllen, müssen einige WML-Seiten und neue java-Klassen implementiert werden. Zuerst wird die WML-Seite "submitreport-cont.wml" bearbeitet. Die einzige Änderung ist die Erweiterung um den Link „Check Old Report“. Der Link ermöglicht dem Begutachter den Zugriff auf seine Berichtliste bezüglich eines Artikels. Der Link hat die ArtikelID als Argument und wird als Parameter an einen neu geschriebenen SIB CheckOldReport übergeben. Dieser SIB generiert aus der ArtikelID einen Vektor OldReportVec, der alle vorherigen Berichte enthält und schreibt diesen Vektor in den Call Context. Der nächste SIB CallContext2PreShow holt die Daten aus dem Call Context und gibt sie dem SIB ShowOldReport (siehe Abbildung 4.32).

Als nächstes wird die neue WML-Seite "oldreport-cont.wml" gefüllt und angezeigt. Mit Hilfe der Radio-Button wählt der Reviewer seinen Bericht und klickt auf den "use"-Button. Dadurch schreibt der SIB Request2CallContext die ReportID in den Call Context. Der SIB "GenerateReportSubmitDate" generiert aus der ReportID die Reportparameter und schreibt sie in den Call Context. Somit werden alle Informationen gesammelt, die für die Erfüllung der Aufgabe gebraucht werden. In dem nachfolgenden SIB CallContext2PreShow werden die Daten von dem Call Context geholt und dem "Show File" ShowSubmitReport übergeben. Letztendlich wird das Berichtformular vorgefüllt und der Reviewer kann nach der durchgeführten Modifikation seinen neuen Bericht abschicken.

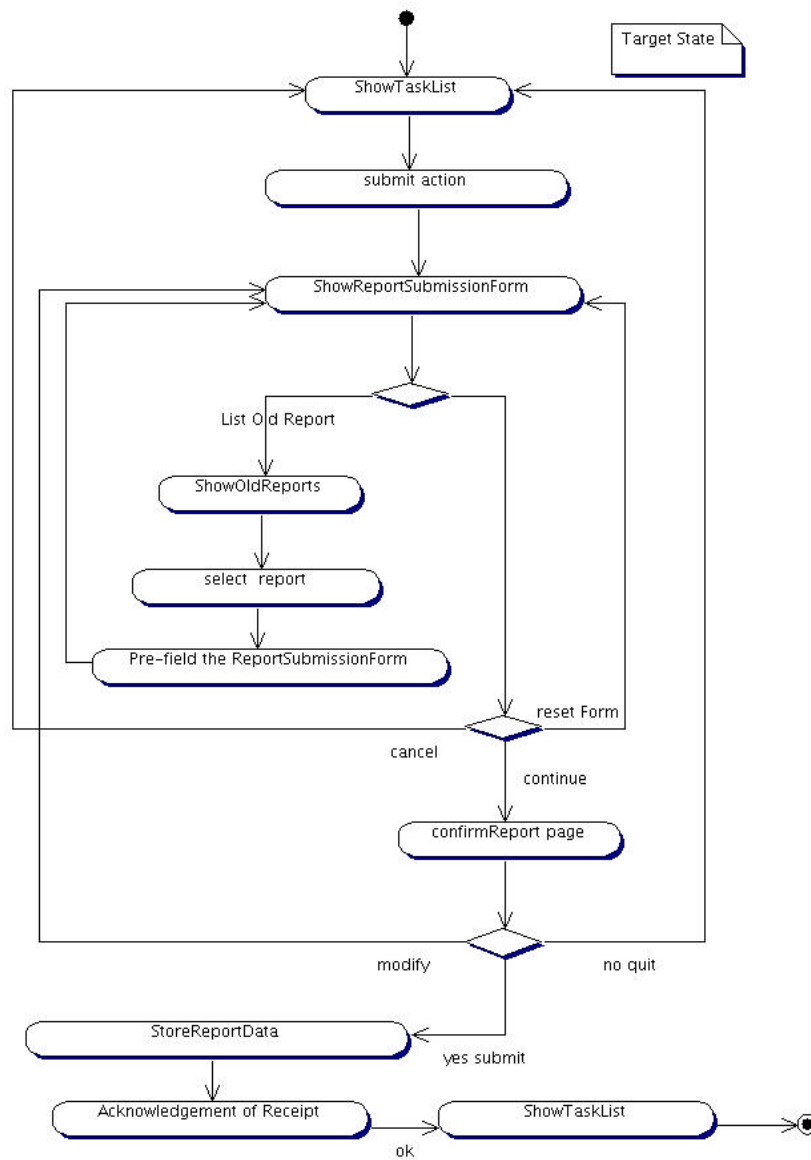


Abbildung 4.31: Das neue Activity Chart

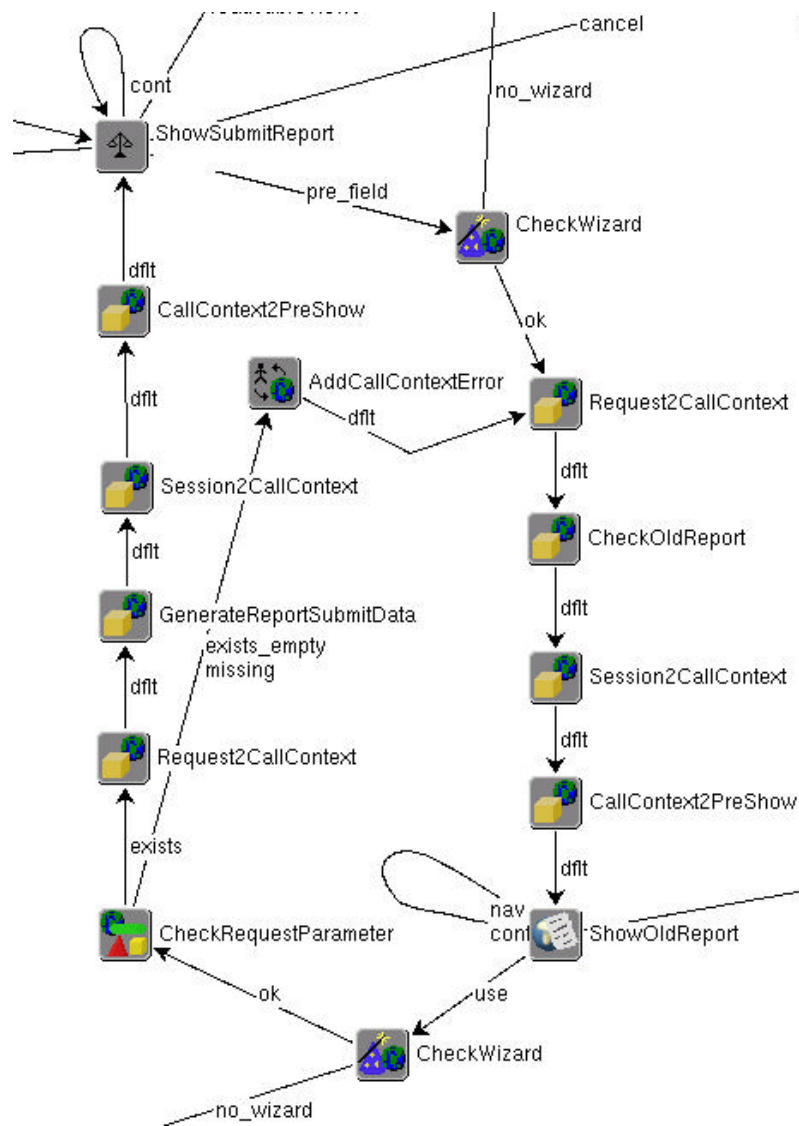


Abbildung 4.32: Der modifizierte Workflow

## 4.5.4 Testphase

Die Aufgabe wird laut den Anforderungen erfüllt. Es wurde der Link "Information to old reports" eingefügt (siehe Abbildung 4.33). Nach dem Anklicken auf diesem Link wird eine Reportliste angezeigt (siehe Abbildung 4.34). Bei der Auswahl eines Reports wird das Berichtformular mit den Daten des entsprechenden Reports ausgefüllt. Beim Testen lassen sich keine Fehler erkennen

### Report Submission Form

**Note: Prevent the loss of your report!**

Type your review directly into this form, **ONLY** if you don't need more than 120 minutes.

- **Problem:** From the server point of view typing your report is an inactive behaviour (as for all HTML-based online forms, filling them is local to your browser). The service will automatically log you out after 120 minutes inactivity! This means, that you will then lose your text!
- **Solution:** Write your review in a text editor. Then come back to this form and copy and paste the text into the corresponding text areas!

### Information to old reports

Abbildung 4.33: ReportSubmissionForm mit dem Link Information to old report

### Check Old Report

Choice	ID	Referee	Subreviewer	Score	Confidence	Date
<input type="radio"/>	007-rv-9-R1	Andersson, Nancy		5	3	2004-11-30
<input type="radio"/>	007-rv-8-R1	Andersson, Nancy		4	2	2004-11-30
<input type="radio"/>	007-rv-7-R1	Andersson, Nancy		2	3	2004-11-27
<input type="radio"/>	007-rv-6-R1	Andersson, Nancy		5	2	2004-11-27
<input type="radio"/>	007-rv-5-R1	Andersson, Nancy		4	3	2004-11-27
<input type="radio"/>	007-rv-4-R1	Andersson, Nancy		4	3	2004-11-27
<input type="radio"/>	007-rv-3-R1	Andersson, Nancy		4	3	2004-11-27
<input type="radio"/>	007-rv-2-R1	Andersson, Nancy		4	3	2004-11-24
<input type="radio"/>	007-rv-1-R1	Andersson, Nancy		3	2	2004-11-24
<input type="radio"/>	007-R1	Andersson, Nancy		2	2	2004-11-24

use Clear Cancel

Abbildung 4.34: Old report Liste

## 4.6 Evaluation Feature

### 4.6.1 Beschreibung der Aufgabe

Die Aufgabenstellung umfasste zwei Positionen, die beide die Evaluation Matrix betreffen und den Umgang mit dieser für den Benutzer erleichtern.

- Zusammenfassung der Bewertungsdaten der Reviewer in eine Spalte unter Ausschluss der Reviewer, die an der Bewertung des betroffenen Artikels nicht teilgenommen haben (vorher wurde für jeden im System registrierten Reviewer eine Spalte angelegt, unabhängig davon, ob dieser bei der Bewertung tätig war oder nicht)
- Möglichkeit der Mehrfachselektion und Modifikation der Artikel in der Evaluation Matrix (ursprünglich war es möglich nur einzelne Artikel nacheinander zu Modifizieren)

### 4.6.2 Modellierungsphase

**Use Case Diagramm** Die folgende Abbildung 4.35 zeigt das Use Case Diagram aus der Sicht eines Editorial Office. Der Editorial Office User wertet die Evaluation Matrix aus und schliesst vier folgende Aktionen in den Prozeß des Auswertens ein:

- Änderung von Scores oder/und Confidences für selektierte Artikel
- Änderung der Stati der selektierten Artikel auf angenommen
- Änderung der Stati der selektierten Artikel auf vielleicht angenommen
- Änderung der Stati der selektierten Artikel auf zurückgewiesen

#### **Aktivitätsdiagramme**

Es liegen zwei Arten von Aktivitäten vor:

- Änderung von Scores / Confidences für selektierte Artikel
- Änderung der Stati für selektierte Artikel

*Änderung von Scores / Confidences* Bei der Änderung von Scores und/oder Confidences selektiert der Benutzer die zu modifizierenden Artikel, trägt die zu ändernden Scores und/oder Confidences ein und wählt als Aktion "Modify scores / confidences". Dieses Verhalten wird im folgenden Diagramm 4.36 modelliert.

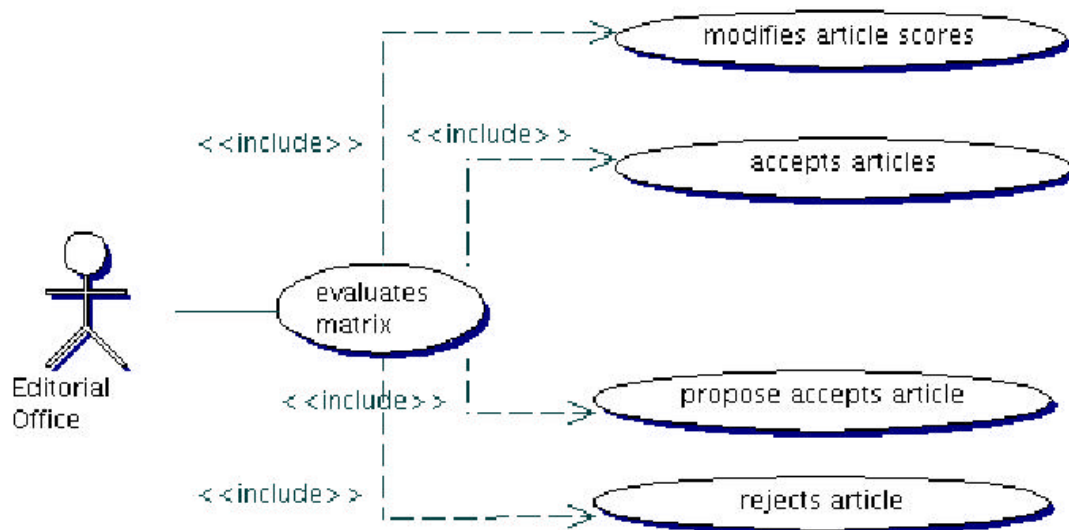


Abbildung 4.35: Use Case Auswertung/Modifikation der Evaluation Matrix

*Änderung der Stati für selektierte Artikel* Bei der Änderung der Stati für die selektierten Artikel gibt es drei mögliche Werte, die der Status annehmen kann:

- angenommen (accepted)
- vielleicht angenommen (propose accepted)
- abgelehnt (rejected)

Da der einzige Unterschied in dem Wert, den der neue Status für den Artikel annimmt, besteht, demonstrieren wir hier exemplarisch den Fall, dass ein Artikel angenommen wird (accepted). Dieser wird im folgenden Aktivitätsdiagramm 4.37 modelliert.

### 4.6.3 Implementierungspahse

Während der Implementierung wurden folgende SIB's modifiziert oder entwickelt:

- BuildArticleList
- BuildEvaluationMatrixConfirm
- CalculateEvaluationMatrixEntryList
- CollectRequestInfo
- EvaluationMatrixActionsList

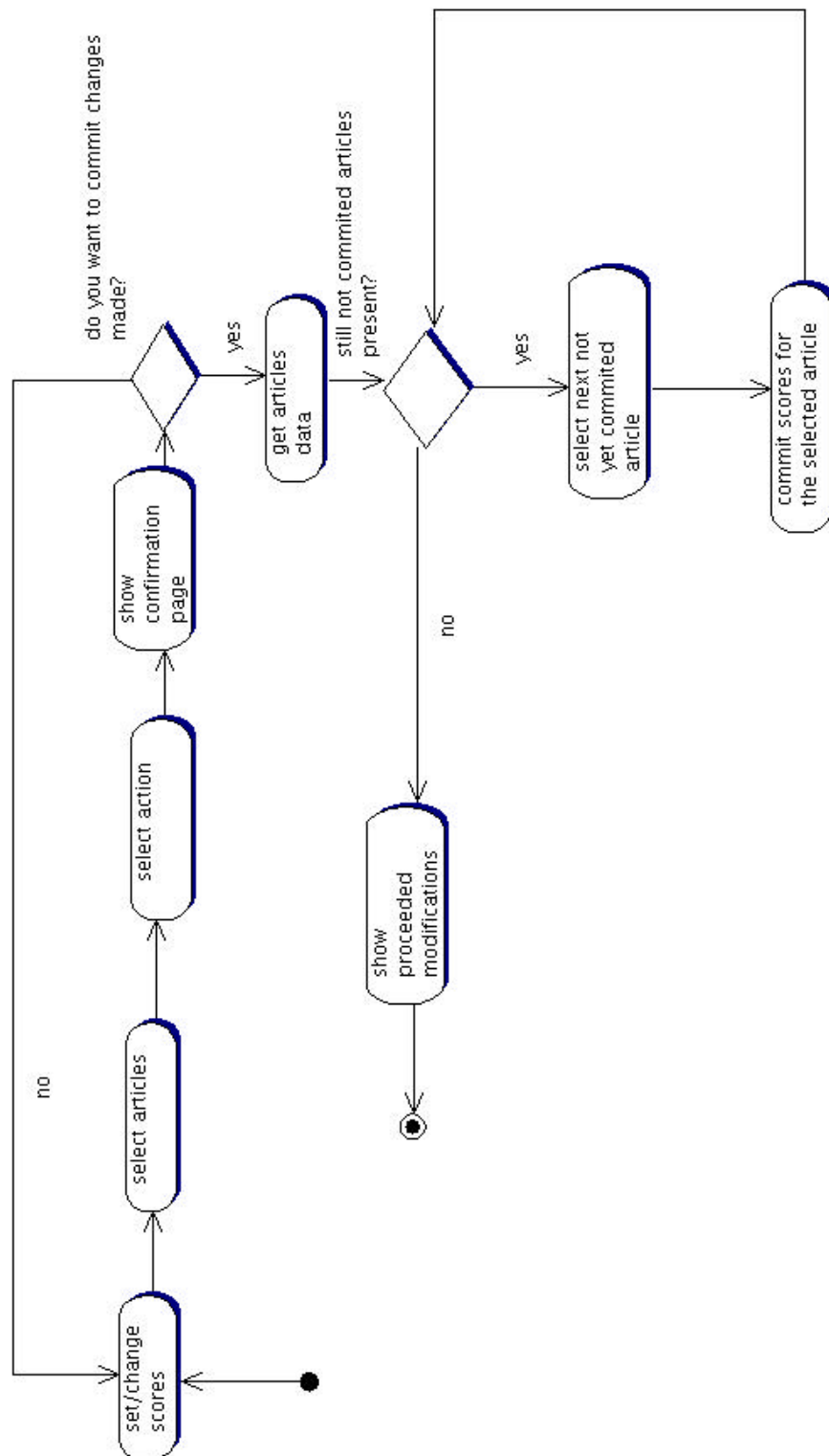


Abbildung 4.36: Activity Chart: Änderung von Scores / Confidences

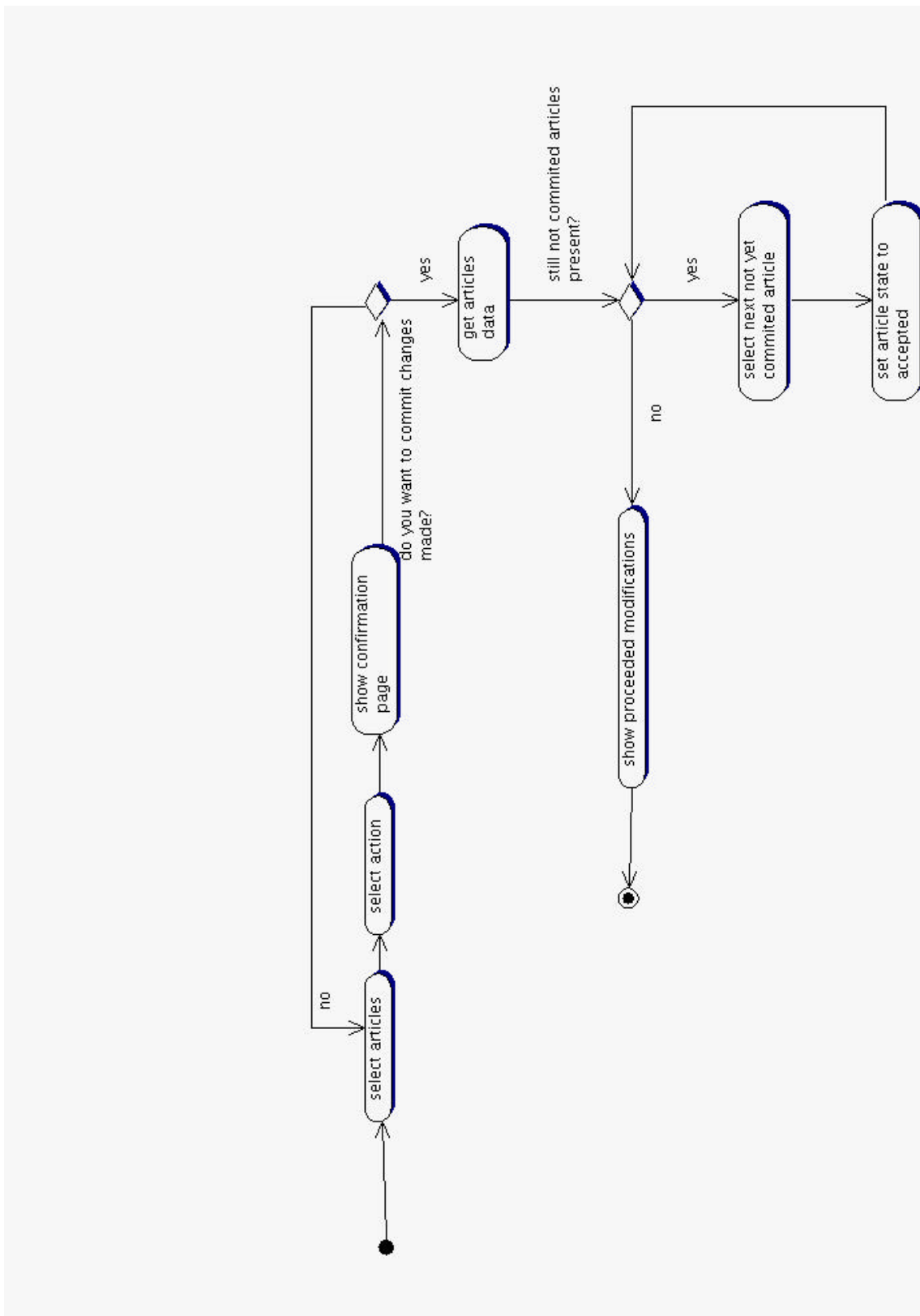


Abbildung 4.37: Activity Chart: Artikelstatus auf angenommen setzen

- UpdateScoreConfidenceList
- SetArticleStateList

Der zu implementierende Workflow besteht aus drei logische Abschnitten:

- Vorbereitungsphase
- Durchführungsphase
- Benachrichtigungsphase

#### **Vorbereitungsphase**

Die Abbildung 4.38 zeigt die Vorbereitungsphase. Hierbei werden die vom Benutzer eingegebene Werte durch den `CollectRequestInfo` SIB gesammelt, anschliessend wird vom SIB `BuildArticleList` die Liste der zu modifizierender Artikel zusammengestellt und daraufhin bildet der SIB `BuildEvaluationMatrixConfirm` die Daten für die Bestätigungsseite der Aktion zusammen.

#### **Durchführungsphase**

In der Durchführungsphase, die in der Abbildung 4.39 dargestellt wird, wird zunächst durch den SIB `EvaluationMatrixActionsList` der richtige Pfad für den Workflow, abhängig von der getätigten Aktion, selektiert. Anschliessend wird die zuvor zusammengestellte Liste der Artikel sequentiell durchgelaufen und je nach der ausgewählten Aktion entweder der SIB `UpdateScoreConfidenceList` ausgeführt, der die Scores und/oder die Confidences für einen Artikel ändert, oder der SIB `SetArticleStateList`, der den Artikelstatus auf einen zuvor selektierten Wert setzt.

#### **Benachrichtigungsphase**

In der Benachrichtigungsphase, wie die Abbildung 4.40 zeigt, werden nochmals alle Artikel, die von den aktuellen Modifikationen betroffen sind, durchgegangen, und der SIB `CalculateEvaluationMatrixEntryList` stellt die Daten für die Benachrichtigung über die Ergebnisse der Modifikation zusammen, die dem Benutzer anschliessend auf einer Seite angezeigt werden.

### **4.6.4 Testphase**

Die folgenden Abbildungen demonstrieren noch einmal den Ablauf des Prozesses aus der Sicht des Benutzers. Zunächst werden die gewünschten Änderungen angegeben, wie die Abbildung 4.41 zeigt, dann wird vom Benutzer eine Bestätigung über die auszuführende Aktionen eingeholt, wie die Abbildung 4.42 zeigt, und anschliessend, nachdem die Operationen abgeschlossen sind, wird dem Benutzer die Benachrichtigung über die vorgenommenen Änderungen angezeigt, was in der Abbildung 4.43 dargestellt wird.

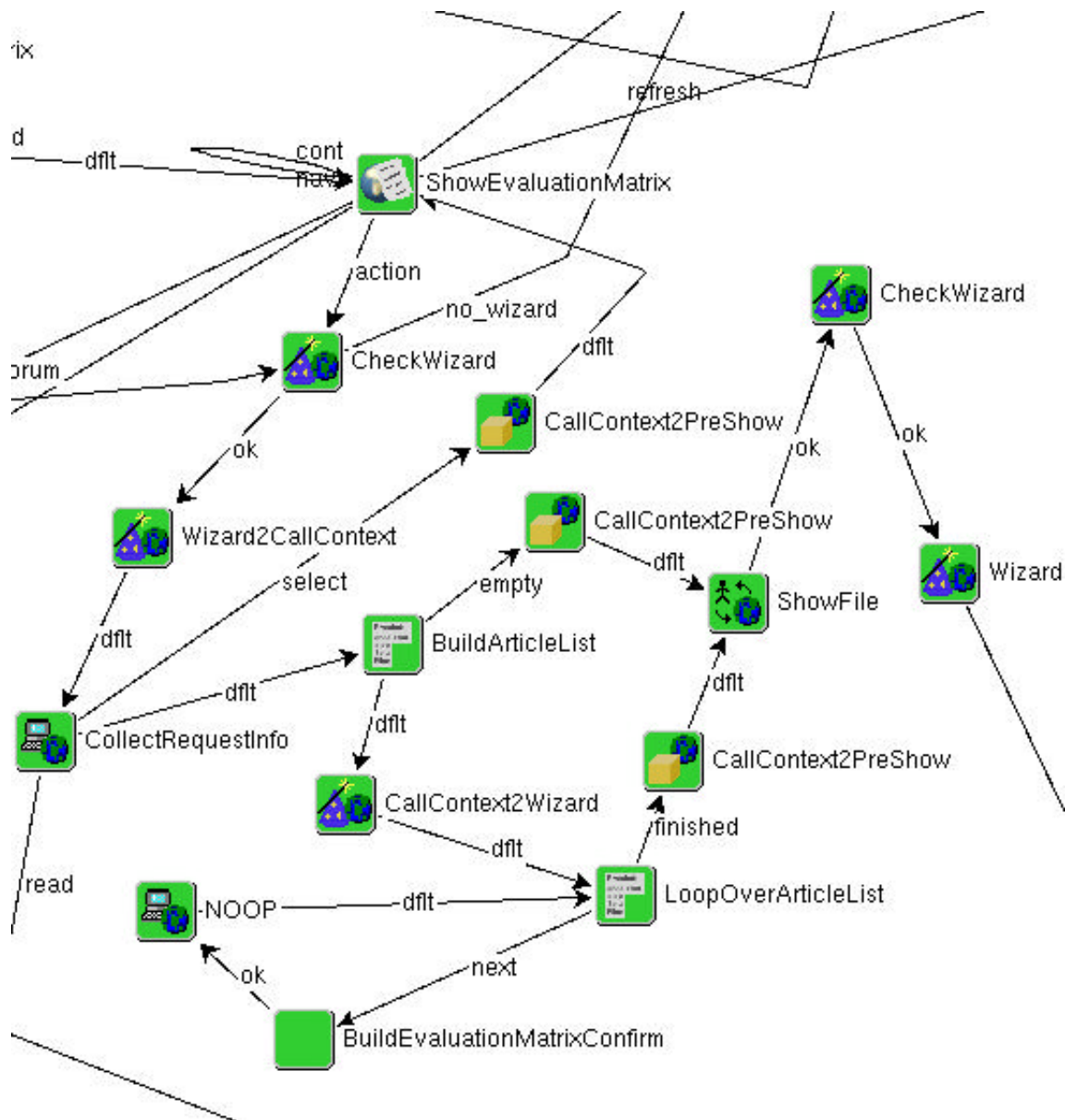


Abbildung 4.38: Vorbereitungsphase des Workflows

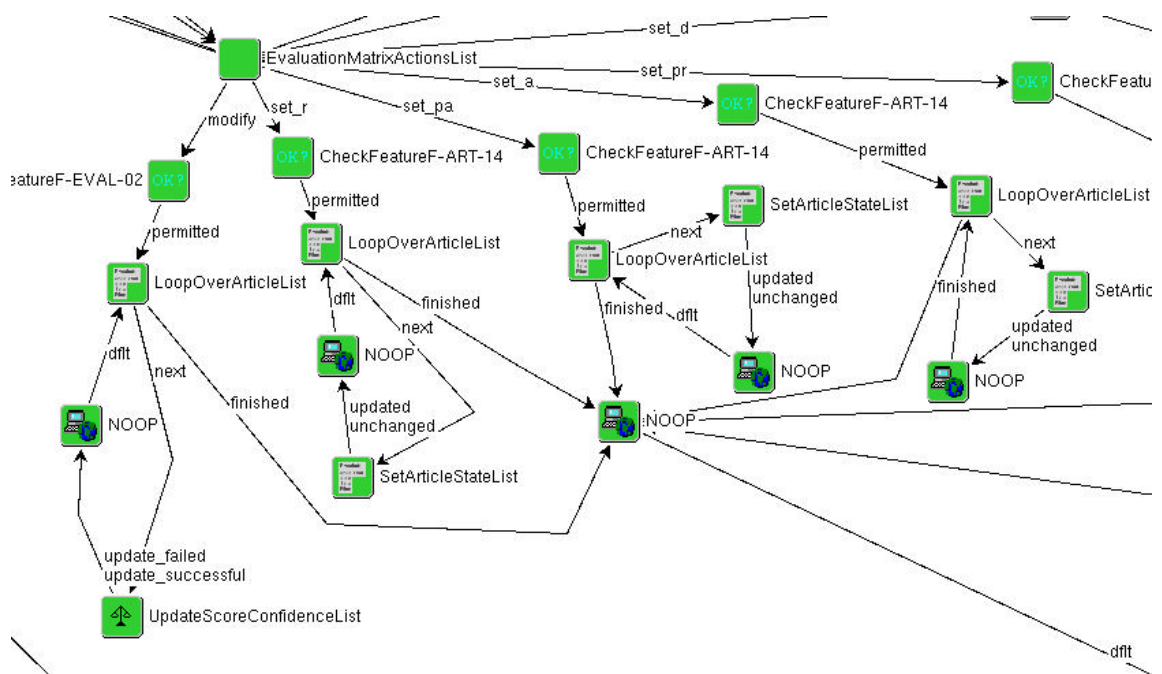


Abbildung 4.39: Durchführungsphase des Workflows

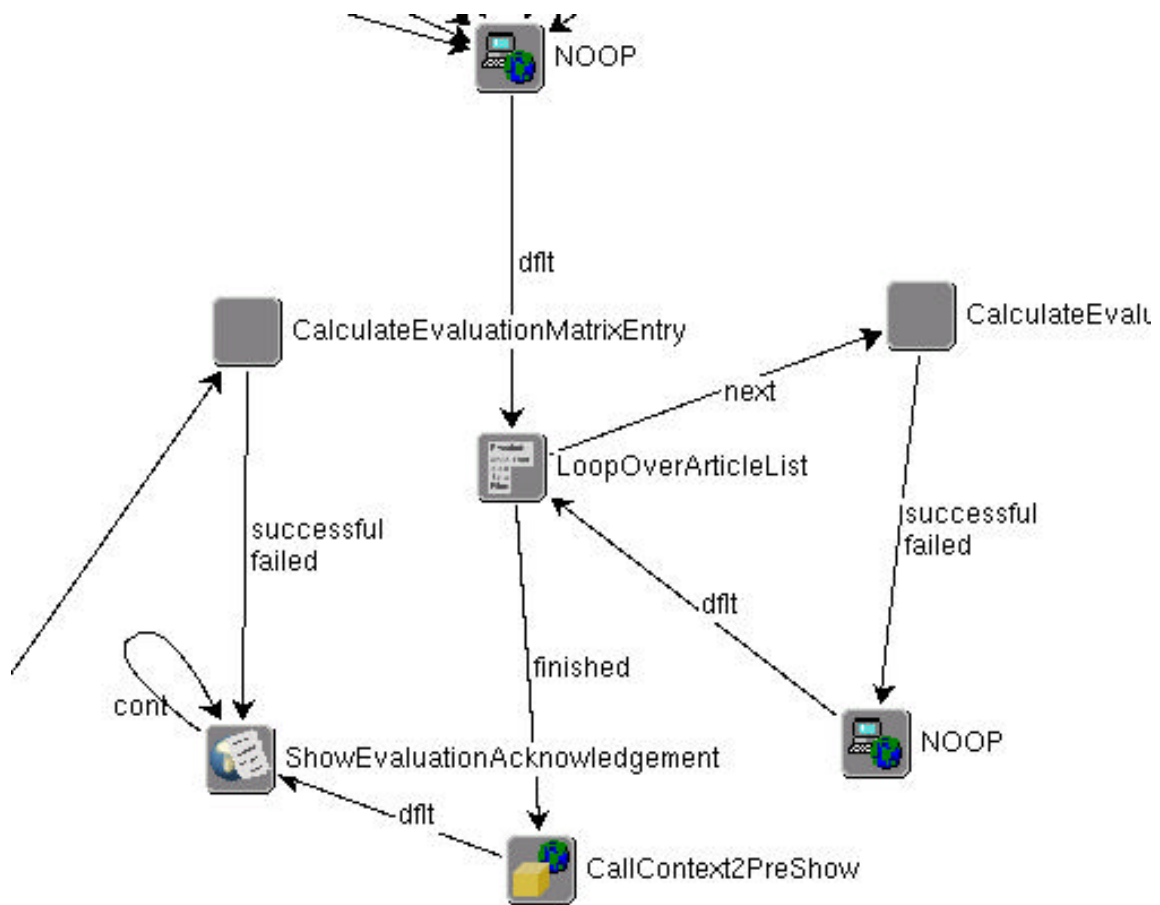


Abbildung 4.40: Benachrichtigungsphase des Workflows

Sel.	Art.ID	R.	Ø	Scores	Status	Type	Art.ID
<input checked="" type="checkbox"/>	006-rv-1   R	1	5.0	JD: 5/2	revised_version_submitted	Physik	006-rv-1   R
<input checked="" type="checkbox"/>	006   R	1	5.0	JD: 5/2	revised_version_submitted	Physik	006   R
<input type="checkbox"/>	007   R	2	4.0	JD: 4/2	all_reports_available	Mathematik	007   R
<input type="checkbox"/>	006-rv-2   R	2	4.0	JD: 4/3	all_reports_available	Physik	006-rv-2   R
<input type="checkbox"/>	005-rv-2   R	2	4.0	JD: 4/2	all_reports_available	Mathematik	005-rv-2   R
<input type="checkbox"/>	005-rv-1   R	2	4.0	JD: 4/2	revised_version_submitted	Mathematik	005-rv-1   R
<input type="checkbox"/>	005   R	2	4.0	JD: 4/2	revised_version_submitted	Mathematik	005   R
<input type="checkbox"/>	004-rv-2   R	2	4.0	JD: 4/2	revised_version_submitted	Regular Paper	004-rv-2   R
<input type="checkbox"/>	004-rv-1   R	2	4.0	JD: 4/2	revised_version_submitted	Regular Paper	004-rv-1   R
<input type="checkbox"/>	002   R	3	3.5	JD: 3/2 KS: 4/2	all_reports_available	Regular Paper	002   R
<input type="checkbox"/>	012   R	4	3.0	BH: 3/3	some_reports_available	Chemie	012   R
<input type="checkbox"/>	001   R	4	3.0	HS: 3/2	all_reports_available	Regular Paper	001   R
<input type="checkbox"/>	004   R	5	1.07	JD: 3/3 KS: 0/2 HS: 0/3	revised_version_submitted	Regular Paper	004   R
<input checked="" type="checkbox"/>	004-rv-3   R	6	1.0	JD: 3/3 KS: 0/3 HS: 0/3	accepted	Regular Paper	004-rv-3   R
<input checked="" type="checkbox"/>	003   R	7	0.0	HS: 0/3	accepted	Regular Paper	003   R

- modifies the score/confidence values
- set the new article state to **Accepted**
- set the new article state to **Propose Accepted**
- set the new article state to **Rejected**

**Caution, operating with the Select All or the Unselect All actions will have t1 as consequence. First select the articles, than update the sc**

Abbildung 4.41: Selektion der Artikel und Eingabe der zu modifizierender Daten

### Evaluation Confirmation

Do you want to proceed with following changes?

Art.ID	Scores / Confidences	Status
006	"JD": 5/2;	revised version submitted
006-rv-1	"JD": 5/2;	revised version submitted

Abbildung 4.42: Bestätigung der gewünschten Änderungen

Close Window

### Evaluation Acknowledgement

Art.ID	Average	Scores / Confidences	Status
006	5.0	"JD": 5/2;	revised version submitted
006-rv-1	5.0	"JD": 5/2;	revised version submitted

Abbildung 4.43: Benachrichtigung über durchgeführte Änderungen

## Kapitel 5

# Gruppenthemen 1. Semester

Nach Beendigung der Kurzthemen fand erneut eine Unterteilung in kleinere Gruppen statt. Jetzt wurden 4 Gruppen mit jeweils 3 Personen gebildet und etwas umfangreichere Aufgaben gestellt. Diese Gruppenthemen sollten in 2 Monaten umgesetzt werden. Es folgt eine Übersicht über die einzelnen Gruppenthemen.

### 5.1 Verwaltung der Co-Autoren

#### 5.1.1 Beschreibung der Aufgabe

Der neue Entwurf des Workflows im Modul `Article` soll den Autoren bei der Einreichung eines neuen Artikels ermöglichen, neue Benutzer in den Dienst einzufügen. Diese Benutzer werden als Co-Autoren des entsprechenden Autoren behandelt.

Im alten Entwurf wurde die Liste der weiteren Autoren als ein komma-separierter String eingegeben. Dies hat einerseits den Nachteil, dass der Vorname und Nachname eines Co-Autoren nicht genau spezifiziert werden kann (wenn ein Benutzer mehrere Vornamen und/oder Nachnamen hat), andererseits ist der Co-Autor nicht als Dienstnutzer registriert und hat somit keinen eigenen Zugriff auf den Dienst und den eingereichten Artikel. Dies sollte dahingehend geändert werden, dass der einreichende Autor in der Lage ist, eine nicht limitierte Anzahl von Co-Autoren anzugeben, sowie zu bestimmen ob sie sich in den Dienst einloggen dürfen oder nicht.

#### 5.1.2 Modellierungsphase

Mittels Anwendungs-, Aktivitäts- und Klassendiagramm wird versucht, die Erweiterungen des neuen Workflows zu verdeutlichen und die Unterschiede zum alten Workflow aufzuzeigen.

Im neuen Workflow sollen die Co-Autoren als eine Liste von Benutzer-Objekten, an Stelle eines komma-separierten Strings gespeichert werden. Da dieser Vorgang an mehreren

Stellen im Modul `Article` vorkommt, wird im Folgenden lediglich `Submit Article` exemplarisch beschrieben.

Hierzu kann der Autor im Rahmen eines `Submit-Formulares` so viele Co-Autoren in die Liste einfügen wie er möchte. Der Inhalt der Liste kann geändert, verworfen oder gespeichert werden. Dann werden die fehlenden Daten eingegeben und die Einreichung bestätigt. Nach der Einreichung ist der Co-Autor endgültig als Benutzer im Dienst registriert.

Das Use Case Diagramm in der Abbildung 5.1 zeigt den `Workflow Article` als Beispiel einer Integration des optionalen `Workflow Edit CoAuthor list`.

Das Aktivitätsdiagramm in der Abbildung 5.2 beschreibt die unterschiedlichen Schritte, die bei der Einreichung eines Artikels von einem Submitter notwendig sind.

Das Klassendiagramm in der Abbildung 5.3 zeigt den neuen Datenbankadministrator `UserArticleIDAdministrator`, der für die Relationen zwischen `userids` und `articleids` zuständig ist. Diese Relationen speichern alle Co-Autoren für einen gegebenen Artikel mittels ihrer IDs in einer neuen Datenbanktabelle `userarticleid`.

### 5.1.3 Implementierungsphase

In diesem Abschnitt wird die neue Implementierung des Workflows dargestellt. Nach einer kurzen Übersicht über die implementierten oder modifizierten SIBs wird auf deren Spezifikationen eingegangen.

#### Übersicht

Die folgende Tabelle zeigt eine Liste der implementierten oder modifizierten SIBs.

Name des SIBs	Ort	Zustand
<code>CreateCoAuthorAssoc</code>	Makro <code>CoAuthorregistration</code>	implementiert
<code>CheckCoAuthorList</code>	Makro <code>CoAuthorregistration</code>	implementiert
<code>DeleteCoAuthorAssoc</code>	Makro <code>CoAuthorregistration</code>	implementiert
<code>CheckRegisterCoAuthor</code>	Makro <code>CoAuthorregistration</code>	implementiert
<code>CreateCoAuthorList</code>	Makro <code>CoAuthorList</code>	implementiert
<code>CheckCoAuthorFields</code>	Makro <code>CoAuthorList</code>	implementiert
<code>RemoveCoAuthor</code>	Makro <code>CoAuthorList</code>	impementiert
<code>FillArticleSubmitForm</code>	Makro <code>CoAuthorList</code>	modifiziert

#### Anwendung der SIBs

Die neu implementierten SIBs unterteilen sich in zwei Kategorien. Während die SIBs der ersten Kategorie als Standard einsetzbar sind, lassen sich die SIBs der zweiten Kategorie nur im Zusammenhang mit den Makros, denen sie angehören, anwenden.

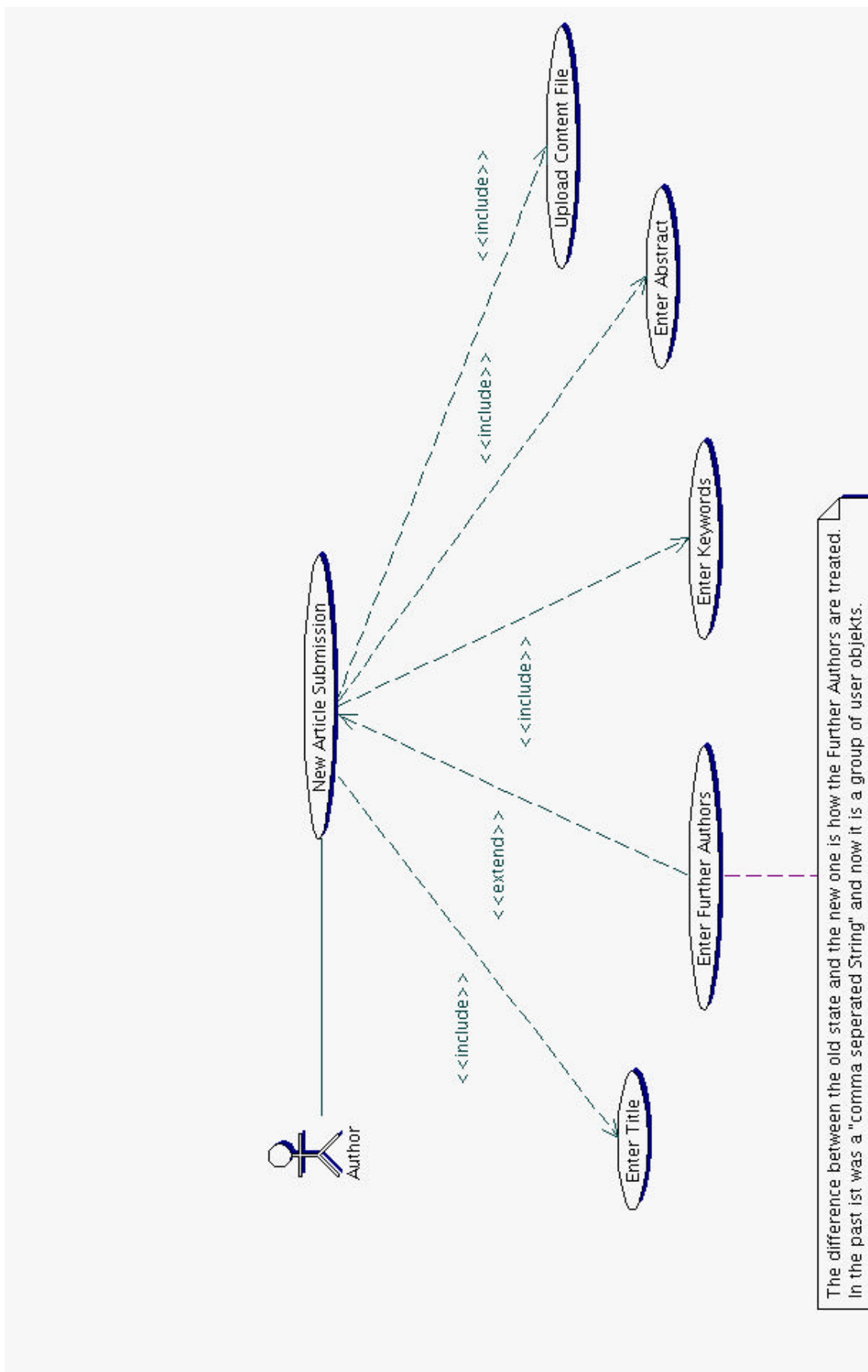


Abbildung 5.1: Use Case

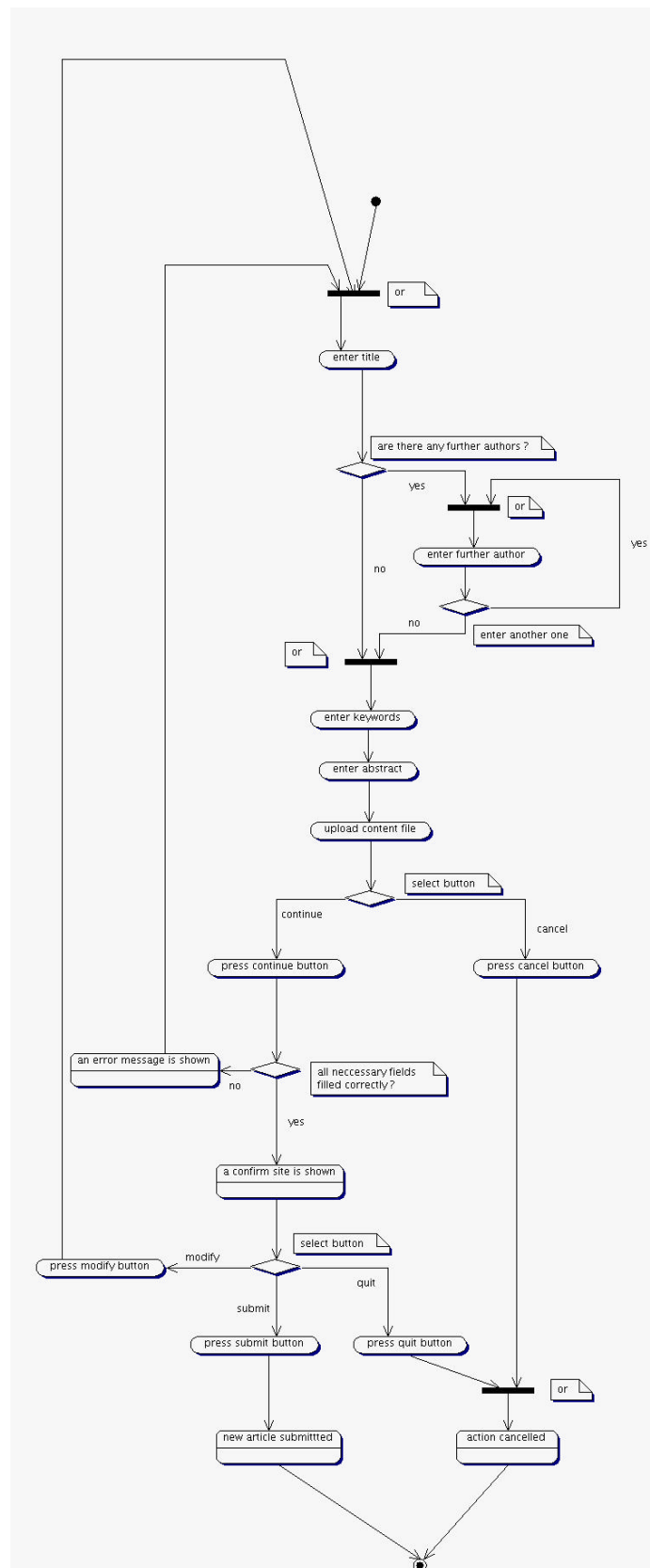


Abbildung 5.2: Activity Chart

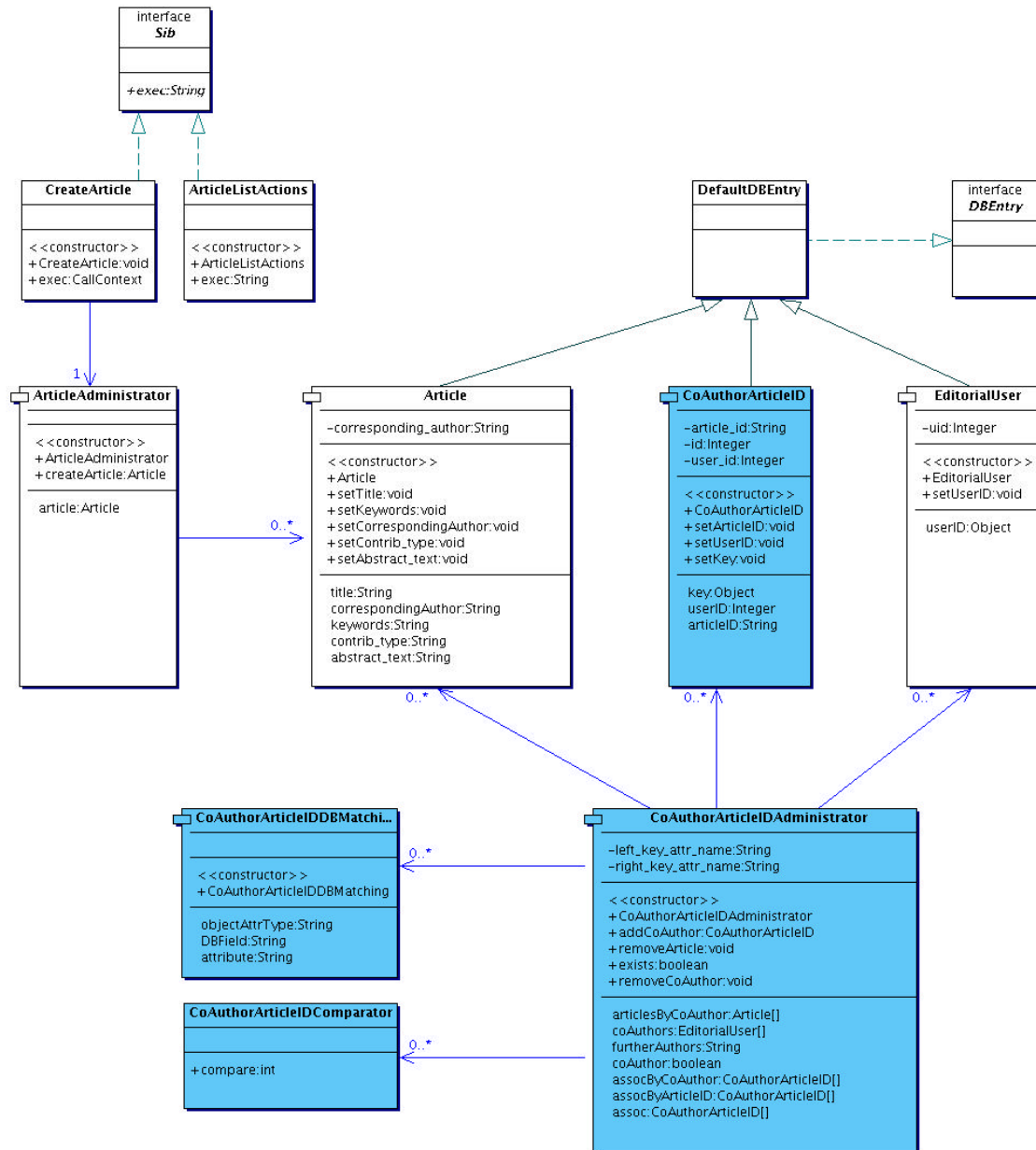


Abbildung 5.3: Klassendiagramm

- `CreateCoAuthorList` erwartet einen Artikel im Call Context und erzeugt einen Vector `CoAuthorList`, der die `EditorialUsers` für diesen Artikel enthält.
- `CreateCoAuthorAssoc` erwartet die `CoAuthorList` und einen Artikel. Der SIB erzeugt einen Datenbankeintrag in der `CoAuthorArticle`-Tabelle für das erste Element der Liste und den Artikel. Nach diesem Vorgang wird dieses Element aus der Liste entfernt.
- `DeleteCoAuthorAssoc` benutzt den vorhandenen Artikel im Call Context und entfernt alle Co-Autoren aus der `CoAuthorArticle`-Tabelle. Dabei werden die `EditorialUsers` nicht aus dem Dienst gelöscht.

Das Makro `CoAuthorList` kann in jedem Workflow benutzt werden, wo eine Modifizierung der `CoAuthorList` erwünscht ist. Das Makro benutzt einen Parameter zum Speichern des Namens der aufgerufenen HTML-Form und der darin gelieferten Daten. Für den Einsatz des Makros müssen die folgenden Objekte im Call Context gespeichert werden:

- `Article` wird als Artikel gespeichert und zur Erzeugung der `CoAuthorList` verwendet.
- Der entsprechende Autor kann nicht gleichzeitig der Co-Autor sein und ist wie folgt definiert:
  - falls ein Artikel vorhanden ist, wird die Artikelmethode `getSubmitterID` benutzt.
  - falls eine Form vorhanden ist, wird das HTML-Feld `Article_Author_Corresponding_Email` zum Definieren des Autoren angewendet.
  - ansonsten wird der gerade eingeloggte Benutzer als Autor definiert.

Die folgenden Abbildungen geben einen Überblick über die beiden implementierten Macros `CoAuthorList` (Abbildung 5.4) und `CoAuthorRegistration` (Abbildung 5.5).

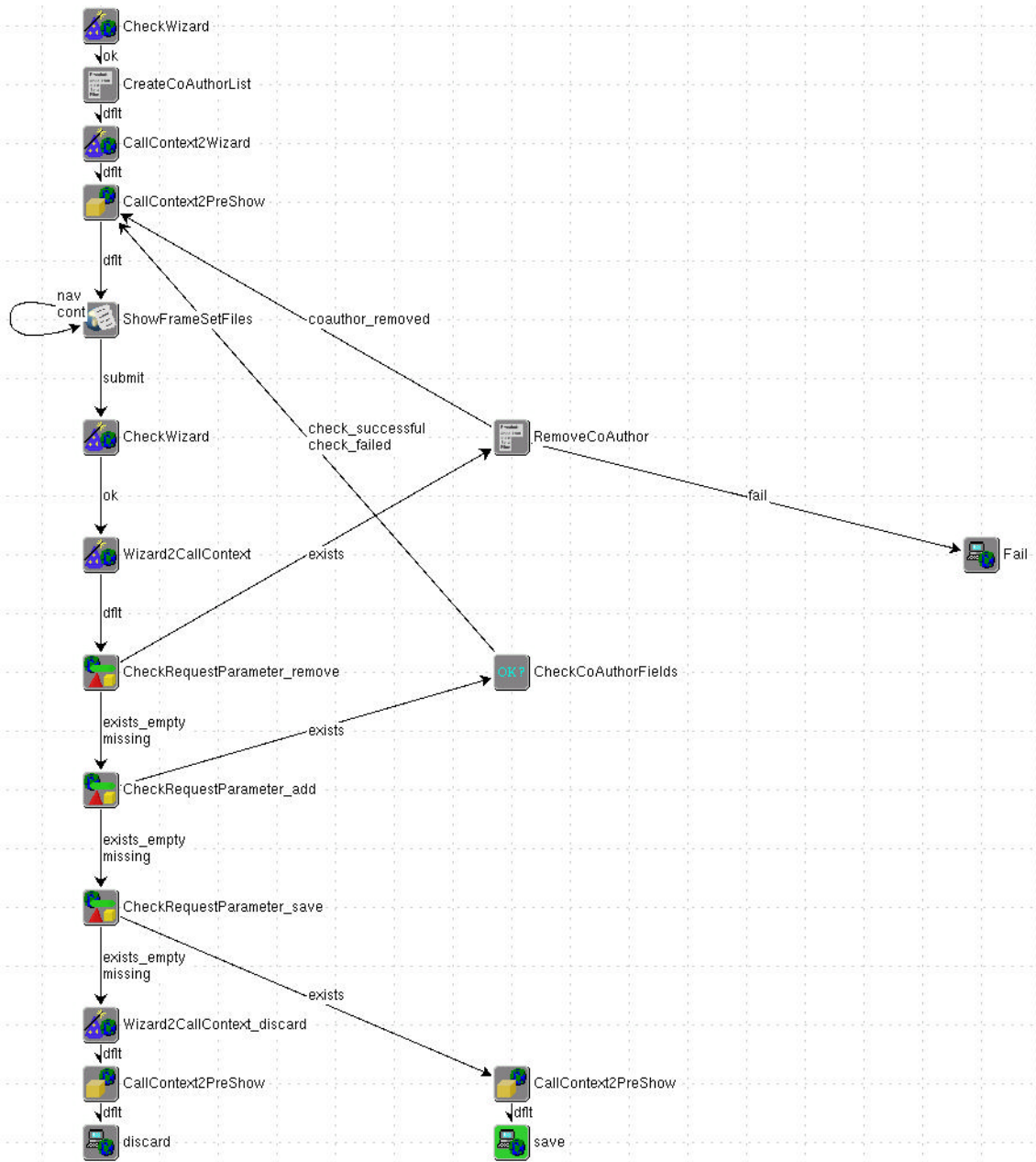


Abbildung 5.4: Macro CoAuthorList

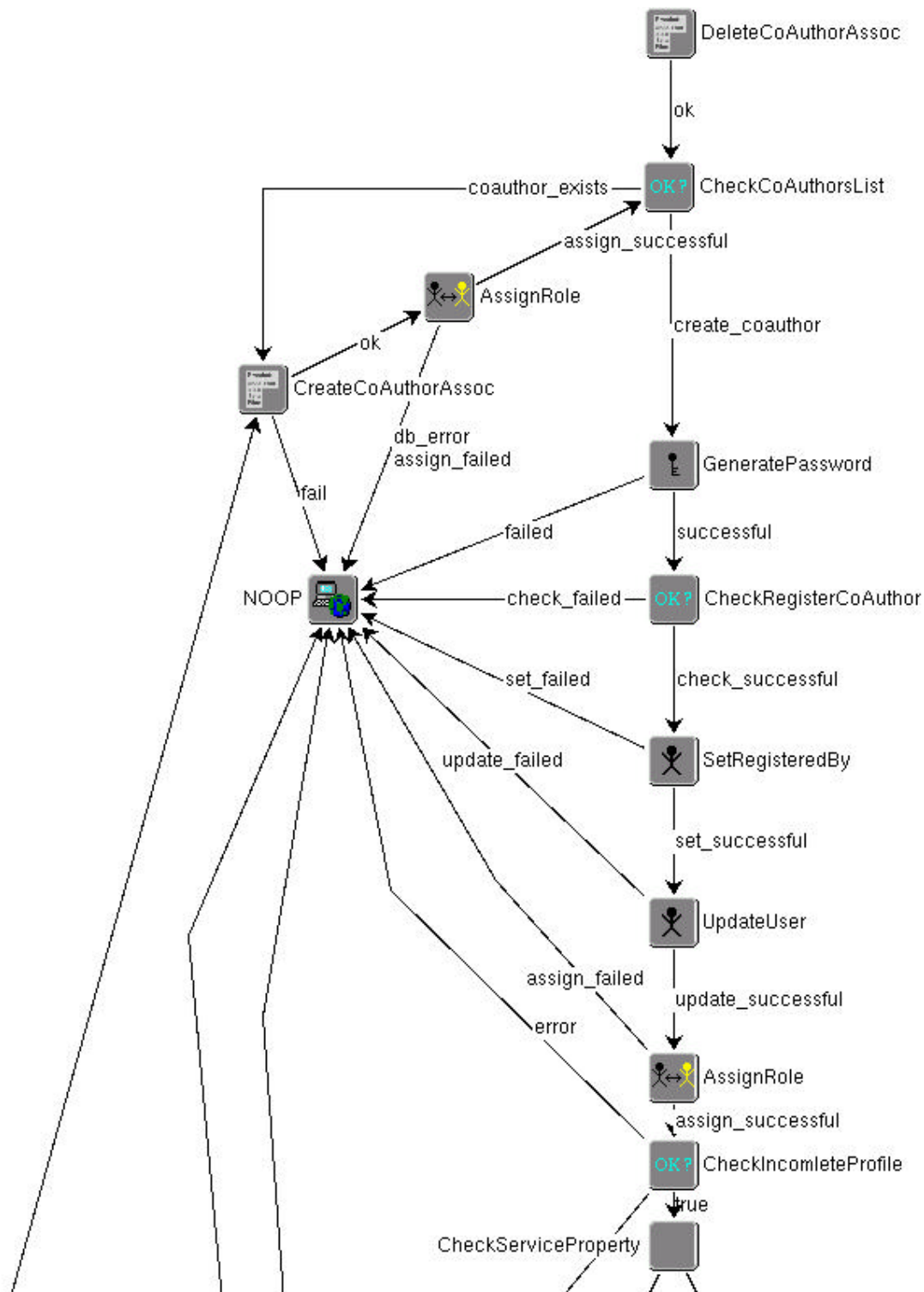


Abbildung 5.5: Macro CoAuthorRegistration

### Beschreibung der beteiligten SIBs

- `CreateCoAuthorAssoc` entfernt den ersten Eintrag der `CoAuthorList` und erzeugt eine Assoziation zwischen diesem Eintrag und einem Artikel. Die Assoziation wird in der Datenbank gespeichert. Ferner werden die Kategorien des ausgewählten Co-Autoren überprüft. Die Kategorie des aktuellen Artikels wird - falls nicht bereits vorhanden- zu den Benutzerkategorien hinzugefügt.
- `CheckCoAuthorList` holt sich den ersten Eintrag der `CoAuthorList` und überprüft, ob der Benutzer in der Datenbanken vorhanden ist.
- `DeleteCoAuthorAssoc` löscht alle Co-Autor-Assoziationen für einen gegebenen Artikel.
- `CheckRegisterCoAuthor` verschlüsselt und setzt das Passwort für den neu erzeugten Co-Autor.
- `CreateCoAuthorList` überprüft, ob eine Liste von Co-Autoren `CoAuthorList` im Call Context vorhanden ist. Falls nicht werden die entsprechenden Co-Autoren für den aktuellen Artikel von der Datenbank geholt und als Liste im Call Context gespeichert.
- `CheckCoAuthorFields` überprüft die sich im Request befindlichen Felder von der HTML-Form nach den folgenden Kriterien:
  - gültiges Email Format
  - Überprüfung der Email Bestätigung
  - Überprüfen, ob die Email bereits in der Datenbank vorhanden ist. Der Benutzer wird benachrichtigt, falls der Eintrag bereits existiert.
  - Überprüfen der Vollständigkeit aller Felder
  - Überprüfen des erzeugten Benutzerkontos
- `RemoveCoAuthor` entfernt einen Co-Autor aus der `CoAuthorList`.
- `FillArticleSubmitForm` liest die Werte aller Formfelder aus dem Request und speichert sie in einem Form-Objekt `ArticleSubmitForm`, das sich im CallContext befindet.

#### 5.1.4 Testphase

Die Testphase wurde manuell durchgeführt indem alle modifizierten Workflows mit allen erdenklichen Situationen durchgespielt wurden. Zum Einen musste die korrekte Darstellung aller Daten (vor allem die der Co-Autoren) überprüft werden. Des Weiteren mussten



Abbildung 5.6: Liste der Co-Autoren

die Verzweigungen (branches) auf ihre Korrektheit hin getestet werden. Ein Beispiel einer Liste von Co-Autoren zeigt Abbildung 5.6

Abbildung 5.7 zeigt das Formular, welches dem Benutzer präsentiert wird, wenn er die Liste der Co-Autoren verändern möchte. Der Benutzer hat die Möglichkeit neue Co-Autoren anzulegen oder zu entfernen. Danach kann er seine Veränderungen entweder speichern oder wieder verwerfen. Während des gesamten Ablaufs des Einreichens oder Modifizierens eines Artikels kann auf das gezeigte Formular zugegriffen werden.

## Modify CoAuthor List

**Further Author(s)**

Please enter the email address and the name of a further author, then click "Add CoAuthor" to add the user to the coauthor list. If the user is already registered with the service, the userdata from the database will be used. Otherwise the data you provide will be used and you can decide if the new user will be able to log into the service or not .

Email:

Email (verification):

First name:

Lastname:

Allow new user to log into the service

Lock new user, do NOT allow to log in

Already added CoAuthors:

<input type="checkbox"/>	1. Barney Gumble	brink06@ls05.cs.uni-dortmund.de	locked
<input type="checkbox"/>	2. Bart Simpson	brink09@ls05.cs.uni-dortmund.de	incomplete profile

Abbildung 5.7: Eingabeformular - Co-Autoren

In der Testphase wurde die korrekte Funktionsweise aller Möglichkeiten des Löschens und Hinzufügens, Verwerfens und Speicherns bestätigt.

Im letzten Teil der Testphase wurden die Zugriffe auf die Datenbanken sichergestellt. Hierzu gehören die `roles` so wie die `categories`. Das Entfernen und Erstellen von Assoziationen in der `coauthorarticleid` Tabelle funktioniert, genauso wie die Überprüfung ob Einträge überhaupt gelöscht werden dürfen.

## 5.2 Article Makro - Manage Versions

### 5.2.1 Beschreibung der Aufgabe

Das Gruppenthema 2 besteht in der Entwicklung vom Workflow und der Implementierung des neuen Features „Manage Versions“, das das Löschen und die Wiederherstellung von bestimmten Artikelfiles ermöglicht.

Um die Aufgabe zu lösen, wurde ein neues `ArticleVersion` Makro erstellt, das den neuen Workflow kapselt. Dieses Makro wurde später in das `Article` Makro eingebaut. Bei der Workflowentwicklung wurden sowohl bestehende als auch neuerstellte SIBs benutzt.

Die neuen SIBs sind:

- `BuildArticleVersionInfo`
- `SetImageDir`
- `BuildListFromRequestParams`
- `BuildRemovedArticleVersionList`
- `BuildCurrentArticleVersionList`
- `LoopOverArticleVersionList`
- `AddAVMResultToList`
- `PostRemoveArticleVersion`
- `RemoveArticleVersion`
- `RestoreArticleVersion`
- `PrepareAVCVSRestoreData`

Um die Files aus dem CVS löschen zu können und sie wieder in den Dienst einzupflegen, wurden zwei neue Makros entwickelt und realisiert:

- `RemoveArticleVersionFile`
- `RestoreArticleVersionFile`

Diese wurden in das `ArticleVersion` Makro eingebaut. Für ihre Entwicklung wurden sowohl bestehende als auch neuerstellte SIBs verwendet, die für die Verwaltung von den Files zuständig sind:

- `CVSSaveFileInfoToDisk`
- `CVSSaveFileInfoContent`
- `CVSRestoreFileInfoContent`
- `CVSGetLatestFileVersion`

Um die Daten der gelöschten Datei zu speichern, wurden bestehende Geschäftsobjekte und Administratoren angepasst und neue Datenbanktabellen, Administratoren und Geschäftsobjekte erstellt. Die Action-Spalte der ArticleList wurde um die Aktion 'manage versions' erweitert. Um die neue Aktion an eine Rolle annotieren zu können, wurde die neue Berechtigung F-ART-20 in die `rolepermissions.inc` Datei eingebaut.

### 5.2.2 Modellierungsphase

In diesem Kapitel werden die Aspekte vom UML-Design von den Anfangs- und Zielzuständen mittels Use Case-, Aktivitäts- und Klassendiagramm beschrieben.

Das Use Case Diagramm in der Abbildung 5.8 spiegelt den Anfangsworkflow des „Article Feature“ wieder. Jeder Use Case ist als einzelne Aktion in der Liste von den eingereichten Artikeln dargestellt. Das Diagramm besteht aus sieben Use Cases:

- **Article List** – in der Übersichtstabelle werden alle Artikel angezeigt, die der Benutzer sehen darf. Dabei werden zu jedem Artikel 'ID', 'State', 'Titel', 'Author', 'Type' und die Liste von möglichen Aktionen angezeigt;
- **Remove Article** – ermöglicht das Löschen vom ausgewählten Artikel;
- **Read Article** – zeigt detaillierte Information über den Artikel und erlaubt alle zugehörigen Dateien durch das Anklicken des Links herunterzuladen;
- **Mail To Author** – stellt das Formular zum Schreiben der Email zur Verfügung;
- **Submit Final Report** – stellt das Formular zum Erfassen und Einreichen des Endreports zur Verfügung;
- **Delegate Article** – ermöglicht dem Editorial Office den Artikel an die Reviewer zu delegieren;
- **Modify Article** – ermöglicht dem Editorial Office den Artikel zu modifizieren.

Das „Article Feature“ wurde um drei neue Funktionalitäten erweitert:

- **Manage Versions** – zeigt für den ausgewählten Artikel die Listen von aktuellen und gelöschten Revisionen an;

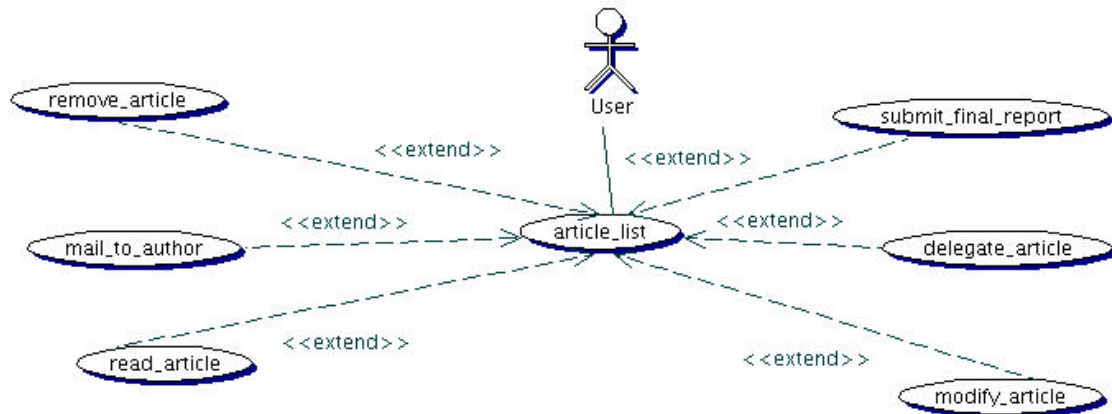


Abbildung 5.8: Use Case „Article Feature“ – Anfangszustand

- **Delete Article Version** – ermöglicht die Revisionen zu entfernen;
- **Restore Article Version** – gibt die Möglichkeit die Revisionen wieder einzupflegen.

Das Use Case Diagramm in der Abbildung 5.9 spiegelt die vorgenommenen Änderungen des Workflows wieder.

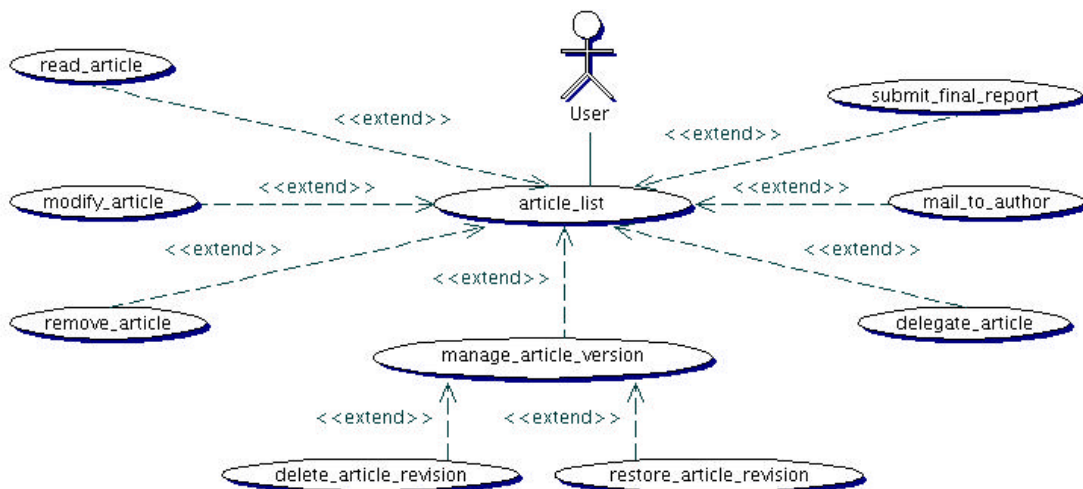


Abbildung 5.9: Use Case „Article Feature“ – Zielzustand

Das Aktivitätsdiagramm in der Abbildung 5.10 beschreibt die unterschiedlichen Schritte, die beim Entfernen von Revisionen notwendig sind.

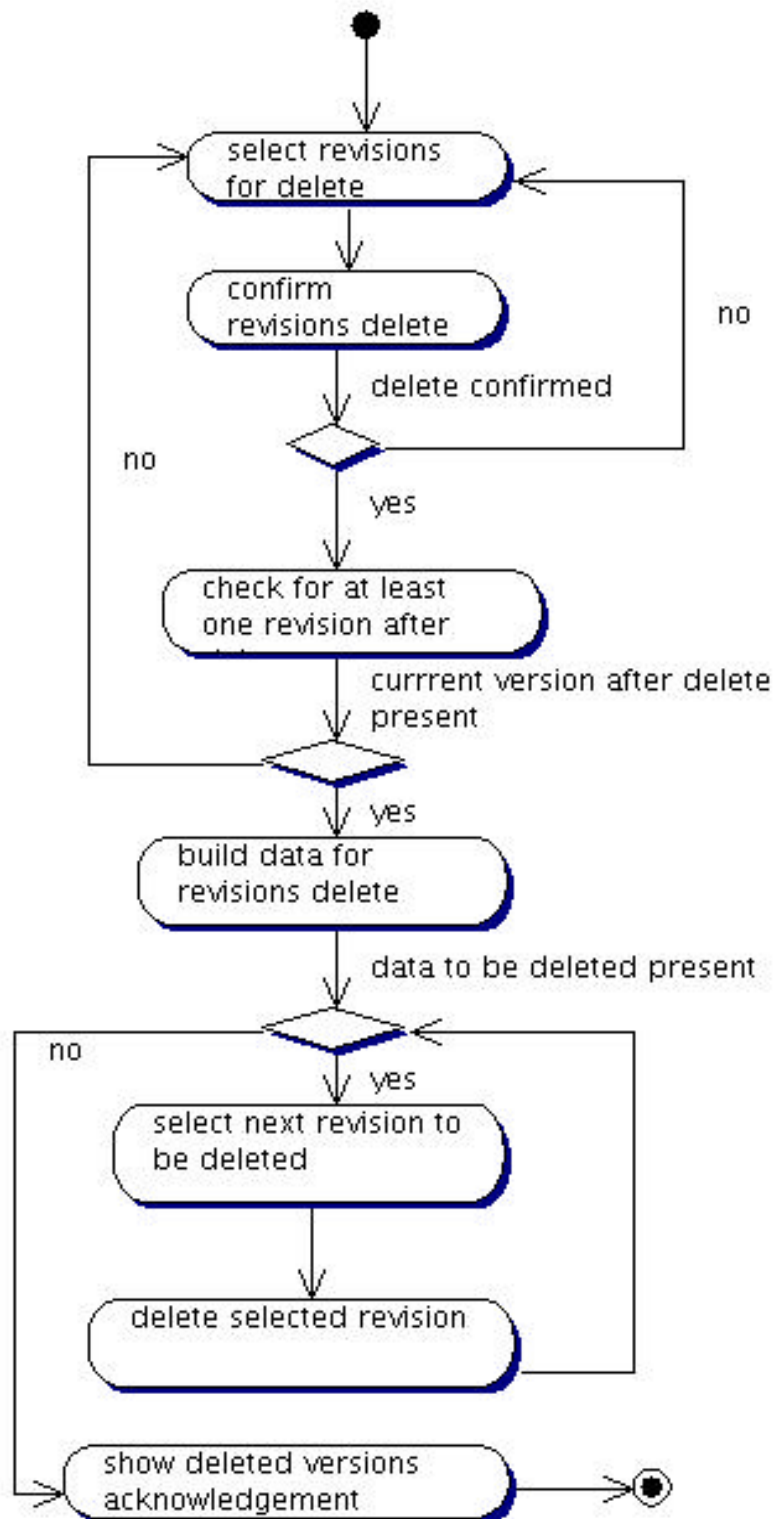


Abbildung 5.10: Activity Chart „Delete Article Version“ – Zielzustand

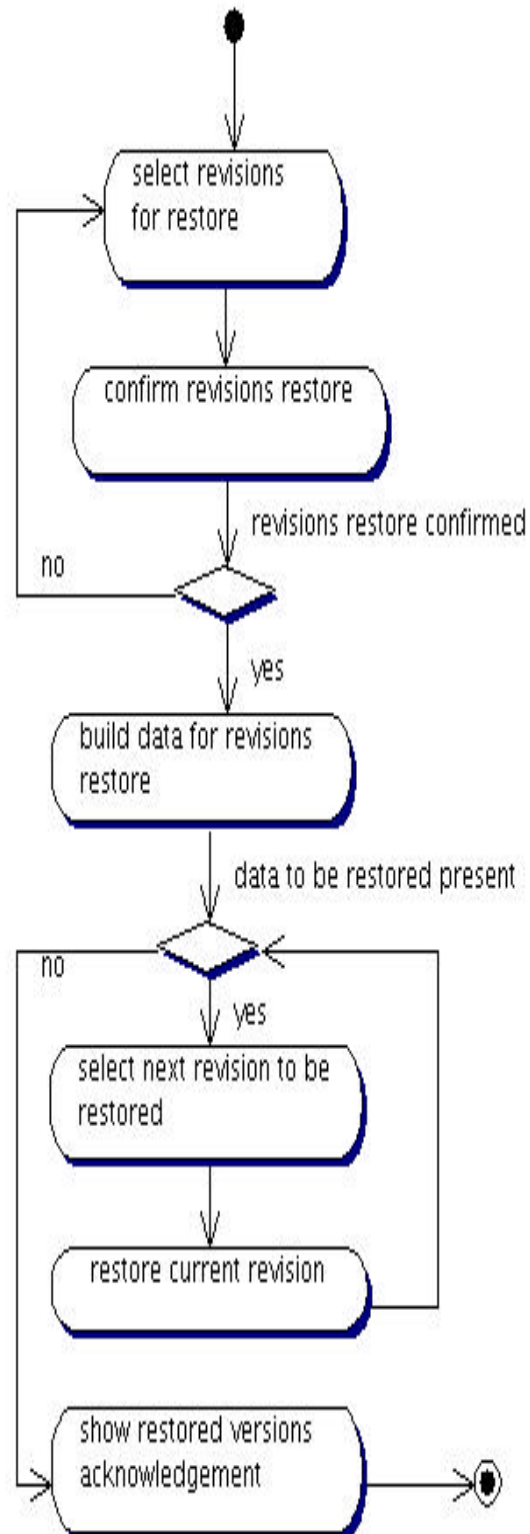


Abbildung 5.11: Activity Chart „Restore Article Version“ – Zielzustand

Das Aktivitätsdiagramm in der Abbildung 5.11 beschreibt die unterschiedlichen Schritte, die bei der Wiederherstellung von Revisionen notwendig sind.

Beim Entfernen des Artikels müssen alle zugehörigen Files auch gelöscht werden (siehe Abbildung 5.12).

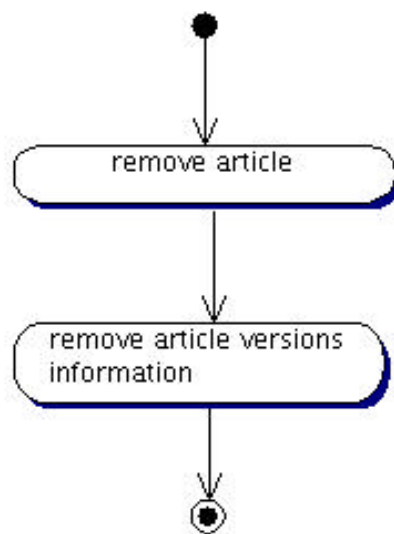


Abbildung 5.12: Activity Chart „Entfernen des Artikels“ – Zielzustand

Um die neue Funktionalität zu realisieren, wurden sowohl bestehende Geschäftsobjekte und Administratoren angepasst als auch neue Datenbanktabellen, Administratoren und Geschäftsobjekte erstellt (siehe Abbildung 5.13).

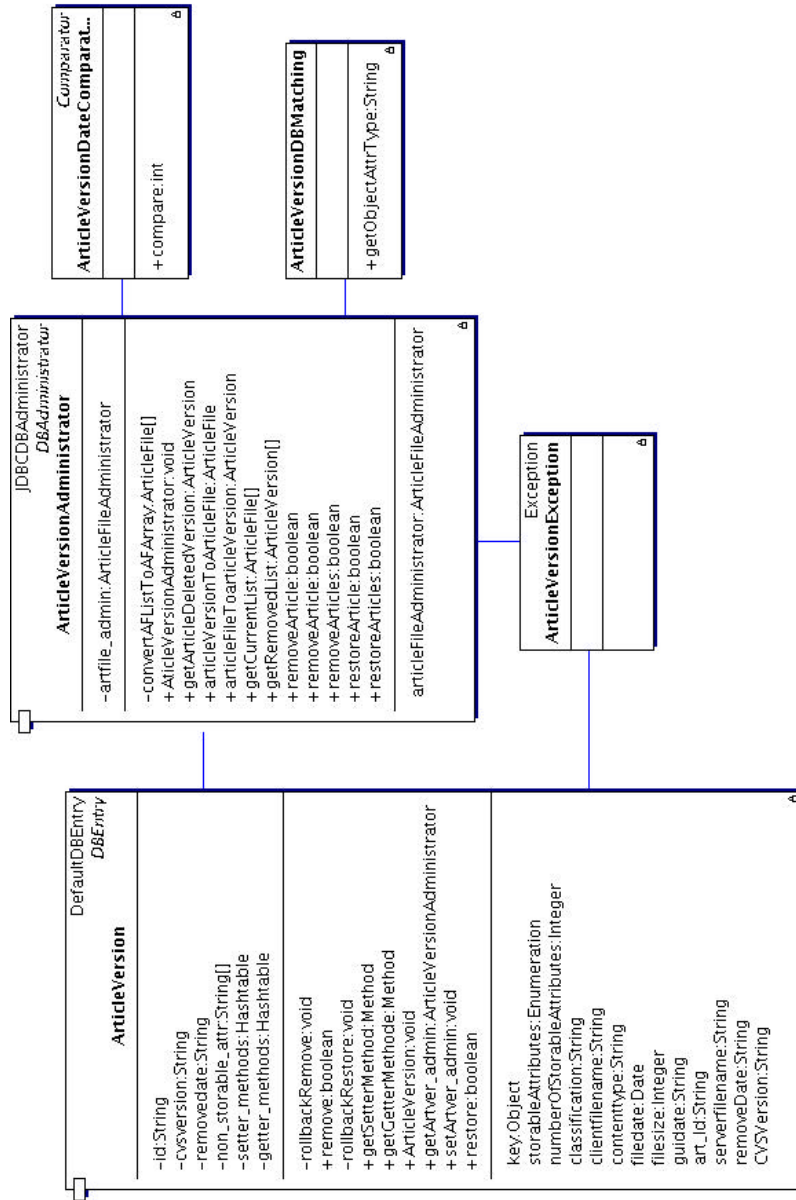


Abbildung 5.13: Klassendiagramm

### 5.2.3 Implementierungsphase

In diesem Abschnitt wird die Implementierung vom neuen Workflow beschrieben. Um das „Manage Versions“ Feature zu realisieren, wurden drei neue Makros entwickelt:

- **ArticleVersion Makro** – kapselt den neuen Workflow, in dem elf neue SIBs eingefügt wurden (siehe Abbildung 5.14)
  - `BuildArticleVersionInfo` – bereitet die Listen von den momentan vorhandenen und von entfernten Dateien für den ausgewählten Artikel vor. Die Daten werden im `CallContext` gespeichert;
  - `SetImageDir` – definiert ein Verzeichnis, unter dem Bilddateien für das Web gefunden werden können;
  - `BuildListFromRequestParams` – erzeugt die Liste von Parameters aus dem `requestParamObject` unter Berücksichtigung des Präfixes;
  - `BuildRemovedArticleVersionList` – bereitet die Liste von `ArticleVersion` Objekten für alle Dateien, die zum Löschen ausgewählt sind, vor;
  - `BuildCurrentArticleVersionList` – erstellt die Liste von `ArticleVersion` Objekten für alle Dateien, die nach dem Löschen von ausgewählten Dateien vorhanden sind. Es wird auch geprüft, ob die Anzahl der gebliebenen Files größer oder gleich `minArtNumberLeft` ist;
  - `LoopOverArticleVersionList` – geht die Liste von `ArticleVersion` Objekten durch und speichert das ausgewählte Objekt im `CallContext` unter einem Schlüssel;
  - `AddAVMResultToList` – fügt das gespeicherte `ArticleVersion` Objekt in die Liste ein;
  - `PostRemoveArticleVersion` – nach dem Löschen des ausgewählten `ArticleVersion` Objektes aus dem CVS werden die Daten für CVS-Version und Entfernungsdatum entsprechend gesetzt;
  - `RemoveArticleVersion` – führt logisches Löschen von `ArticleVersion` Objekt durch und speichert den entsprechenden Dateipfad im CVS;
  - `RestoreArticleVersion` – führt die Wiederherstellung des ausgewählten `ArticleVersion` Objektes auf der Datenbank-Ebene durch;
  - `PrepareAVCVSRestoreData` – bereitet die Daten für die Wiederherstellung des momentan ausgewählten `ArticleVersion` Objektes im CVS vor.
- **RemoveArticleVersionFile** – wird zum Löschen der Files aus dem CVS gebraucht (siehe Abbildung 5.15).

- **RestoreArticleVersionFile** – ist für die Wiederherstellung der Files im CVS verantwortlich (siehe Abbildung 5.16).

Die beiden letzten Makros beinhalten folgende neuimplementierte SIBs:

- CVSSaveFileInfoToDisk
- CVSSaveFileInfoContent
- CVSRestoreFileInfoContent
- CVSGetLatestFileVersion

die für die Verwaltung von den Files zuständig sind.

Der SIB `RemoveArticleVersionData` wird im `Article` Makro eingebaut und ist für das Löschen von allen Daten des Artikels aus der Datenbanktabelle zuständig (siehe Abbildung 5.17).

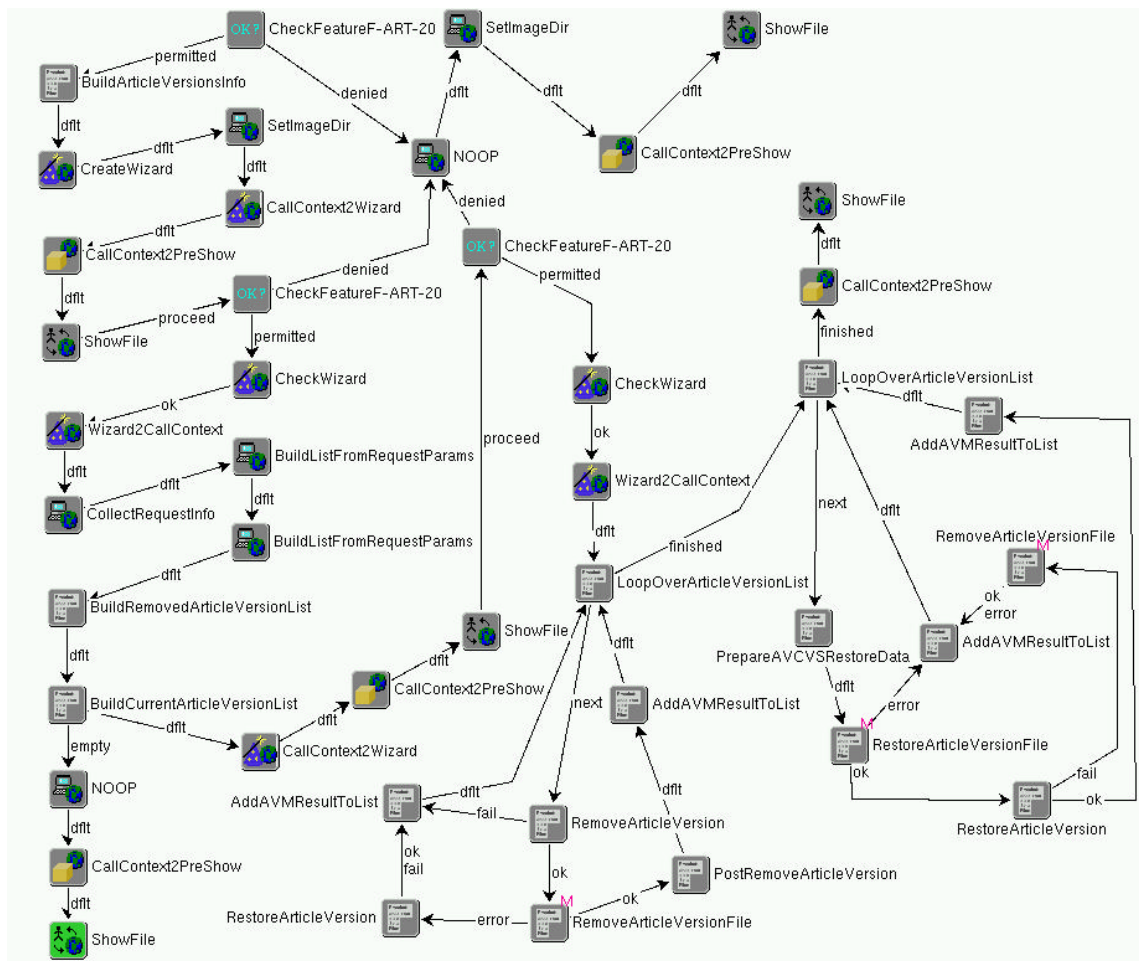


Abbildung 5.14: ArticleVersion Makro

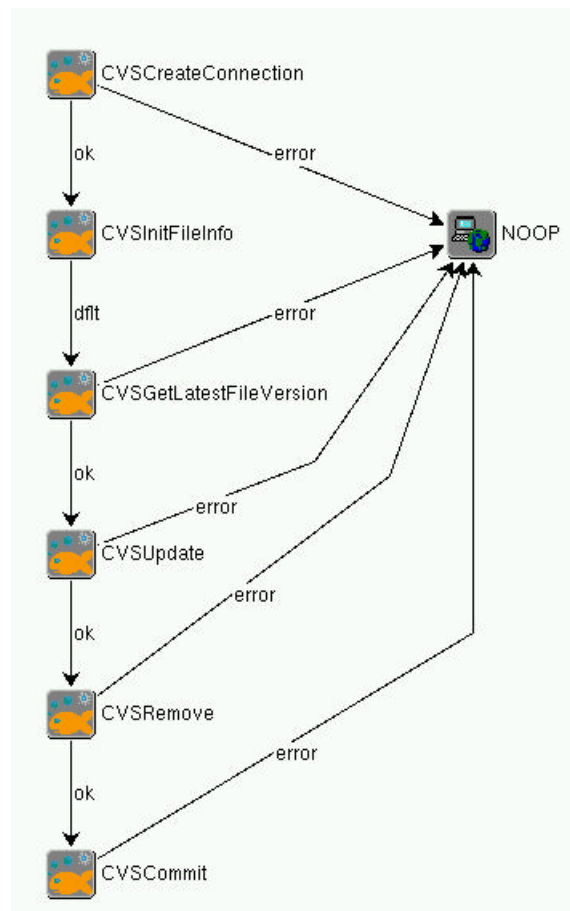


Abbildung 5.15: RemoveArticleVersionFile Makro

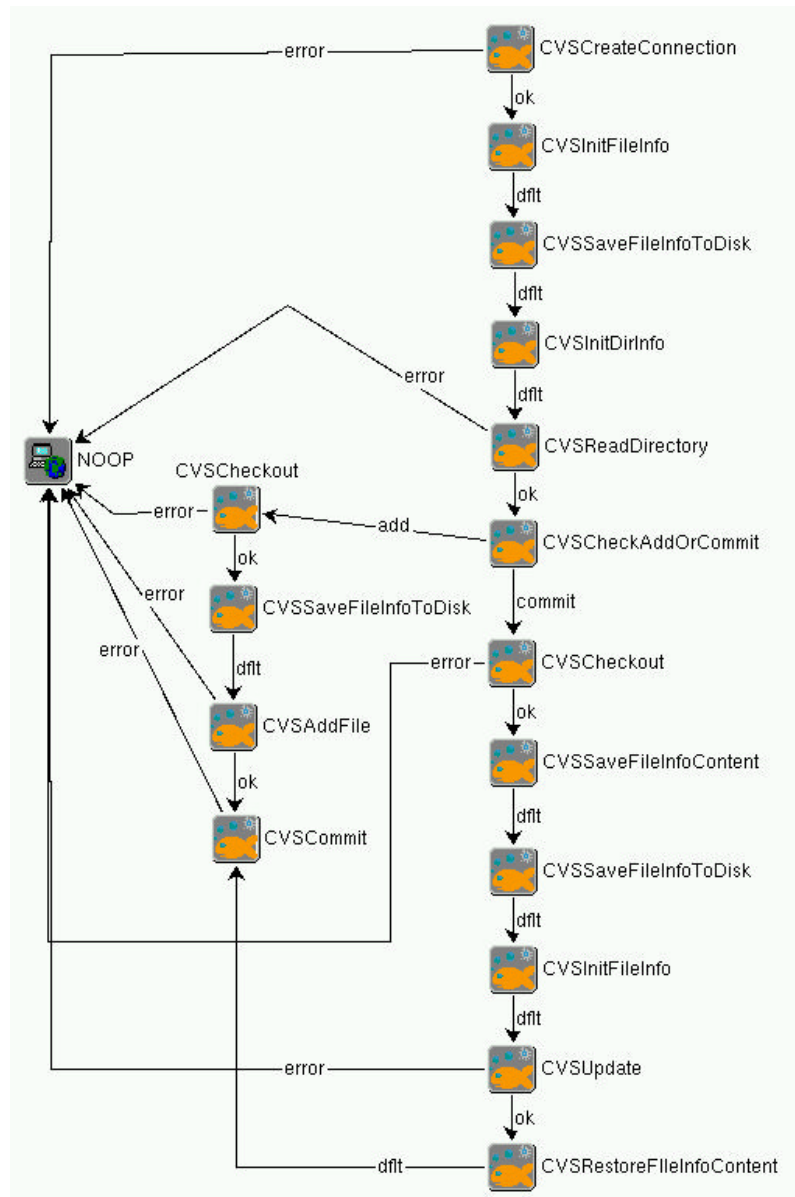


Abbildung 5.16: RestoreArticleVersionFile Makro

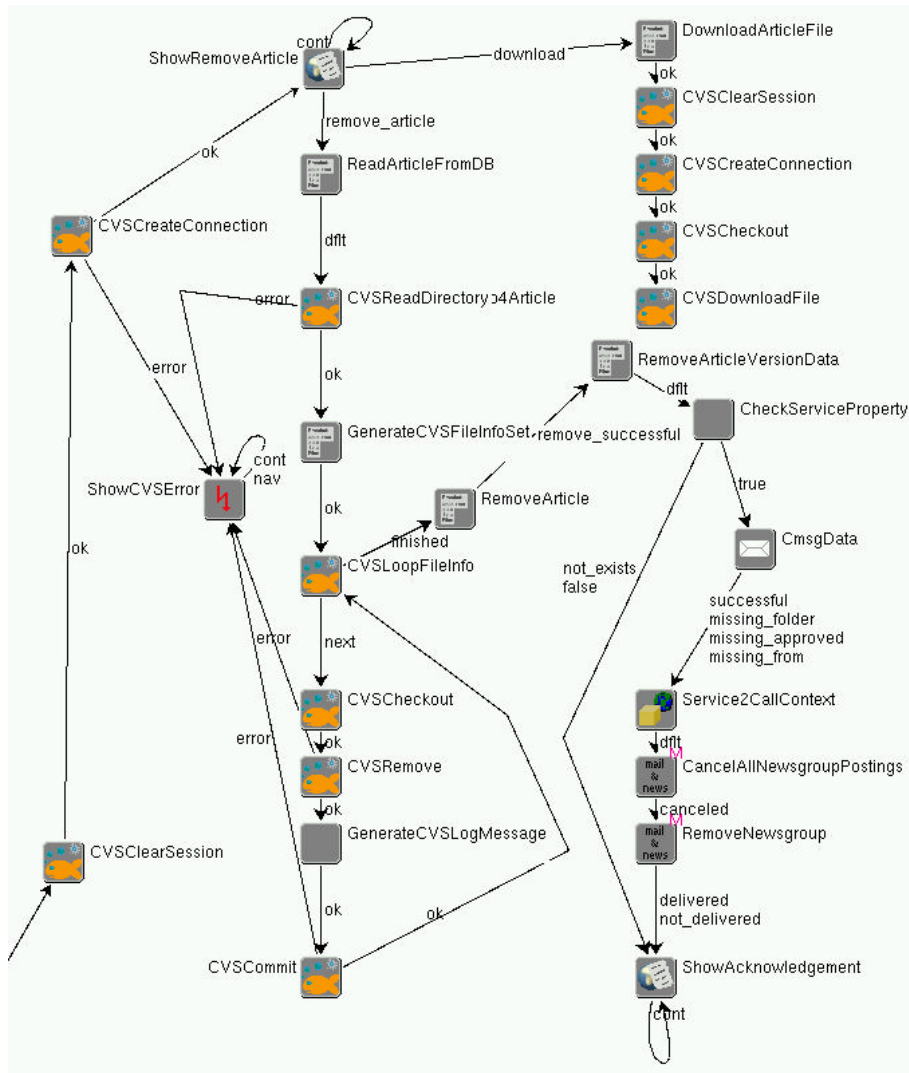


Abbildung 5.17: SIB RemoveArticleVersionData

## 5.2.4 Testphase

Wenn man in der Liste der Artikel (Abbildung 5.18) auf 'manage versions' klickt, gelangt man zu den Tabellen, die in Abbildung 5.19 dargestellt werden. Die obere Tabelle zeigt eine Liste von den vorhandenen Dateien an und die untere Tabelle (diese Tabelle ist hier noch leer) – eine Liste von den gelöschten Dateien. Zu jeder Datei werden noch zusätzliche Informationen wie Dateiname, Dateigröße etc. angezeigt. Jetzt soll eine Datei gelöscht werden. Dafür wird durch das Anklicken der Auswahlbox eine Datei in der oberen Tabelle ausgewählt, z.B. `av.txt` und durch Anklicken des Buttons 'Submit' bestätigt. Nach diesem Vorgang erscheint ein Bestätigungsfenster (Abbildung 5.20), ob die gewählte Datei tatsächlich gelöscht werden soll. Durch Bestätigung mit dem Button 'Submit' wird die Datei gelöscht. Über das erfolgreiche Löschen wird man in einem Fenster (Abbildung 5.21) informiert. Wenn man dieses Fenster durch das Anklicken des Buttons 'Close Window' schließt, gelangt man in das Hauptmenü dieses Features. Die Abbildung 5.22 stellt das Ergebnis dar. Das gleiche Verfahren ist für die Wiederherstellung der Datei gültig. Man wählt in der unteren Tabelle eine Datei aus, die wieder hergestellt werden soll (Abbildungen 5.23 und 5.24).

Mehrfach ausgeführte Testdurchläufe des obigen Workflows haben gezeigt, dass die Implementierung korrekt ist.

List of Articles					
The List of Articles page lists all articles you are permitted to read and work with. The legend below the table explains the content of each column.					
<a href="#">Substitute Submission</a>					
The following table can be sorted according to the following criteria: <a href="#">Article ID</a> , <a href="#">Title</a> , <a href="#">Author</a> , <a href="#">Type</a> . Click on the desired one.					
To get an overview of all article abstracts click here: <a href="#">ArticleAbstracts</a> .					
To get an overview of all articles delegated to you click here: <a href="#">DelegatedArticles</a> .					
<a href="#">Refresh</a> (last refresh: <i>Tue Apr 12 14:56:18 CEST 2005</i> )					
ID	State	Title	Corresp. Author	Further Authors	Action
001		123452345	a, a		• read article • remove article • delegate article • mail to author • manage versions
002		55555	a, a		• read article • remove article • mail to author • manage versions
003	rejected	flgfl	FF, FF		• read article • modify article • remove article • mail to author • submit final report • manage versions
004	accepted	tztjt	FF, FF		• read article • remove article • mail to author • submit final report • manage versions
005	accepted	ndkfmodf	FF, FF		• read article • remove article • mail to author • submit final report • manage versions
006	delegated	dfgdrg	C, C		• read article • modify article • remove article • delegate article • mail to author • manage versions

Abbildung 5.18: Liste der Artikel

### Currently available uploaded files

The table contains a list of current files that are visible to the user and can be deleted.  
Caution, files are not deleted physically, they can be restored at anytime later

Sel.	Original Filename	Server Filename	Size	Date	Content Type	Classification
<input type="checkbox"/>	av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable
<input type="checkbox"/>	pref.epf	001_pref.epf	7 bytes	Mon Feb 07 11:30:01 CET 2005	application/octet-stream	Printable

---

### Available to restore files for this article

The table contains a list of already deleted files, which can be restored at any time.

Sel.	Original Filename	Server Filename	Size	Date	Content Type	Classification	Remove date
------	-------------------	-----------------	------	------	--------------	----------------	-------------

Abbildung 5.19: Dateiliste der Artikel

### Following files are to be deleted

Original Filename	Server Filename	Size	Date	Content Type	Classification
av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable

Abbildung 5.20: Ausführen – Löschen

### Following files were successfully deleted

Original Filename	Server Filename	Size	Date	Content Type	Classification
av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable

Abbildung 5.21: Benachrichtigung über Löschenvorgang

### Currently available uploaded files

The table contains a list of current files that are visible to the user and can be deleted.  
Caution, files are not deleted physically, they can be restored at anytime later

Sel.	Original Filename	Server Filename	Size	Date	Content Type	Classification
<input type="checkbox"/>	pref.epf	001_pref.epf	7 bytes	Mon Feb 07 11:30:01 CET 2005	application/octet-stream	Printable

### Available to restore files for this article

The table contains a list of already deleted files, which can be restored at any time.

Sel.	Original Filename	Server Filename	Size	Date	Content Type	Classification	Remove date
<input type="checkbox"/>	av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable	Tue Apr 12 15:07:14 CEST 2005

Abbildung 5.22: Das Löschen von der Liste von zugänglichen Files des Artikels

### Following files are to be restored

Original Filename	Server Filename	Size	Date	Content Type	Classification	Remove date
av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable	Tue Apr 12 15:07:14 CEST 2005

Abbildung 5.23: Bestätigung des Wiederherstellens

### Following files were successfully restored

Original Filename	Server Filename	Size	Date	Content Type	Classification	Remove date
av.txt	001_av.txt	54 bytes	Fri Jan 28 15:03:14 CET 2005	text/plain	Printable	Tue Apr 12 15:07:14 CEST 2005

Abbildung 5.24: Benachrichtigung über Dateiwiederherstellung

## 5.3 Role Feature

### 5.3.1 Beschreibung der Aufgabe

Die Aufgabe bestand aus dem Problem, einzelnen Benutzern Permissions zu erteilen. Es war nur möglich, Rollen gewisse Permissions zu geben oder zu entziehen. Dies betraf dann sämtliche Benutzer des Dienstes, die diese Rolle hatten. Es wurde das neue Feature F-ROLE-08 eingeführt. Dieses Feature versetzt ein Editorial Office in die Lage, Benutzern einzelne Permissions zu geben. Dazu war das User-Role-Management zu erweitern. Abhängig von F-ROLE-08 erscheint auf der Service Users Seite eine zusätzliche Aktionspalte mit dem Namen `user specific permissions`. In dieser Spalte ist es möglich, einem Benutzer einzelne Permissions zu erteilen oder zu entziehen. Zur Lösung dieser Aufgabe wurde ein Makro mit dem Namen `userpermissions` designt. Dieses Makro enthält den gesamten Workflow des neuen Sub-Features. Außerdem musste die `include-Datei rolepermission.inc` modifiziert und zudem eine neue Datenbanktabelle angelegt werden, welche die benutzerspezifischen Permissions enthält. Des Weiteren wurde ein neues Geschäftsobjekt und ein Administrator implementiert werden, der diese Daten verwaltet.

### 5.3.2 Modellierungsphase

Im nun folgenden Kapitel geht es um die Modellierung des alten und des neuen Workflows bezogen auf die oben genannte Aufgabe. Es wurde mit Hilfe von UML-Diagrammen der alte und der neue Workflow modelliert. Im Vordergrund standen die Aktionen, die ein Editorial Office auf einem Benutzer ausführen kann. Der neue Workflow ist eine Erweiterung des alten mit der neu hinzugefügten Aktion `grant/revoke user specific permissions`.

Abbildung 5.25 und 5.26 zeigen den alten Workflow. Dieser stellt dem Editorial Office fünf Aktionen bereit, mit denen er einen Benutzer des OJS modifizieren kann. Diese Aktionen befinden sich auf der Service Users Seite. Das Aktivitätsdiagramm 5.26 zeigt den Weg, den ein Editorial Office geht, um zur Service Users Seite mit den fünf Aktionen zu kommen. Auf dieser Seite wählt er einen User aus und führt dann eine der fünf Aktionen aus.

Abbildung 5.27 und 5.28 zeigen den neuen Workflow. Das Aktivitätsdiagramm 5.28 modelliert die Aktionen, die vom System und dem Editorial Office ausgeführt werden, wenn das Editorial Office die neue Aktion `grant/revoke user specific permissions` ausführen will.

Diese Aktion kann das Editorial Office nur in Abhängigkeit des neuen Features F-Role-08 ausführen. Ist ihm dieses Feature nicht gegeben, kann er die Aktion `grant/revoke user specific permissions` nicht benutzen.

Das Klassendiagramm in Abbildung 5.29 zeigt die Klassen, die die Daten der neuen Aktion `grant/revoke user specific permissions` und das dazugehörige Feature

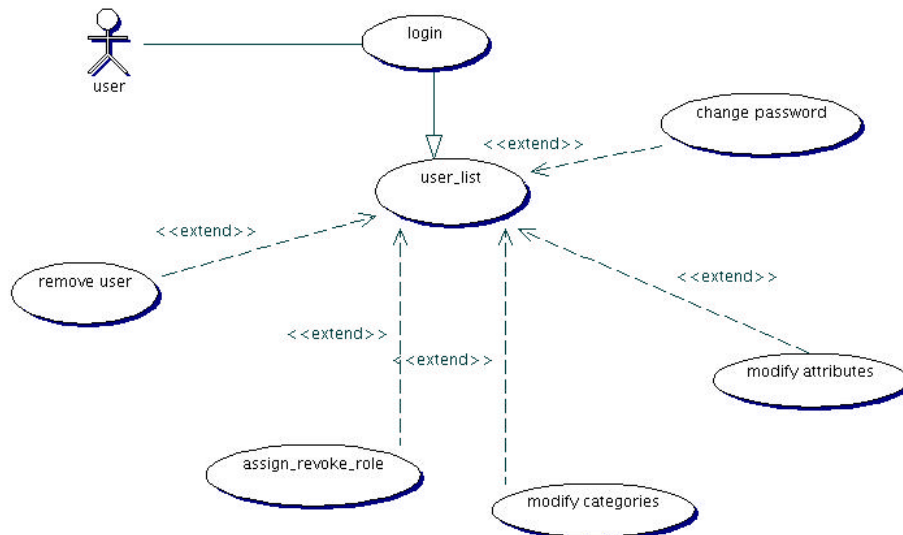


Abbildung 5.25: Use case Diagramm: Ist-Zustand

F-Role-08 verwalten. Die Klassen werden im folgenden näher erläutert.

- `UserPermissionAdministrator` ist die zentrale Klasse. Er bildet die Schnittstelle zur neuen Datenbanktabelle `userpermission`. Er verwaltet die benutzerspezifischen Permissions.
- `UserPermissions` sind Behälterklassen für `PermissionEntries`. Sie besitzen eine Methode, die mit den Parametern `user`, `role` und `permission` den entsprechenden `PermissionEntry` sucht. Außerdem besitzen sie noch Methoden zum Speichern und Zurückgeben von `PermissionEntries`.
- `PermissionEntry` repräsentiert eine Permission, die einem Benutzer gegeben wurde. Jede Permission, die er hat (definiert durch seine Rolle) kann entzogen (revoked) werden. Wenn eine Permission entzogen ist, wird diese von seinen möglichen Features gelöscht. Alle anderen Permissions, die er nicht hat, können bewilligt (granted) werden. Wird eine Permission bewilligt, wird diese seinen möglichen Features hinzugefügt, ungeachtet seiner Rolle. Diese Klasse ist rollenabhängig. Das bedeutet, dass bewilligte/entzogene Permissions für jede Rolle eines Benutzers einzeln definiert werden müssen.
- `DefaultPermissionComparator` vergleicht zwei `PermissionEntry`-Objekte.

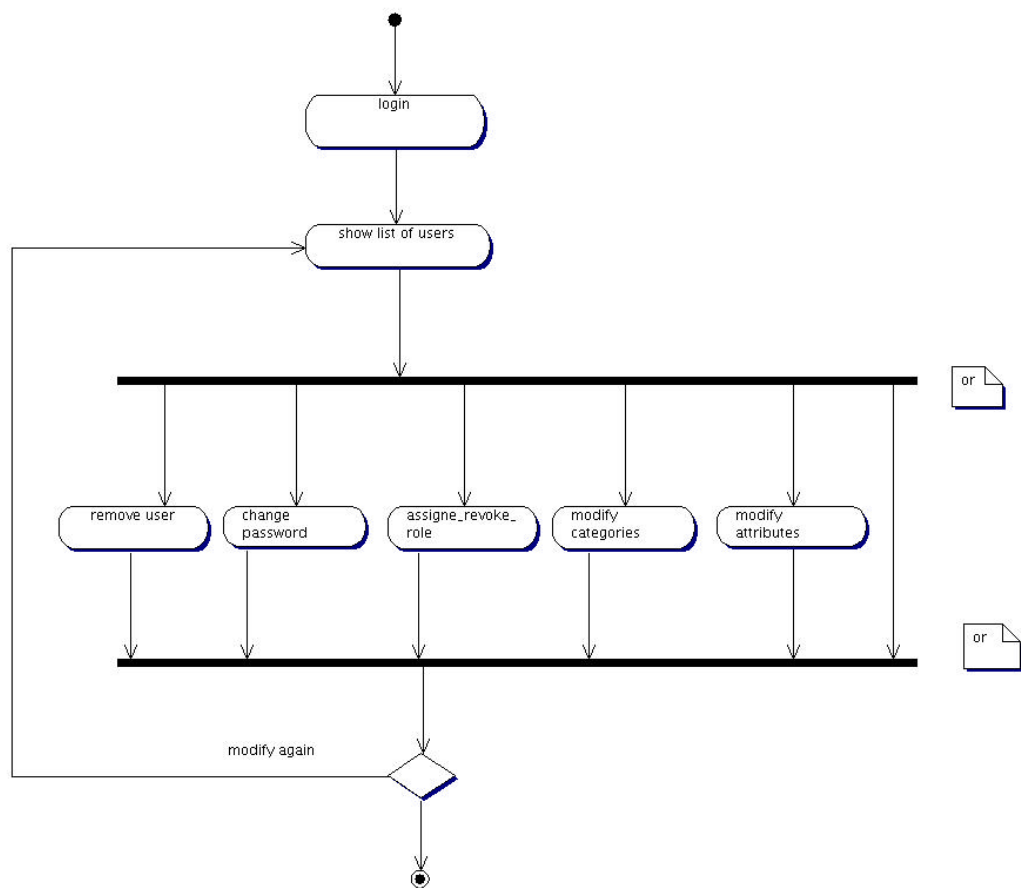


Abbildung 5.26: Aktivitätsdiagramm: Ist-Zustand

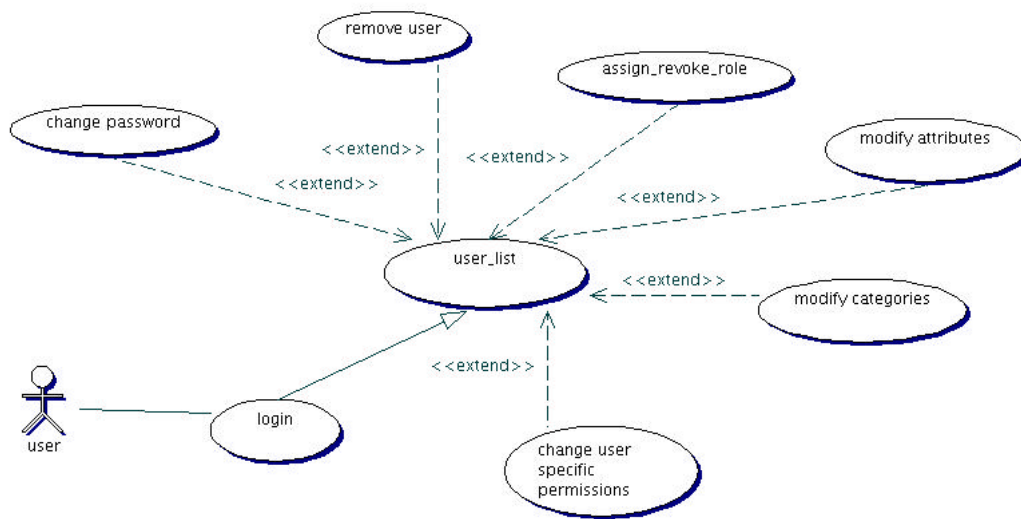


Abbildung 5.27: Use case Diagramm: Soll-Zustand.

### 5.3.3 Implementierungsphase

Im ABC wurde ein neues Makro, `UserPermission`, für das Feature `F-Role-08` implementiert. In diesem Makro wurden vier neue SIBs eingefügt: `GrantRevokePermission`, `UserRoleList`, `UserPermissionList` und `RequestToUserSpecificFeatures`.

Die SIBs `UserRoleList` und `UserPermissionList` werden gebraucht, um mit Hilfe des `User`-Objekts den Benutzer und seine Rollen, insbesondere die zugehörigen Features (Permissions), zu ermitteln. Der SIB `RequestToUserSpecificFeatures` liest die Features aus der `modifyuserperm-cont.html`-Seite aus. Die ausgelesenen Features werden dann in den Call Context gespeichert. Zu guter Letzt werden die Features vom `GrantRevokePermission` SIB unter den Keys `granted_key` und `revoked_key` aus dem Call Context gelesen und dann in der Datenbank gespeichert.

Die SIBs `ShowRoleUserList`, `ShowConfirm` und `ShowUserPermissionList` zeigen die entsprechenden Informationen in einer HTML-Seite an. Diese SIBs selber brauchten nicht programmiert werden, es waren nur entsprechende wml-Seiten zu generieren.

Das neu entwickelte Makro (Abbildung 5.30) wurde in das bestehende `User` Makro integriert. Der SIB `UserActions` in diesem Makro bekam eine neue ausgehende Kante (`grant_revoke_permission`), die zu dem neuen Makro führt. Es ist jetzt möglich mittels dieser Kante zum SIB `UserRoleList` zu kommen. Wenn dieser SIB erreicht wurde sind von jetzt an alle Aktivitäten innerhalb des neuen Makros gekapselt. Durch das SIB `CheckFeaturePermitted` (Abbildung 5.30) wird sichergestellt, dass es nur demjenigen möglich ist in das Makro zu gelangen, dem die entsprechende Berechtigung

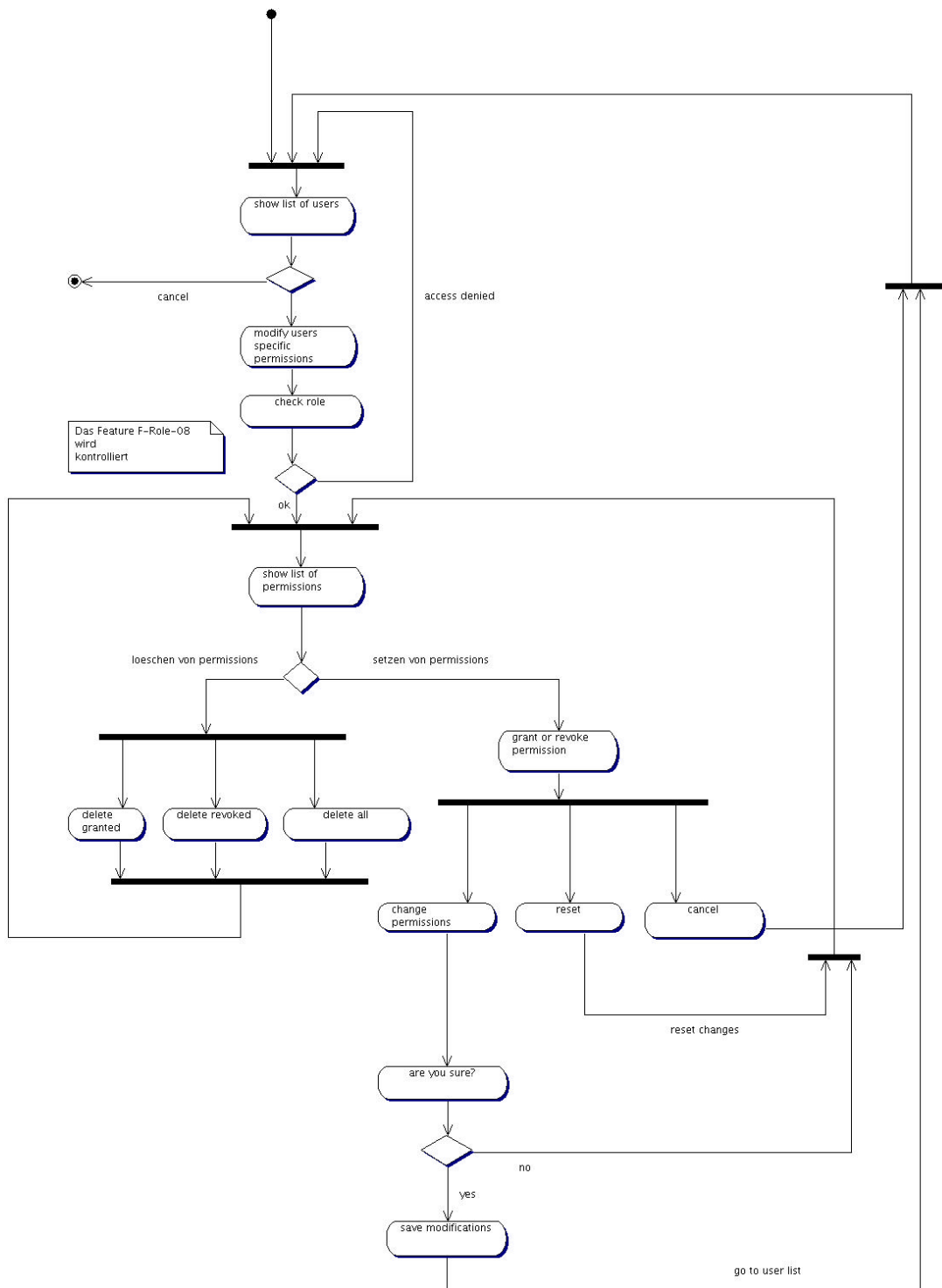


Abbildung 5.28: Aktivitätsdiagramm

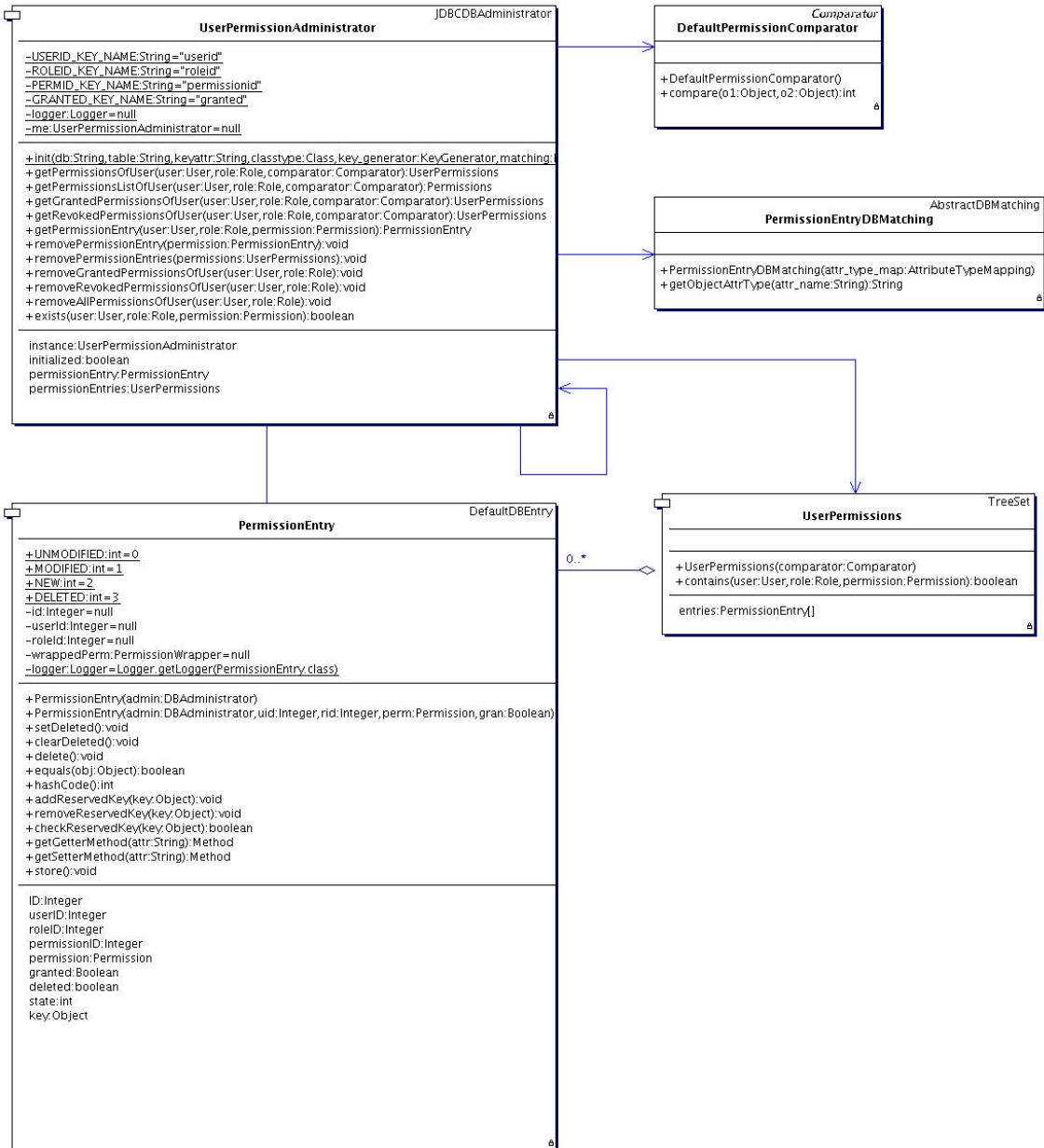


Abbildung 5.29: Klassendiagramm

F-Role-08 zugewiesen wurde.

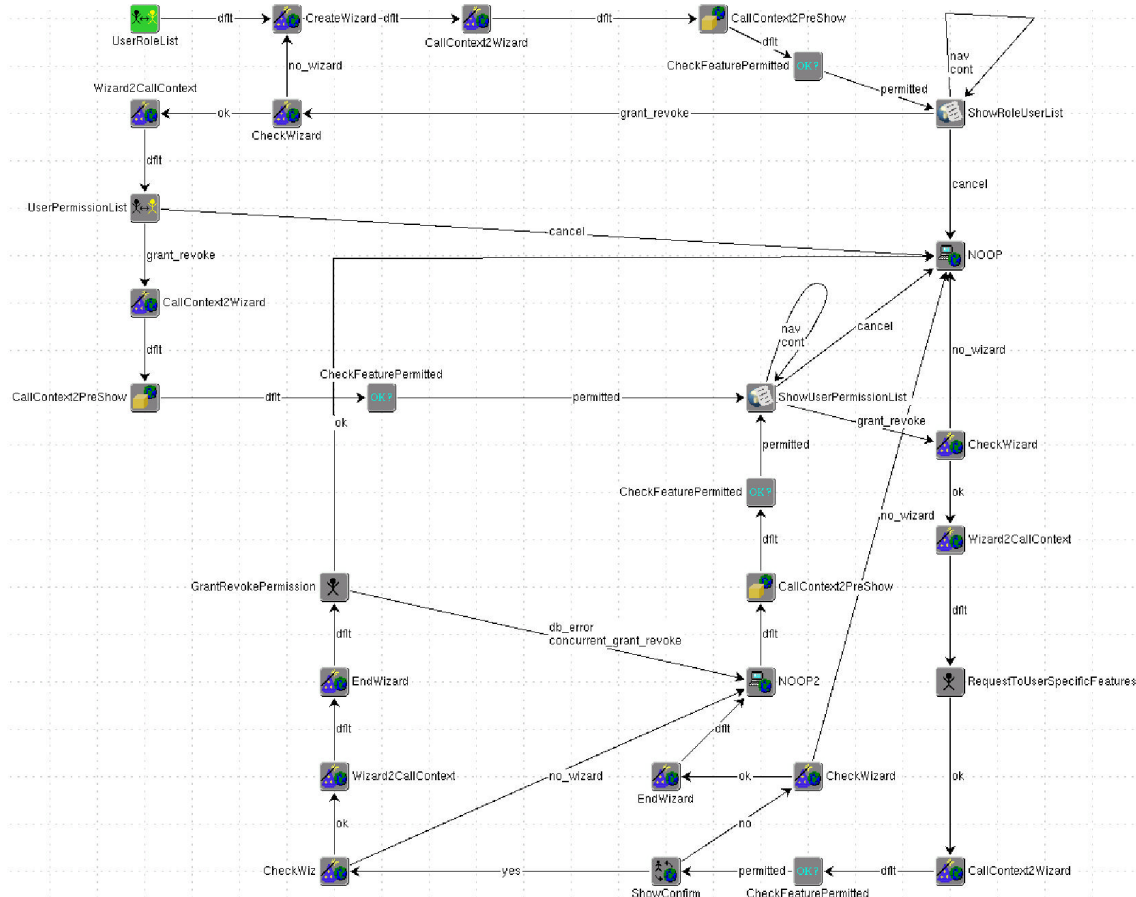


Abbildung 5.30: Makro

Der Startknoten des Makros ist der `UserRoleList` SIB. Der SIB holt sich das Userobjekt und die dazugehörige Rolle mit samt seinen Features. Es wird nur der Benutzer berücksichtigt, der vom Editorial Office in der `ShowUserList`-Seite ausgewählt wurde. Die Daten werden in den Call Context geschrieben und von den Wizards dort gehalten. Die Daten würden ohne die Wizards nach jeder „Show-Seite“ sonst nicht mehr im Call Context enthalten sein.

Der erste SIB, der im Workflow vorkommt, ist der `ShowRoleUser` SIB. Mit diesem SIB wird dem Editorial Office der selektierte Benutzer mit seinen entsprechenden Rollen in einer HTML-Seite angezeigt. Der `UserPermissionList` SIB holt sich die Daten, des ausgewählten Benutzers und die vom Editorial Office ausgewählte(n) Rolle(n) samt deren Features; insbesondere die dem Benutzer speziell betreffenden zusätzlichen Permissions. Diese werden dann mit `ShowUserPermissionList` angezeigt. Der SIB zeigt die Rolle des Benutzers und seine Features, in einer zusätzlichen Tabelle werden die Benutzer

spezifischen Permissions angezeigt. Nur diese zusätzlichen Permissions sind veränderbar, die Features der Rolle können hier nicht verändert werden.

Der SIB `RequestToUserSpecificFeatures` liest die entsprechenden Veränderungen der Permissions aus und leitet sie dann weiter. Der SIB `ShowConfirm` fragt das Editorial Office, ob er die Modifikationen wirklich in die Datenbank schreiben möchte. Wenn er das nicht möchte wird er wieder zur Eingabeseite geleitet. Ansonsten werden die Modifikationen in die Datenbank geschrieben, das übernimmt der `GrantRevokePermission` SIB. Nach der Speicherung wird das Editorial Office wieder auf die `ShowUserList`-Seite verwiesen.

### 5.3.4 Testphase

Getestet wurde von der Service-Users-Seite aus (Abbildung 5.31), die dem eingeloggtten Editorial Office zur Verfügung steht. Durch Anklicken des Links „grant/revoke permission“ wird der Benutzer in das Makro geleitet und er gelangt so zur „Role(s) of User“-Seite (Abbildung 5.32). Dort wird der User und seine Rollen, die er im OJS hat, angezeigt. Durch Auswählen einer Rolle öffnet sich die „Modify User Permission“-Seite (Abbildung 5.33). Es darf auf dieser Seite nur der ausgewählte Benutzer und deren selektierte Rolle angezeigt werden. Die Tabelle auf der rechten Seite zeigt dann die Benutzer speziellen Permissions, die der Benutzer innerhalb dieser Rolle inne hat, an. Das Anklicken zweier Checkboxen (grant und revoke) für dasselbe Feature erzeugt eine Fehlermeldung beim Abspeichern und der Editor wird wieder zur Eingabeseite geführt.

Der Resetbutton im unterem Teil der Seite setzt die Tabelle auf den Ursprungszustand zurück, also den Stand in der Datenbank. Die Seite wurde mit mehreren Benutzern und Rollen getestet, das Anklicken zweier Boxen, sowie das neu Laden der Seite mit dem gleichen Benutzer wurden mehrfach durchgeführt. Durch das neue Laden wurde getestet, ob die Veränderungen im System bestehen bleiben und ob die Permissions entsprechend angezeigt werden. Des Weiteren wurden dementsprechend Versuche unternommen, mit dem User und seiner erweiterten oder beschnittenen Rolle im System zu arbeiten.

Die Datenbank mit der entsprechend neu generierten Tabelle wurde zudem kontrolliert. Entsprechende Eintragungen mussten in den beteiligten Tabellen eingetragen sein. Die Tests verliefen ohne Probleme.

## Service Users

On this page you find an overview of all registered service users.

Give the large number of users, the overview is organized alphabetically according to the lastnames. Clicking on a letter will list the corresponding registered users.

All - [A](#) - [B](#) - [C](#) - [D](#) - [E](#) - [F](#) - [G](#) - [H](#) - [I](#) - [J](#) - [K](#) - [L](#) - [M](#) - [N](#) - [O](#) - [P](#) - [Q](#) - [R](#) - [S](#) - [T](#) - [U](#) - [V](#) - [W](#) - [X](#) - [Y](#) - [Z](#)

Use the appropriate of the following buttons to create new users

- as guest, i.e. in a shortened form(without profile). This is useful e.g to allow new editors or reviewers to be assigned tasks before they enter the service or reviewers to be assigned tasks before they enter the service for the first time, or
- in a complete form, comprising the user's profile.



Click on one of the following roles to get the associated users.

[Administrator](#) [Author](#) [Editor](#) [Editorial Office](#) [Reviewer](#)

Last name	First name	Email	Role(s)	UID	Categories	Action
Bart	Bert	<a href="mailto:hopf02@ls05.cs.uni-dortmund.de">hopf02@ls05.cs.uni-dortmund.de</a>	Author	4	Regular Paper	<ul style="list-style-type: none"> <li>• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a></li> <li>• <a href="#">grant/revoke permissions</a> • <a href="#">lock user</a></li> <li>• <a href="#">modify profile</a> • <a href="#">change password</a></li> <li>• <a href="#">remove user</a></li> </ul>
Egal	Herbert	<a href="mailto:hopf05@ls05.cs.uni-dortmund.de">hopf05@ls05.cs.uni-dortmund.de</a>	Author Reviewer Editor	3	Regular Paper	<ul style="list-style-type: none"> <li>• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a></li> <li>• <a href="#">grant/revoke permissions</a> • <a href="#">lock user</a></li> <li>• <a href="#">modify profile</a> • <a href="#">change password</a></li> <li>• <a href="#">remove user</a></li> </ul>
Hopf	Holger	<a href="mailto:hopf01@ls05.cs.uni-dortmund.de">hopf01@ls05.cs.uni-dortmund.de</a>	Author Reviewer Editorial Office	1	Regular Paper	<ul style="list-style-type: none"> <li>• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a></li> <li>• <a href="#">grant/revoke permissions</a> • <a href="#">lock user</a></li> <li>• <a href="#">modify profile</a> • <a href="#">change password</a></li> <li>• <a href="#">remove user</a></li> </ul>

Abbildung 5.31: Benutzerliste

## Role(s) of User

In this table you can select a role associated to the user. After this you can grant and revoke particular permissions of the role. This modifications concern only the user who has this role. Not the role itself in the system.

Name	Role(s)
Herbert Egal	<a href="#">Author</a> <a href="#">Reviewer</a> <a href="#">Editor</a>

[Cancel](#)

Abbildung 5.32: Liste der Rollen

## Modify User Permission

**On this page the permissions of the indicated user with the given role can be changed. With "grant" and "revoke" the separate features can be added or removed from the allowed features of this user. The changes will take effect only for this user, they neither have effects to the role in the system nor the other roles of the user.**

Name: Herbert Egal  
 Role: Reviewer

### Permissions

Feature Category	Allowed Actions for Role	Grant	Revoke
Role Management	define a new role (F-ROLE-01)	<input type="checkbox"/>	<input type="checkbox"/>
	modify an existing role (F-ROLE-02)	<input type="checkbox"/>	<input type="checkbox"/>
	remove an existing role (F-ROLE-03)	<input type="checkbox"/>	<input type="checkbox"/>
	assign a role to user (F-ROLE-04)	<input type="checkbox"/>	<input type="checkbox"/>
	revoke role from user (F-ROLE-05)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	switch role(F-ROLE-06)	<input type="checkbox"/>	<input type="checkbox"/>
	get a list of all existing roles(F-ROLE-07)	<input type="checkbox"/>	<input type="checkbox"/>
	set user specific permissions(F-ROLE-08)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
User Management	create new user(F-USER-04)	<input type="checkbox"/>	<input type="checkbox"/>
	remove user(F-USER-05)	<input type="checkbox"/>	<input type="checkbox"/>
	modify user attributes (F-USER-06)	<input type="checkbox"/>	<input type="checkbox"/>
	own	<input type="checkbox"/>	<input type="checkbox"/>
	all	<input type="checkbox"/>	<input type="checkbox"/>
	register new user(full profile)(F-USER-08)	<input type="checkbox"/>	<input type="checkbox"/>
	register new user(empty profile)(F-USER-09)	<input type="checkbox"/>	<input type="checkbox"/>
	rlogin(F-USER-10)	<input type="checkbox"/>	<input type="checkbox"/>
	use preferences(F-USER-11)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	get a list of all existing users (F-USER-12)	<input type="checkbox"/>	<input type="checkbox"/>
	this journal	<input type="checkbox"/>	<input type="checkbox"/>
	all	<input type="checkbox"/>	<input type="checkbox"/>
	lock/unlock user account(F-USER-13)	<input type="checkbox"/>	<input type="checkbox"/>
	user overview(F-USER-14)	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.33: Maske zum Modifizieren der einzelnen Rolle

## 5.4 Erweiterung des User Features

### 5.4.1 Beschreibung der Aufgabe

Bisher waren viele Informationen bezüglich eines Benutzers auf mehrere Feature verteilt. Bezogen auf das jeweilige Feature erhält man alle notwendigen Informationen. Da ein OJS-Benutzer viele Informationen hat, muss das Editorial Office unnötig Seiten wechseln, um diese Informationen zu erhalten. Das ist sehr zeitintensiv. Das neue Sub-Feature des USER-Features "F-USER-14" soll dem Editorial Office die Möglichkeit bieten, eine Übersicht eines bestimmten Benutzers einzusehen. Diese Übersicht soll die folgenden Punkte beinhalten:

- Anzeigen des Profiles: Email Adresse, Anrede, Titel, Vorname, Präfix, Nachname, Straße, Postleitzahl, Stadt, Land, Institut, Abteilung, Telefon- und Faxnummer.
- Status des Benutzers
- Rollen
- Kategorien
- Wo ist der Benutzer Staff-Member?
- Konflikte bezüglich der Artikel
- Wo ist er Author oder Co-Author?
- Welche Artikel wurden ihm delegiert, welche Reports sind vorhanden oder fehlen?
- Welchen "Reviewtask" hat er weiterdelegiert? An wem? Status dieser Delegation?

Die nächste Entscheidung, die getroffen werden muss, ist die Daten als CSV-Datei zu exportieren. Die CSV-Datei muss alle Informationen über einen Benutzer enthalten. Des Weiteren soll die "Services Users" Seite nach den existierenden Rollen gefiltert werden können.

### 5.4.2 Modellierungsphase

Mittels Anwendungsfalldiagramm und Aktivitätsdiagramm wurde versucht, die Erweiterungen des neuen Workflows zu verdeutlichen. Der neue Workflow erweitert den alten um die neue Aktion "useroverview" (siehe Abbildung 5.34). Diese Aktion wird dem Editorial Office nur in Abhängigkeit des neuen Features "F-USER-14" gegeben. Das Editorial Office hat zuerst die Möglichkeit die Userliste nach den existierenden Rollen zu filtern und dann den "useroverview" Link anzuklicken oder einfach "useroverview" auszuwählen.

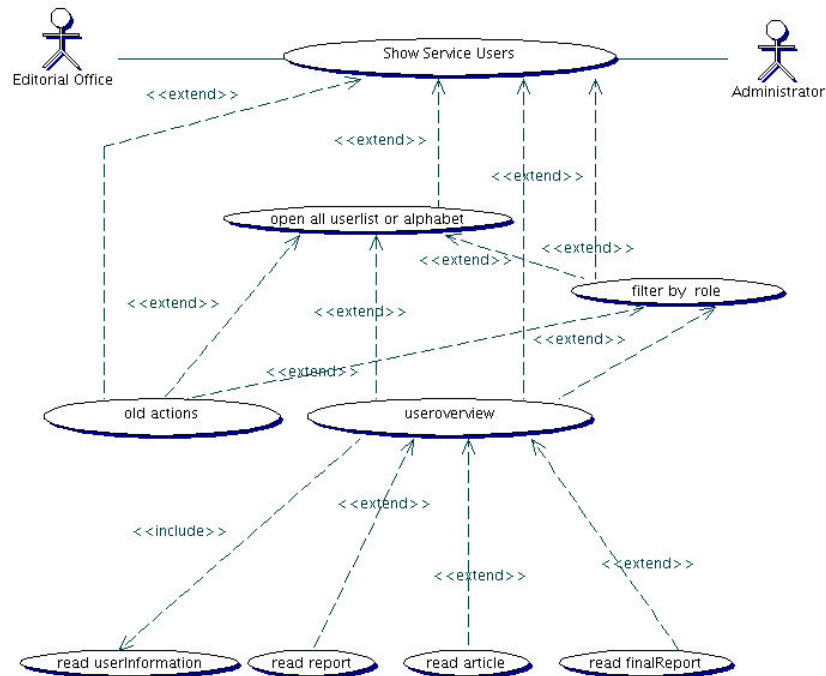


Abbildung 5.34: Das neue Use Case

Die erste Alternative, die Liste nach existierende Rollen zu filtern, ist praktischer, da die Userlist in diesem Fall weniger Einträge enthält. Abhängig davon - durch das Anklicken des "useroverview" Links in der "Service Users" AktionsSpalte - werden die Daten eines Benutzers angezeigt. Diese werden in einer neuen Seite "User Information" gezeigt. Diese Seite enthält die wichtigsten Daten wie Profile, Status des Benutzers, Rollen, Kategorien, Konflikte bezüglich eines Artikels und „, Usertask“. Das Aktivitätsdiagramm beschreibt, wie das Editorial Office die Benutzerdaten lesen kann (siehe Abbildung 5.35).

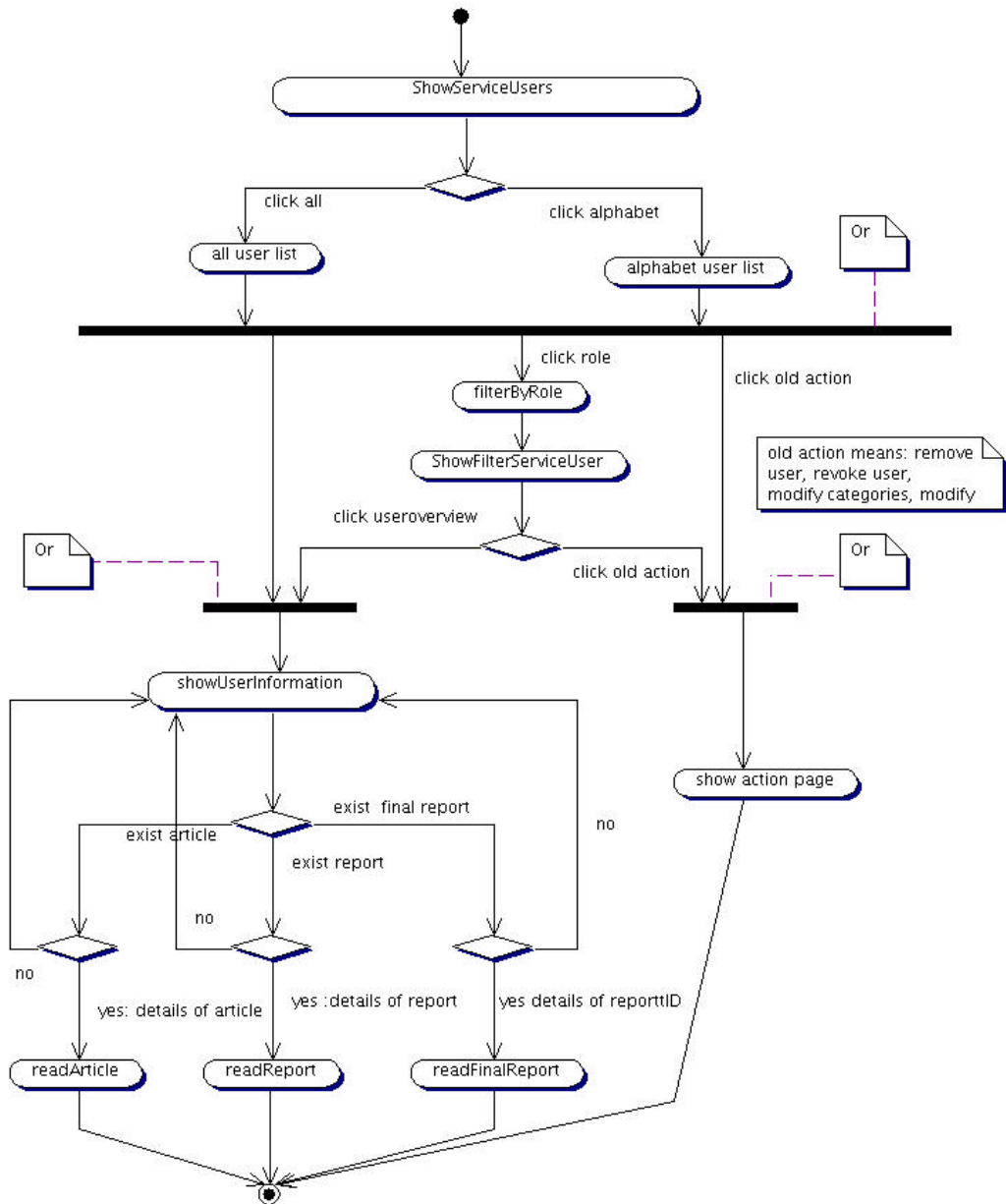


Abbildung 5.35: Das neue Activity Chart

In der Konflikttabelle kann das Editorial Office die Details der Artikel lesen, indem es auf einer ausgewählten ArtikelID klickt. Mit einem "Back"-Button gelangt man wieder auf der "User information" Seite. In der "User tasklist" kann das Editorial Office die Artikel-details lesen, die der Benutzer eingereicht hat oder von denen er Co-Autor ist und es kann sogar Notizen schreiben und speichern. Es kann die Reportdetails lesen, die der Benutzer geschrieben hat. Es kann auch die Finalreportdetails lesen, die der Benutzer zugeschickt hat und auch hier Notizen dazu schreiben. Um die Daten als CSV-Datei zu exportieren, klickt das Editorial Office auf den "CSV-Export"-Button. Es wird eine CSV-Datei erzeugt, in der sich alle Informationen befinden.

### 5.4.3 Implementierungsphase

Das Makro "Users" wurde erweitert, indem ein Sub-Makro "useroverview" eingefügt wurde (siehe Abbildung 5.36).

Dieses Sub-Makro besteht aus mehreren neu geschriebenen SIBs (siehe Abbildung 5.37).

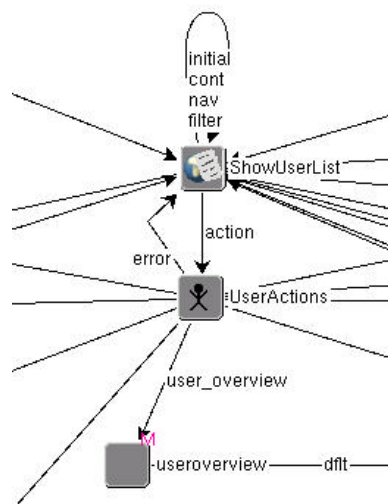


Abbildung 5.36: Das eingefügte useroverview Sub-Macro in Users Macro

Die folgenden Dateien (SIBs und WML) wurden geändert bzw. neu erstellt:

- ShowUserList (geändert).
- UserAction (geändert).
- UserOverview (neu).
- UserOverviewAction (neu).
- UserOverviewCSV (neu).

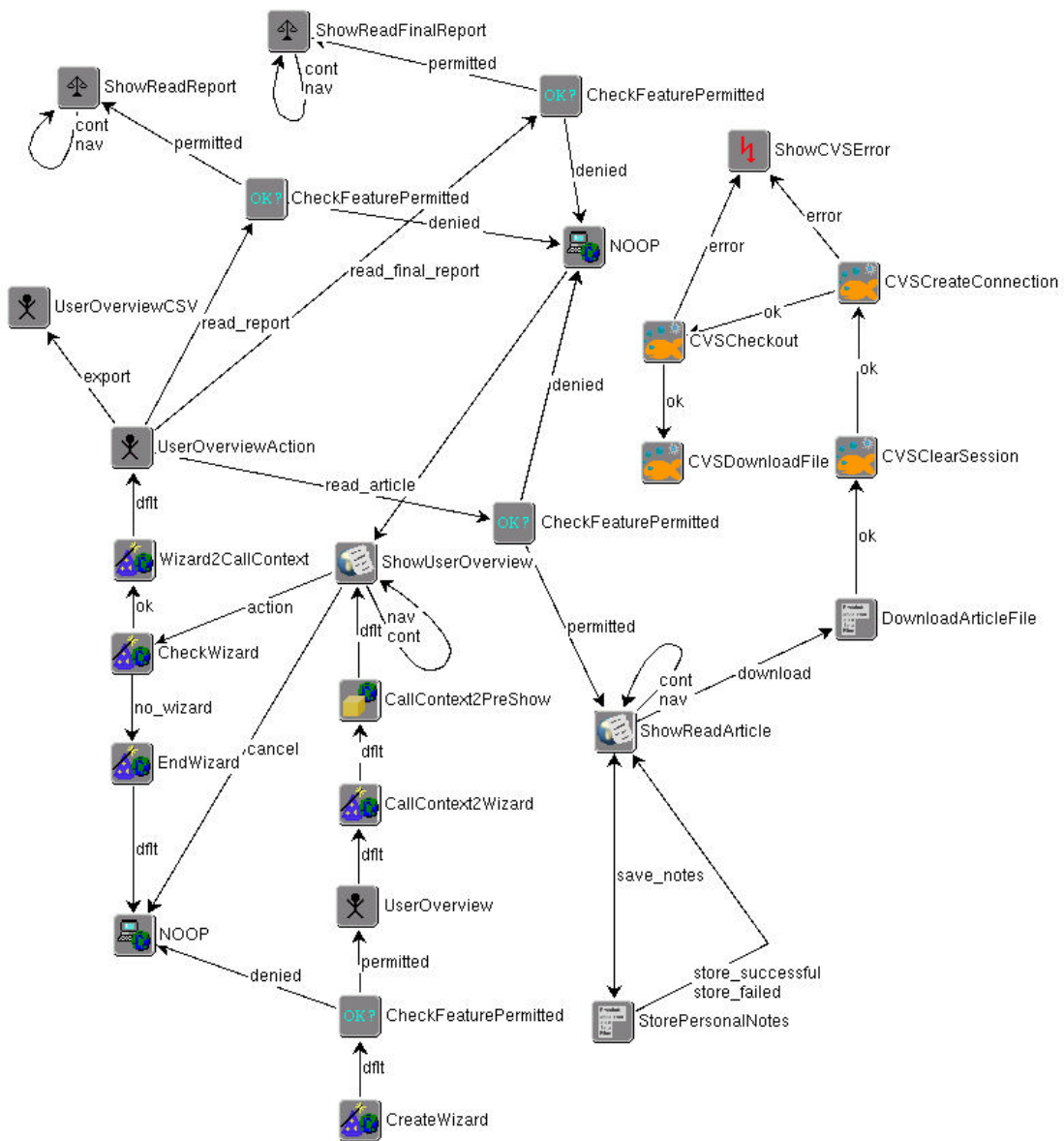


Abbildung 5.37: Das Macro useroverview

- `articledetails-cont.wml` (geändert).
- `finalreportdetails-cont.wml` (geändert).
- `reportdetails-cont.wml` (geändert).
- `userlist-cont.wml` (geändert).
- `useroverview-cont.wml` (neu).
- `rolepermissions.inc` (geändert).

**ShowUserList:** Es wurde die Implementierung des SIBs erweitert, so dass es möglich ist, die Benutzerliste nach der Rolle zu filtern. Dieser SIB zeigt eine HTML-Seite an, von der aus weitere "Framesets" geöffnet werden können.

**UserAction:** Der SIB wurde geändert, so dass die Aktion "user overview" in die Benutzertabelle eingefügt wird.

**UserOverview:** In diesem SIB wird zunächst die UserID aus dem Call Context geholt. Anhand der Administratoren werden die folgenden Objekte erzeugt: user, roles, StaffMemberships, SubmittedFinalReports, AssignedArticles, SubmittedArticles, SubmittedReports, AllReports, DelegatedArticles und ArticleConflicts. Danach werden die Objekte in den Call Context geschrieben.

**UserOverviewAction:** Dieser SIB leitet zu den entsprechenden SIB weiter, in Abhängigkeit der Aktion.

**UserOverviewCSV:** Die Daten bezüglich eines Benutzers werden in einem Array angelegt und dann zum ShowUserOverview weitergereicht. Der SIB ermöglicht dem Editorial Office oder dem Administrator eine CSV-Ansicht. Die Informationen der "User-Information"-Seite werden exportiert und als Vorname\_Nachname\_useroverview.csv gespeichert. Sie wird zum Speichern oder direkt zum öffnen angeboten.

**CallContext2PreShow:** Dieser SIB wird zur Vorbereitung der Daten im PreShow verwendet. Diese Daten werden von dem Call Context geholt und dem SIB "ShowUserOverview" übergeben.

**articledetails-cont.wml:** Die WML-Seite wurde durch ein Back-Button erweitert, um nach dem Lesen der articledetails das Fenster zu schließen.

**finalreportdetails-cont.wml:** Die WML-Seite wurde durch ein Back-Button erweitert, um nach dem Lesen der finalreportdetails das Fenster zu schließen.

**reportdetails-cont.wml:** Die WML-Seite wurde durch ein Back-Button erweitert, um nach dem Lesen der reportdetails das Fenster zu schließen.

**userlist-cont.wml:** Wenn ShowUserList die userlist.html anzeigt, beinhaltet die HTML-Seite alle Rollen außer "root" die im OJS verfügbar sind. Beim Erstellen neuer Rollen werden diese auch angezeigt.

**useroverview-cont.wml:** Aus allen diesen Daten erzeugt die neugeschriebene WML-Datei "useroverview-cont.wml" eine Benutzerübersicht. Die WML-Seite liest die Benutzerdaten aus den Vektoren und den Objekten. Sie zeigt dem Editorial Office die Benutzerdaten zum Einsehen an. Durch den "CSV-Export" Button wird der Download der Daten als CSV-Datei gestartet.

**rolepermission.inc:** Es wurde ein neues Feature in "User Management" eingefügt. Es handelt sich um "F-USER-14". Diese Änderung wird durch ShowUserList angezeigt. Der Link heißt "user overview" und wird beim Anklicken des "User Feature" in der Benutzertabelle ein bzw. ausgeblendet.

#### 5.4.4 Testphase

Der gesamte Workflow wurde getestet. Zu Beginn erfolgt die Modifizierung der Rollen: Das Feature "user overview" (F-USER-14) funktioniert einwandfrei. Man wählt das Feature "F-USER-14" an bzw. ab. Die Änderung wird gespeichert und das jeweilige Feature ein bzw. ausgeblendet. Beim Einschalten des Features "F-USER-14" wird der Link "user overview" innerhalb der Benutzertabelle in der Aktionsspalte angezeigt (siehe Abbildung 5.38).

to get the original list click here: [all-users](#)

[Administrator](#) [Author](#) [CoAuthor](#) [Editor](#) [Editorial Office](#) [Reviewer](#)

Last name	First name	Email	Role(s)	UID	Categories	Action
Alvarez	Ricardo	zaouaou05@ls05.cs.uni-dortmund.de	Reviewer	1	All	• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a> • <a href="#">lock user</a> • <a href="#">modify profile</a> • <a href="#">user overview</a>
Andersson	Nancy	zaouaou03@ls05.cs.uni-dortmund.de	Author Reviewer	2	All	• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a> • <a href="#">lock user</a> • <a href="#">modify profile</a> • <a href="#">user overview</a>
Alofs	Klaus	zaouaou04@ls05.cs.uni-dortmund.de	Author Reviewer	3	All	• <a href="#">assign/revoke roles</a> • <a href="#">modify categories</a> • <a href="#">lock user</a> • <a href="#">modify profile</a> • <a href="#">user overview</a>

Abbildung 5.38: Der Link user overview und die Möglichkeit die Benutzerliste nach Rollen zu filtern

Der Link erleichtert es dem Editorial Office oder dem Administrator Informationen über einen bestimmten Benutzer anzuzeigen (siehe Abbildung 5.39 und 5.40).

Durch das Anklicken des Buttons "CSV-Export" werden die Informationen über den entsprechenden Benutzer in einer CSV-Datei exportiert (siehe Abbildung 5.41).

Im Fall, dass das Editorial Office eine weitere Rolle z.B. Author hat, darf er, trotz seiner Rolle als Editorial Office, nicht die geschriebenen Berichte und Endberichte über seinen Artikel lesen. Wenn der Benutzer zwei Rollen hat sowohl "Reviewer" als auch "Author" und einen Artikel eingereicht hat, dann wird ihm dieser Artikel nicht zugewiesen.

## User Information

**Prof Klaus Alofs**

### Information about the User

#### Profile

Forms of Address: Mr.  
Title: Prof  
First name: Klaus  
Prefix: Ph.  
Last name: Alofs  
Initials: KA  
Email Address: zaouaou04@ls05.cs.uni-dortmund.de  
Institute: Universität Hannover  
Department: Computer Science  
Street: Kamenerstr.  
Zipcode: 56732  
City: Bergkamen  
Country: Germany  
*Phone* 004956234532  
*Fax* 004956234533

#### State

- ◆ idle

#### Assigned Roles

- ◆ Author
- ◆ Reviewer

Abbildung 5.39: Informationen über einen Benutzer

## Categories

- ◆ All

## Staff Memberships

- ◆ Chief: Prof Nancy H. Andersson / Member: Prof Klaus Ph. Alofs (since 2005-01-13)
- ◆ Chief: Ricardo Alvarez / Member: Prof Klaus Ph. Alofs (since 2005-01-17)
- ◆ Chief: Prof Nadja Ch. Benjelun / Member: Prof Klaus Ph. Alofs (since 2004-11-04)
- ◆ Chief: Prof Arthur J. King / Member: Prof Klaus Ph. Alofs (since 2005-01-13)

## Conflicts

Conflicts with articles
<a href="#">014</a>
<a href="#">013</a>

## User Tasks

Submitted Articles	Contributed Articles	Assigned Articles	Delegated Articles	Submitted Reports	Submitted Final Reports
<a href="#">012</a>		<a href="#">001</a> <a href="#">002</a> <a href="#">003</a> <a href="#">005-rv-2</a> <a href="#">006</a> <a href="#">017</a>	<a href="#">017</a>	<a href="#">001-R1</a> <a href="#">002-R1</a> <a href="#">003-R2</a>	

[CSV-Export](#)
[Back](#)

Abbildung 5.40: Informationen über einen Benutzer (Ergänzung)

	A	B	C	D	E	F	G
1	Profile						
2	Email Address:	zaouaou04@ls05.cs.uni-dortmund.de					
3	Forms of Address:	Mr.					
4	Title:	Prof					
5	First name:	Klaus					
6	Prefix:	Ph.					
7	Last name:	Alofs					
8	Initials:	KA					
9	Institute:	Universität Hannover					
10	Department:	Computer Science					
11	Street:	Kamenerstr.					
12	Zipcode:	56732					
13	City:	Bergkamen					
14	Country:	Germany					
15	Phone:	004956234532					
16	Fax:	004956234533					
17							
18	State:	idle					
19							
20	Categories:	All					
21							
22	Assigned Roles:	Author					
23		Reviewer					
24							
25	Staff Memberships						
26		Chief	Member				
27		Prof Nancy H. Andersson	Prof Klaus Ph. Alofs ( Since: 2005-01-13)				
28		Ricardo Alvarez	Prof Klaus Ph. Alofs ( Since: 2005-01-17)				
29		Prof Nadja Ch. Benjellun	Prof Klaus Ph. Alofs ( Since: 2004-11-04)				
30		Prof Arthur J. King	Prof Klaus Ph. Alofs ( Since: 2005-01-13)				
31							
32	Conflicts						
33		Conflicts with articles:					
34		014					
35		013					
36							
37	User Task						
38		Submitted Articles	Contributed Articles	Assigned Articles	Delegated Articles	Submitted Reports	Submitted Final Reports
39		012	001	001	001-R1	001-R1	
40			003	002	002-R1	002-R1	
41			005-IV-2	003	003-R2	003-R2	
42							

Abbildung 5.41: CSV-Ansicht der Benutzerinformation

Die Filterung der Userliste bezüglich einer Rolle wurde erfolgreich getestet (siehe Abbildung 5.38).

# Kapitel 6

## Gruppenthemen 2. Semester

Mit dem Beginn des neuen Semesters wurden wiederum neue Themen verteilt. Diese Hauptthemen waren noch umfangreicher als die ersten beiden Themen. Es folgt eine Übersicht.

### 6.1 Delegation Stack

#### 6.1.1 Beschreibung der Aufgabe

Die Aufgabe bestand darin eine allgemeine Softwarekomponente zu erstellen, die es ermöglicht auf einfache Weise Objekte (z.B. im OJS: Artikel) unterschiedlichen Objekten (z.B. im OJS: Editor, Reviewer) zuzuweisen. Diese zugewiesenen Objekte sollen mit Nutzungsrechten (Permissions) annotierbar sein. Des Weiteren sollte die Tiefe der Zuweisungshierarchie uneingeschränkt sein, d.h. dass ein Objekt schrittweise unbeschränkt weitergeleitet werden kann. Hierdurch kann die im OJS ausgeschaltete Funktion der `SecondlevelDelegation` ermöglicht werden.

Diese allgemein gehaltene Softwarekomponente kann dann im OJS eingesetzt werden und die vorhandene Delegationsfunktionalität ersetzen. Zur Lösung dieser Aufgabe wurde ein neuer Delegation-Workflow modelliert. Das Use Case Diagramm und das Aktivitätsdiagramm zeigen den neuen Workflow, der durch Verwenden des neuen Moduls möglich ist. Ferner wird das Design der Softwarekomponente mittels eines Klassendiagramms aufgezeigt. Nochmals anzumerken ist, dass die volle Funktionalität der entwickelten Softwarekomponente im aktuellen OJS zurzeit nicht genutzt wird, so dass sich der Workflow der Artikeldelegation für die Benutzer nicht verändert hat. Momentan ersetzt das Softwaremodul nur die bisher im OJS integrierte Möglichkeit der Delegation.

Schließlich wurde eine Drei-Schichten-Architektur entworfen mittels derer die allgemeine Softwarekomponente in den OJS eingebaut wurde. Zur Modellierung wurde die Software `Together 6.2` benutzt und die Implementierung der Softwarekomponente erfolgte mit der Entwicklungssoftware `Eclipse 3.1`.

## 6.1.2 Modellierungsphase

Das Use Case Diagramm 6.1 zeigt den Workflow des Editorial Office, so wie er mit dem neuen Modul möglich wäre. Es ist weiterhin das Delegieren so durchzuführen, so wie es im ursprünglichen System schon durchgeführt wurde. Durch Aktivieren des neuen Softwaremoduls bekommt das Editorial Office allerdings jetzt die Möglichkeit Zugriffsrechte zu setzen. Ferner ist es nun möglich die `SecondLevelDelegation` durch Setzen der entsprechenden Permission im System freizuschalten.

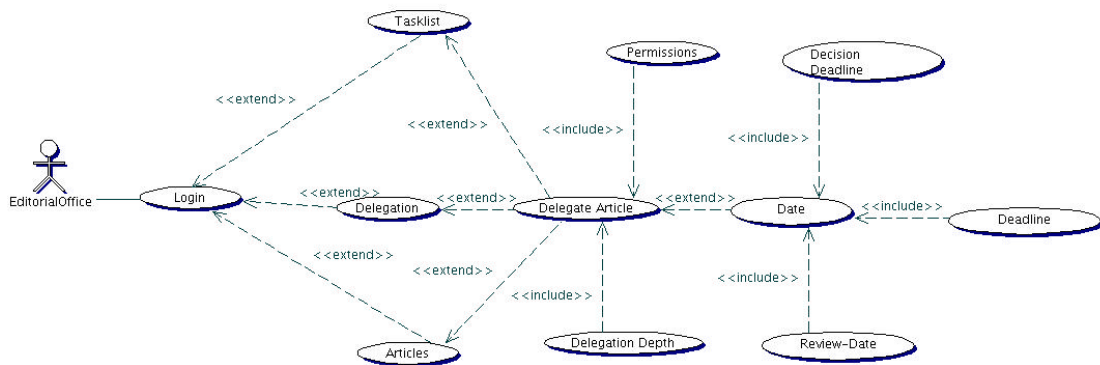


Abbildung 6.1: Use Case Diagramm

Das Aktivitätsdiagramm in der Abbildung 6.2 zeigt den in der neuen Softwarekomponente möglichen Workflow zum Delegieren von Objekten. Im OJS ist diese Funktion allerdings nicht vorgesehen. Die Veränderung, die in das System eingepflegt wurde, zeigt die neue Aktivität `Set Article Permissions`. Der Sendende kann mit Hilfe dieser möglichen neuen Aktion explizit auf den zu delegierenden Artikel Nutzungsrechte setzen wie z.B. die Leserechte und die Anzahl der möglichen Delegationen, die auf einem bestimmten Artikel durchgeführt werden können. Mit der Möglichkeit des Setzens der Tiefe ist es nun möglich im System die `SecondLevelDelegation` freizuschalten.

In der Abbildung 6.3 ist das allgemeine Prinzip der Aufgabenstellung zu sehen. Einem Objekt kann hierbei ein anderes Objekt zugewiesen werden. Diese beiden Objekte können dabei mit Nutzungsrechten belegt werden. Das Weiterleiten eines Objektes ist hierbei unbeschränkt möglich, d.h. die Zuweisungshierarchie wird nicht eingeschränkt.

Abbildung 6.4 zeigt die neue Architektur der Delegation wie sie im OJS integriert wurde. An erster Stelle steht die SIB-Ebene, die das neue `Delegation-Modul` benutzt. Die SIBs, die an der Delegation im OJS involviert sind, mussten an die Klassen, die in der Delegation-Ebene liegen, angepasst werden. Ansonsten sind auf der SIB-Ebene keine

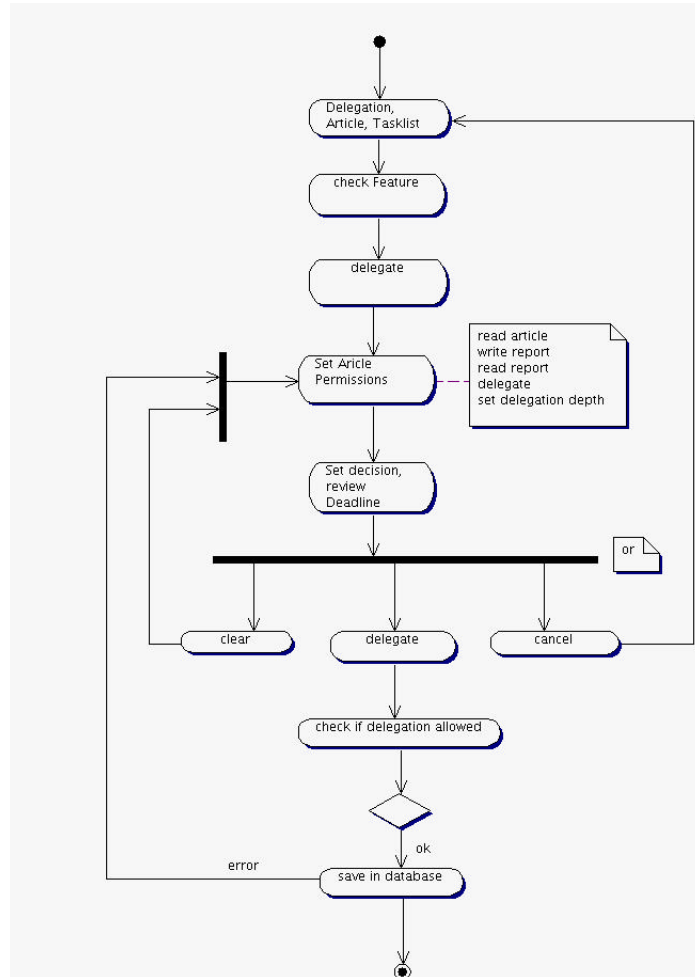


Abbildung 6.2: Aktivitätsdiagramm



Abbildung 6.3: Zuweisung von Objekten

Adaptionen mehr vorgenommen worden.

Die Ebene `Delegation` beinhaltet die neuen, an das System angepassten Klassen, die das neue Modul `Assignment` implementieren. Die Klassen dieser Ebene übernehmen den Datenaustausch zwischen der `SIB`-Ebene, dem `OJS`-System, `Assignment` und dem alten `ArticleDelegationAdministrator`.

Die letzte Schicht beinhaltet das neue Modul `Assignment` und den alten, mit Methoden des `AssignmentManagers` erweiterten `ArticleDelegationAdministrator` des ursprünglichen Systems. Der `ArticleDelegationAdministrator` wurde entsprechend angepasst, um mit `Assignment` die komplette neue und auch alte `Delegation` im `OJS` zu realisieren.

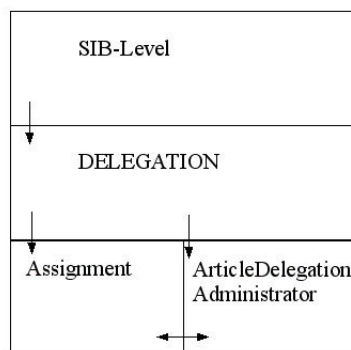


Abbildung 6.4: Architektur der Delegation im OJS

### 6.1.3 Klassendiagramm

Das Klassendiagramm in Abbildung 6.5 zeigt das Design der allgemeinen Softwarekomponente und die konkrete Umsetzung für das `OJS`. Die allgemeine Softwarekomponente besteht aus den `Assignment`-Klassen und Interfaces. Durch die Benutzung von Interfaces ist sie so allgemein gehalten, daß sie überall einsetzbar ist, wo es um die Zuweisung zwischen zwei Objekten geht. Es liegt ein eigenes `JAR`-File für das Modul vor.

Die zentralen Klassen der allgemeinen Softwarekomponente sind der `AssignmentManager`, der `AssignmentGraph` und `Assignment`. Für die Benutzung dieser Softwarekomponente im `OJS` müssen diese Klassen und Interfaces an den `OJS` angepaßt werden. Die Klassen `AssignmentGraph` und `AssignmentManager` werden für den `OJS` durch die Klassen `DelegationStack` und `DelegationManager` erweitert.

Das Interface `Assignment` beschreibt die wesentlichen Elemente einer Zuweisung. Eine Zuweisung besteht aus einem Objekt, das zugewiesen wird, sowie aus einem Sender und einem Empfänger des Objektes. Jede Zuweisung hat ein entsprechendes Datum und, wenn freigeschaltet, Zugriffsrechte, die festlegen was für Rechte ein Empfänger

bezüglich des zugewiesenen Objektes hat. Die Tiefe einer Zuweisung bestimmt, wie weit das jeweilige Objekt weitergeleitet werden darf oder nicht. Die Klasse wird implementiert von `ArticleDelegation`.

`AssignmentManager` ist, wie der Name der Klasse schon vermuten läßt, der Manager der Softwarekomponente. Sie ist ein wichtiger Bestandteil und enthält alle Methoden, die die gesamte Funktionalität der Softwarekomponente darstellen. Der `AssignmentManager` gibt den jeweiligen `AssignmentGraph` zurück. Um die Funktionalität bereit zu stellen greift er auf den `AssignmentAdministrator` zu.

Diese Klasse `AssignmentGraph` bezieht sich konkret auf Zuweisungen eines Zuweisungsobjektes. Sie liefert sämtliche Funktionalitäten bezüglich einer Zuweisung. So stellt sie Methoden bereit, die alle Zuweisungen von einem Sender, alle Zuweisungen zu einem Empfänger sowie eine konkrete Zuweisung von einem speziellen Sender zu einem speziellen Empfänger liefern. Eine weitere Methode liefert alle Zuweisungen eines zugewiesenen Objektes. Ebenso verfügt die Klasse über Methoden zum Löschen und Erzeugen von Zuweisungen.

Diese Klasse `AssignmentAdministrator` ist ein Interface, das die konkreten Assignments speichert und zurückgibt, das gilt für das ganze System. Die Parameter der Methoden müssen nicht alle zugewiesen werden, sie akzeptieren auch NULL-Werte. In der API sind die einzelnen Methoden genau beschrieben. Die Klasse wird für den OJS von `ArticleDelegationAdministrator` implementiert.

Vor dem Hintergrund der Drei-Schichten-Architektur läßt sich sagen, daß die Assignment Ebene, durch die für den OJS abgeleitete Delegation Ebene, verdeckt wird.

Klassendiagramm 6.6 zeigt die Klassen, die zur Implementierung der Zugriffsrechte notwendig sind und den angepassten `ArticleDelegationAdministrator`. Der `ArticleDelegationAdministrator` übernimmt weiterhin die Hauptaufgaben und verdeckt die Klassen `DelegationPermissionAdministrator` und `ArticleDelegationPermission`.

Zum Anlegen von Permissions muss der `DelegationPermissionAdministrator` aufgerufen werden. Das Holen der Permissions ermöglicht die Klasse `ArticleDelegation`. Die Funktionen der Permissions können dadurch implentiert werden, aber sie sind in der aktuellen Version des OJS nicht eingebunden.

#### 6.1.4 Implementierungsphase

Das beschriebene Assignment-Modul wurde als eigene Software-Komponente entwickelt. Dazu kamen das Delegation-Modul und Änderungen am

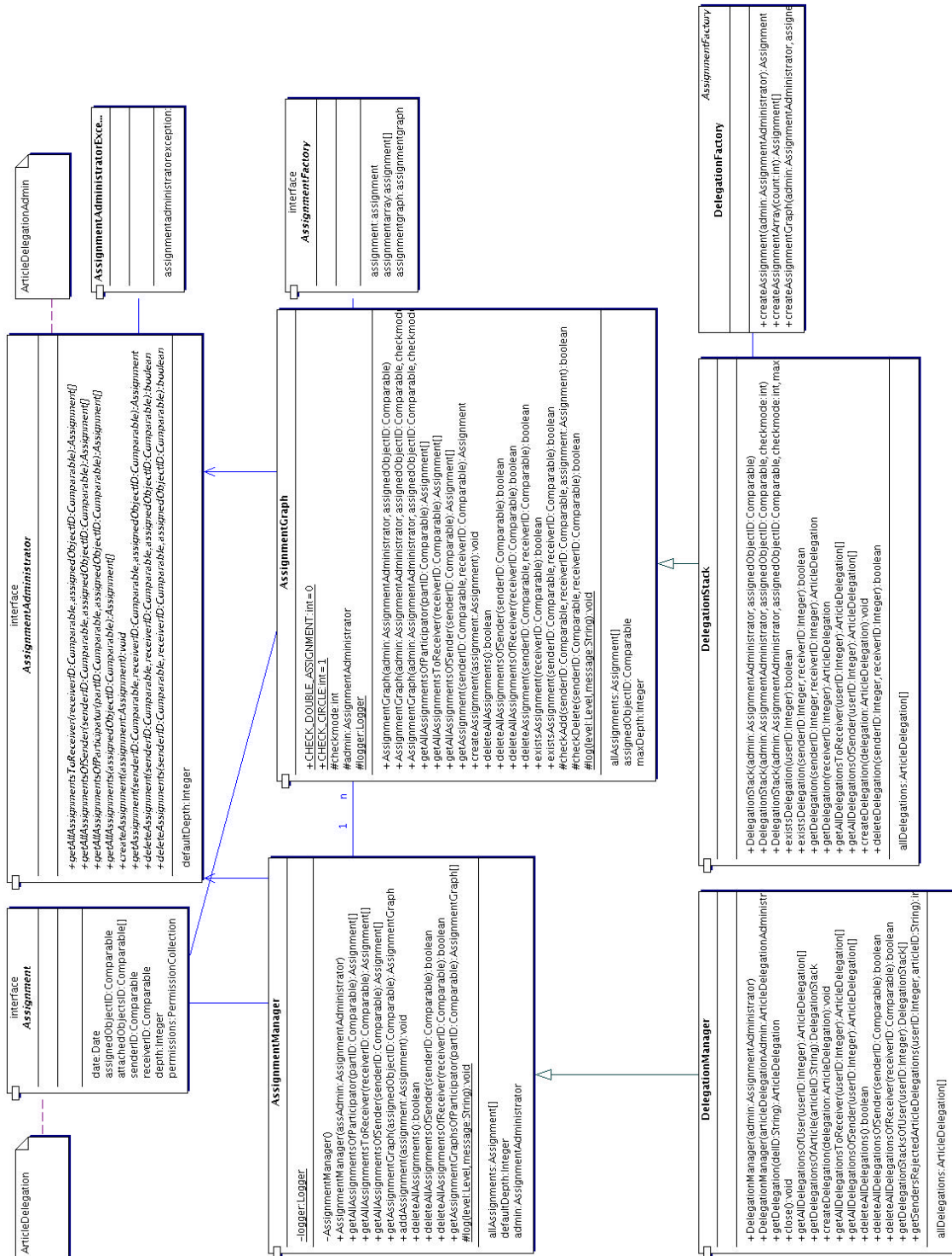


Abbildung 6.5: Die an den OJS angepassten Klassen

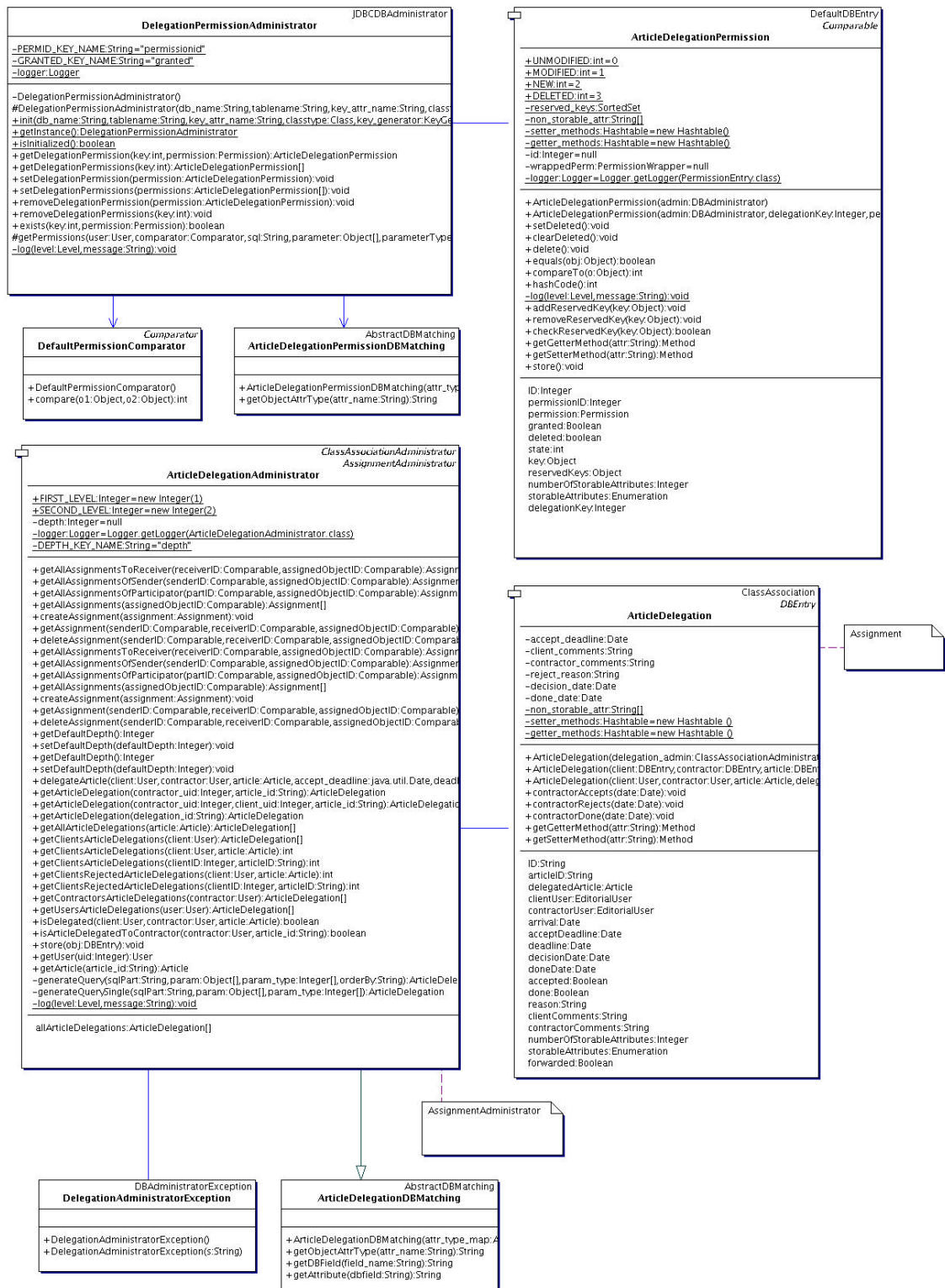


Abbildung 6.6: Allgemeine und die an den OJS angepassten Klassen

`ArticleDelegationAdministrator` und seinen Klassen. Auf SIB-Ebene wurden lediglich Anpassungen an die neuen Funktionalitäten vorgenommen. Durch die Integration des neuen Moduls hat sich an der Delegation für den Benutzer nichts geändert. Die Modifikationen am System laufen im Hintergrund. Es sollte durch das neue Modul nur die Möglichkeit geschaffen werden die `SecondLevelDelegation` im OJS zu ermöglichen. Die Funktion kann jetzt je nach Bedarf freigeschaltet werden.

### 6.1.5 Testphase

Die Abbildung 6.7 zeigt die Liste der Artikel, aus der heraus delegiert werden kann. Der Benutzer muss als Editorial Office eingeloggt sein. Beim Anklicken der Aktion delegieren erscheint die Seite `Delegate Article`, siehe Abbildung 6.8. Diese Seite zeigt die möglichen Benutzer an, an die ein Artikel delegiert werden kann. Staff-Member sind die Benutzer des OJS, denen Artikel zugesandt werden können. Es wurde in der Testphase kontrolliert, ob die Delegationen zwischen dem Editorial Office und den Staff-Membren funktioniert. Entsprechende Einträge, die vom System daraufhin vorgenommen werden müssen sind auf Korrektheit überprüft worden.

Die Reviews 6.9 sind ebenfalls ein Bestandteil der Delegation und mussten zudem auf korrektes Verhalten im System kontrolliert werden.

Zum Einen musste für jede Art der Delegation die korrekte Darstellung aller Daten (vor allem die in der Datenbank vorhandenen) überprüft werden. Des Weiteren müssen die an der Delegation beteiligten SIBs auf korrektes Verhalten hin getestet werden. Im letzten Teil der Testphase wurden die Zugriffe auf die Datenbanken sichergestellt. Die Arbeitsweise des modifizierten `ArticleDelegationAdministrator` wurde besonders kontrolliert. Das Delegieren funktionierte genauso wie zu der Zeit als die neue Komponente noch nicht integriert war.

## List of Articles

The List of Articles page lists all articles you are permitted to read and work with. The legend below the table explains the content of each column.

### Substitute Submission

The following table can be sorted according to the following criteria: [Article ID](#), [Title](#), [Author](#), [Type](#). Click on the desired one. To get an overview of all article abstracts click here: [ArticleAbstracts](#). To get an overview of all articles delegated to you click here: [DelegatedArticles](#).

- There are two possibilities to submit a report: Either online with clicking on "submit report" or with help of a xml file by using the "XML Report" action from the taskactions .
- If you like to submit several reports with help of a xml file, you should click on "Import XML Reports".

**Refresh** (last refresh: *Fri Jul 08 13:45:30 CEST 2005*)

ID	State	Title	Corresp. Author	Further Authors	Action	Type	ID
001	under review	A Financial Test	Weber, Willi		<a href="#">read article</a> • <a href="#">modify article</a> <a href="#">remove article</a> • <a href="#">delegate article</a> <a href="#">mail to author</a> • <a href="#">manage versions</a>	Financial	001
002	under review	New Science Economy	Weber, Willi		<a href="#">read article</a> • <a href="#">modify article</a> <a href="#">remove article</a> • <a href="#">delegate article</a> <a href="#">mail to author</a> • <a href="#">manage versions</a>	Economic	002
003	under review	Astronomy for Kids	Weber, Willi		<a href="#">read article</a> • <a href="#">modify article</a> <a href="#">remove article</a> • <a href="#">delegate article</a> <a href="#">mail to author</a> • <a href="#">manage versions</a>	Astronomy	003
004	under review	Dry Ice in Medicine	Schredder, Herbert		<a href="#">read article</a> • <a href="#">modify article</a> <a href="#">remove article</a> • <a href="#">delegate article</a> <a href="#">mail to author</a> • <a href="#">manage versions</a>	Medicine	004

Abbildung 6.7: Liste der Artikel

## Delegate Article

### Article Information:

<b>Article ID:</b>	002
<b>Title:</b>	New Science Economy
<b>Author:</b>	Weber, Willi

### Delegation to Staff Member(s)

You can do 3 new delegations.

Delegate	Name	Deadline Accept/Reject (dd.mm.yyyy)	Deadline Review (dd.mm.yyyy)	Comments (optional)
<input checked="" type="checkbox"/>	Otto Mueller			
<input type="checkbox"/>	Bell Mueller	13.07.2005	01.01.2006	Please review this paper and submit a report about it.
<input type="checkbox"/>	Alexa Schmidt	13.07.2005	01.01.2006	Please review this paper and submit a report about it.

Abbildung 6.8: Delegation an die Staff-Member

## Overview of Delegated Reviews

This page gives an overview of the delegated reviews, i.e. when and to whom you have delegated a review of an article. So you can get the information about the "position" of articles in the reviewing process.

The legend at the bottom of this page explains the content of each column.

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

**Refresh** (last refresh: *Fri Jul 08 13:36:34 CEST 2005*)

ID	Delegated		Status(Since)	Decision Deadline	Task Deadline	Action	Comments	Type	ID
	from	to							
001	JD	Otto Mueller	review task accepted(on 2005-07-05)		2006-09-03	• extend task • remove delegation	Please review this paper and submit a report about it.	Financial	001
001	JD	Alexa Schmidt	review task rejected(on 2005-07-05), reason: <i>cannot</i>			• delegate article • remove delegation	Please review this paper and submit a report about it.	Financial	001
002	JD	Helen Hunter	review task accepted(on 2005-07-05)		2005-08-02	• extend task • remove delegation	Please review this paper and submit a report about it.	Economic	002
002	JD	Otto Mueller	review task accepted(on 2005-07-05)		2005-08-02	• extend task • remove delegation	Please review this paper and submit a report about it.	Economic	002
003	JD	Helen Hunter	review task accepted(on 2005-07-06)		2005-08-02	• extend task • remove delegation	Please review this paper and submit a report about it.	Astronomy	003

Abbildung 6.9: Delegierte Reviews

## 6.2 Reviewer Allocation Feature

### 6.2.1 Beschreibung der Aufgabe

Das Gruppenthema 2 besteht in der Entwicklung von dem Workflow und der Implementierung des neuen Features „Reviewer Allocation“, das es einem Editor ermöglicht, zu jedem ihm zugeteilten Artikel eine Liste von Reviewern einzugeben. Die Listen müssen vom Dienst verwaltet werden und dem Editorial Office als Grundlage für die eigentliche Verteilung der Gutachten dienen.

Das Feature kann in drei logische Schritte aufgeteilt werden:

- **Schritt 1 – Zuweisung von Editoren:** das Editorial Office muss die Möglichkeit haben den Editoren bestimmte Artikel zuweisen zu können. Diese Zuweisungen müssen in der Datenbank gehalten werden und editierbar sein. Nachdem Zuweisungen gemacht wurden, müssen alle beteiligten Editoren vom Dienst sofort benachrichtigt werden.
- **Schritt 2 – Vorschlag von Reviewern:** der Editor muss die Möglichkeit haben für die Artikel Reviewer vorzuschlagen. Diese Reviewerlisten müssen auch in der Datenbank gehalten werden und editierbar sein. Im Fall, dass der vorgeschlagene Reviewer kein Dienstanutzer ist, wird er im Dienst mit dem „incomplete profile“ Status registriert, der später modifiziert werden kann.
- **Schritt 3 – Delegation von Artikeln:** das Editorial Office muss die Möglichkeit haben an Hand von den Editoren abgegebener und eigener Reviewerlisten die Reviewer für einen Artikel auszuwählen und diesen zu delegieren. Im Fall, dass die ausgewählten Reviewer den „incomplete profile“ Status haben, werden die Zugangsdaten vom Dienst automatisch generiert. Die entsprechenden Reviewer erhalten ihren Account per Email zugesandt und werden vom Dienst sofort über die neuen Tasks benachrichtigt.

Alle Emailbenachrichtigungen müssen vom Dienst automatisch generiert werden.

Um die Aufgabe zu lösen, wurden das Hauptmakro `ReviewerAllocation`, die Submakros `RAFA2E`, `RAFE2D`, `RAF2Delegation` und Hilfsmakros `RAFE2RUserSelect`, `ArticleDelegation` und `RAFRegisterUser` erstellt. Bei der Workflowentwicklung wurden sowohl bestehende als auch neuerstellte SIBs benutzt.

- **ReviewerAllocation**

- `LoopOverArray`

- RAFPreparePermissions
- **RAFA2E**
  - RAFGenA2EData
  - RAFPostsaveA2EData
  - RAFPrepareA2EMail
  - RAFPreSaveA2EData
  - RAFSaveA2EData
  - RAFSelectAction
- **RAFE2D**
  - RAFE2DPostSave
  - RAFE2DSaveData
  - RAFGenE2DData
  - RAFE2DSaveCreateUser
  - CheckRequestParameterByPrefix
  - **RAFE2RUserSelect**
    - RAFE2DLookupCreateUser
    - RAFE2DRemoveSelectedUser
    - EnsureArticlesLoaded
- **RAF2Delegation**
  - RAFGen2DelegationData
  - RAFSave2Delegation
  - RAFPrepareDelegationData
  - RAFPrepareDelegationAckn
  - **ArticleDelegation**
    - DelegateArticleRequest2CallContext
  - **RAFRegisterUser**
    - RAFPrepareRegisterMail

Im Laufe der Implementierung wurden neue Datenbanktabellen, Administratoren, Geschäftsobjekte und verschiedene Emailtemplates erstellt. Die `rolepermissions.inc` und `navbar.properties` Dateien wurden erweitert.

## 6.2.2 Modellierungsphase

In diesem Kapitel werden die Aspekte vom UML-Design ausschließlich vom Zielzustand mittels Use Case-, Aktivitäts- und Klassendiagramm beschrieben, weil das Feature „Reviewer Allocation“ neuimplementiert wurde.

Das Diagramm in Abbildung 6.10 spiegelt den Zielworkflow des Features wieder und besteht aus vier Use Cases:

- **Reviewer Allocation** – ist sowohl dem Editorial Office als auch den Editoren zugänglich. Es beinhaltet drei folgenden Subfeatures, die als Links im Dienst dargestellt sind:
  - **Step 1 – assign editors:** ermöglicht dem Editorial Office den Editoren bestimmte Artikel zu zuweisen;
  - **Step 2 – propose reviewers:** ermöglicht den Editoren Reviewer für die Artikel vorzuschlagen;
  - **Step 3 – delegate articles:** ermöglicht de Editorial Office die Reviewer für einen Artikel auszuwählen und diesen zu delegieren.



Abbildung 6.10: Use Case „Reviewer Allocation Feature“ – Zielzustand

Das Aktivitätsdiagramm in der Abbildung 6.11 beschreibt die unterschiedlichen Schritte, die bei der Zuweisung von Editoren notwendig sind, die Abbildung 6.12 beschreibt die unterschiedlichen Schritte, die beim Vorschlag von Reviewern notwendig sind und die Abbildung 6.13 beschreibt die unterschiedlichen Schritte, die bei der Delegierung von Artikeln notwendig sind.

Um die neue Funktionalität zu realisieren, wurden neue Geschäftsobjekte, Administratoren und Datendanktabellen erstellt (siehe Abbildung 6.14).



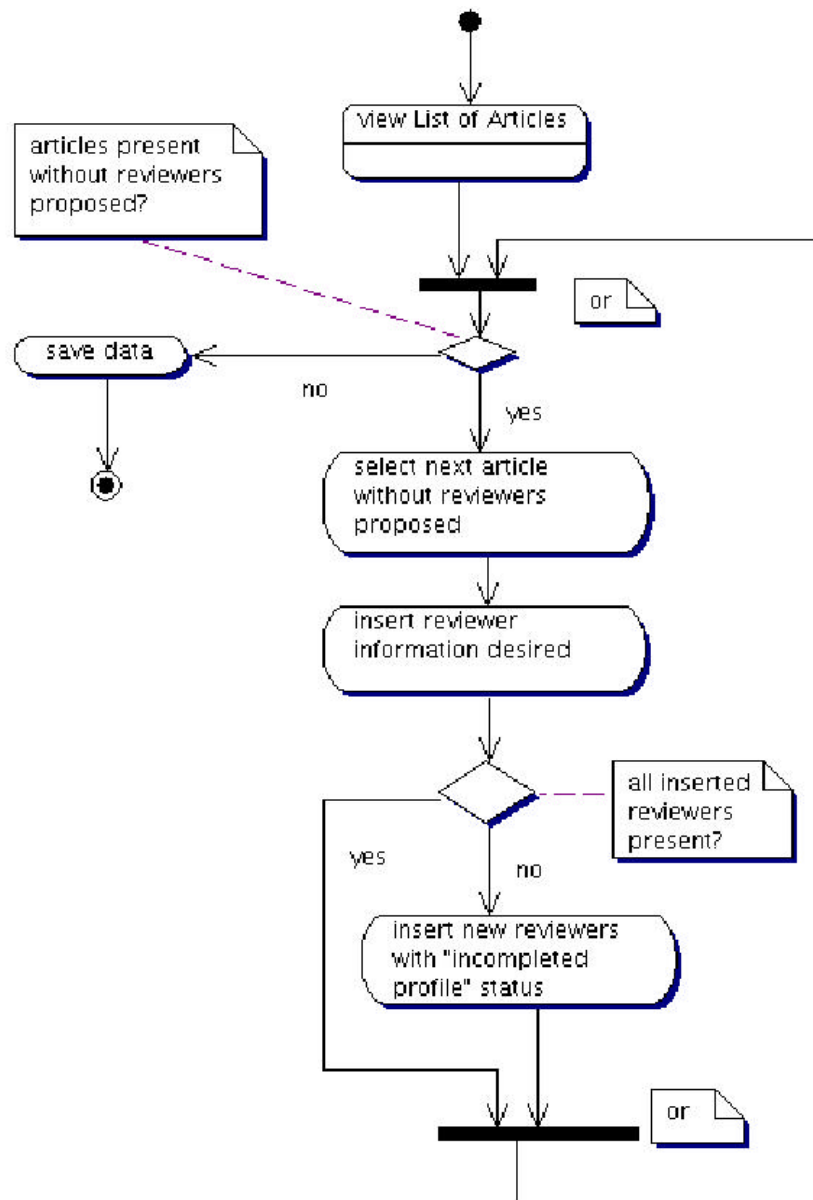


Abbildung 6.12: Activity Chart „Propose Reviewers“ – Zielzustand

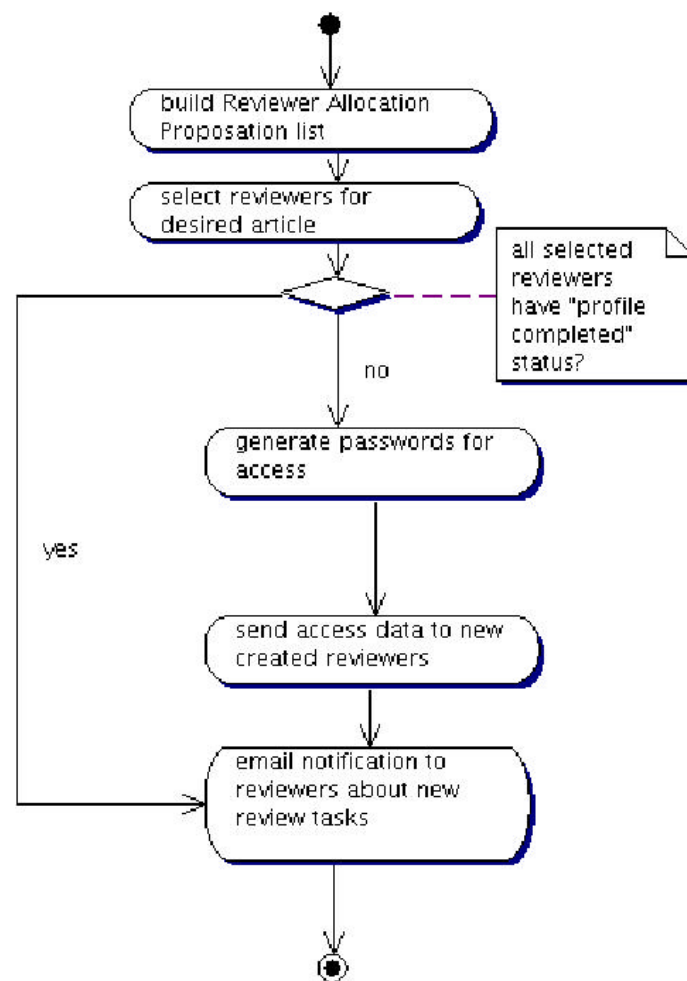


Abbildung 6.13: Activity Chart „Delegate Article“ – Zielzustand

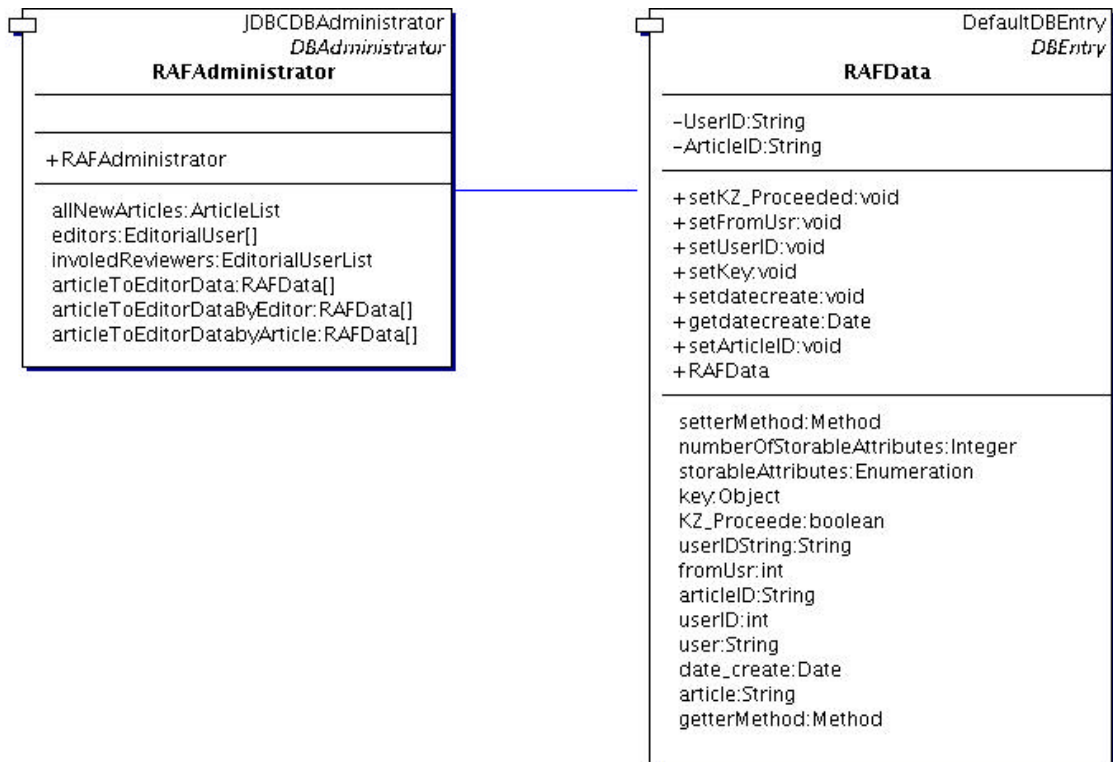


Abbildung 6.14: Klassendiagramm

### 6.2.3 Implementierungsphase

Bei der Implementierung wurden die drei logischen Schritte durch drei neue Makros implementiert. Die drei neue Makros entsprechen: für Schritt 1 – Makro RAFA2E 6.16, für Schritt 2 – Makro RAFE2D 6.17 und für Schritt 3 – Makro RAF2Delegation 6.18. Das Hauptmacro, das unser Feature darstellt, heisst ReviewerAllocation 6.15 und wurde direkt in „Journal\_merged.slg“ eingefügt.

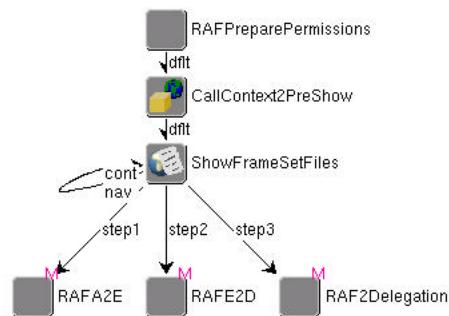


Abbildung 6.15: Makro ReviewerAllocation

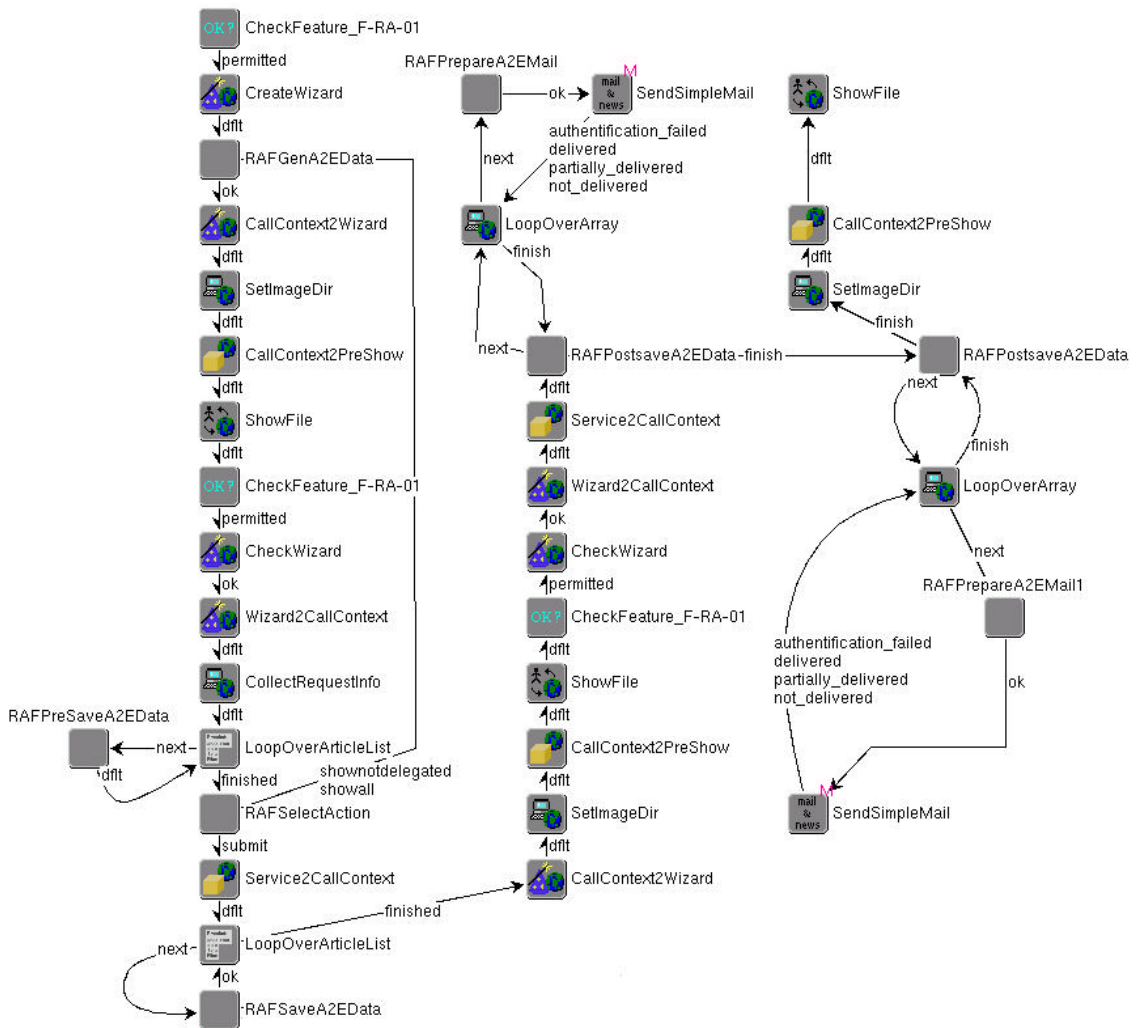


Abbildung 6.16: Makro RAFA2E



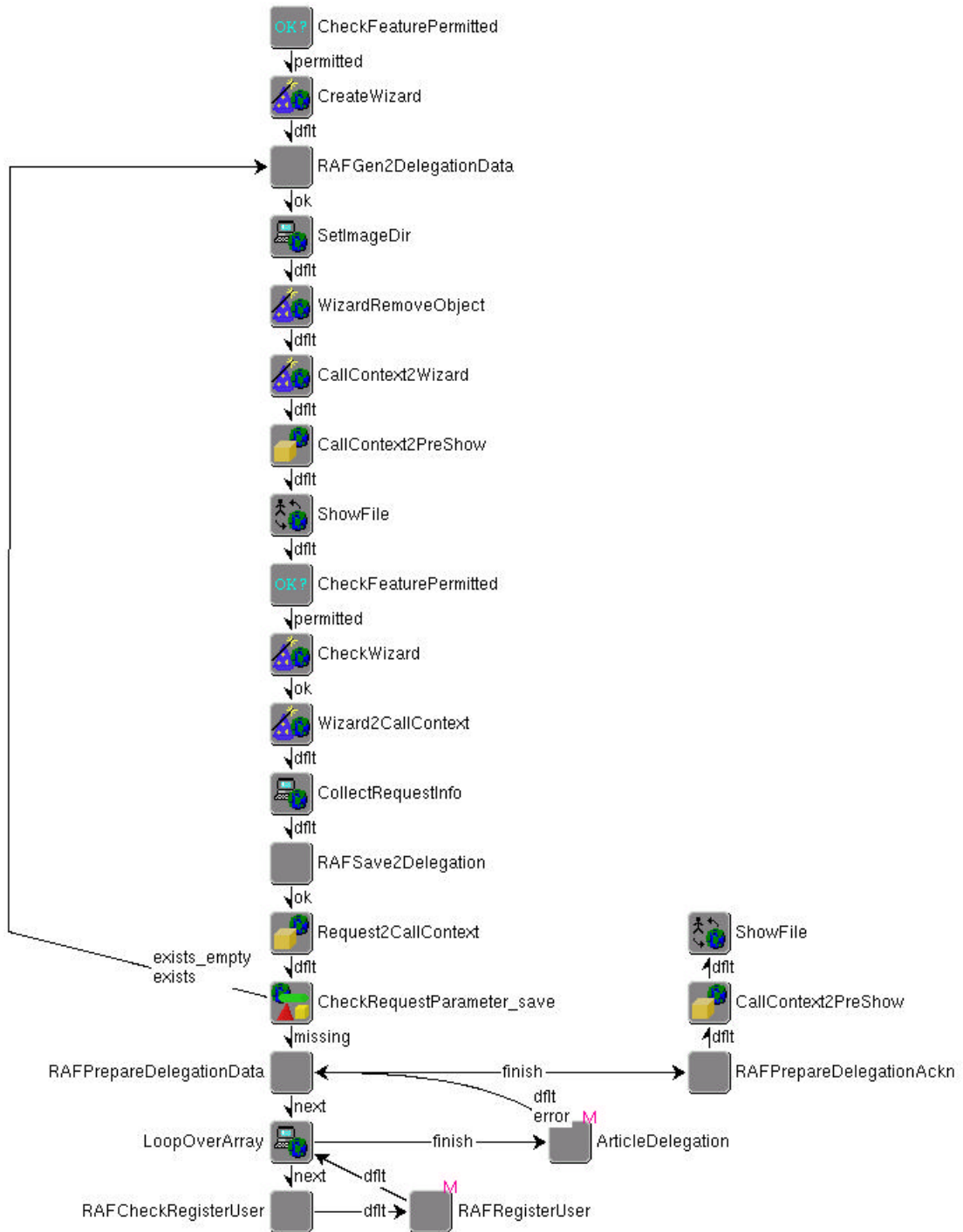


Abbildung 6.18: Makro RAF2Delegation

- Der Schritt 1 enthält folgende neue SIBs: RAFGenA2EData, RAFPostsaveA2EData, RAFPrepareA2EMail, RAFSaveA2EData, RAFSelectAction und RAFPreSaveA2EData. Der SIB RAFGenA2EData stellt bestimmte Daten bereit, die für diesen Schritt benötigt werden: Artikel ID, Vor-, Nachname und Email vom Editor. Der SIB RAFPreSaveA2EData merkt sich alle Änderungen, die das Editorial Office während diesen Schritte gemacht hat. Der SIB RAFSelectAction definiert, ob alle oder nur die delegierten Artikel angezeigt werden. Der SIB RAFSaveA2EData erzeugt zwei getrennte Listen. In einer Liste wird für jeden Editor gespeichert, welche Artikel zugeteilt werden und in der anderen Liste, welche Artikel gelöscht werden. Für jede dieser Listen gibt es entsprechende Schleifen, die hauptsächlich aus den SIBs RAFPostsaveA2EData, LoopOverArray, RAFPrepareA2EMail und einem Makro SendSimpleMail bestehen. Der SIB RAFPrepareA2EMail bereitet die nötigen Informationen vor, um die Email zu verfassen um dann mittels Makro SendSimpleMail an den Reviewer zu senden, ob ihm neue Artikel zugeteilt oder weggenommen wurden. Der SIB RAFPostsaveA2EData speichert Informationen aus diesen Listen in der Tabelle der Datenbank namens 'articletoeditor'.
- Der Schritt 2 enthält folgende neue SIBs: RAFGenE2DData, RAFE2DSaveData, LoopOverArray, RAFE2DPostSave, RAFE2DSaveCreateUser, CheckRequestParameterByPrefix und ein neues Makro RAF2RUserSelect. Der SIB RAFGenE2DData stellt bestimmte Daten bereit, die für diesen Schritt benötigt werden. Dies sind: Artikel ID, Beschreibung von Artikeln, Vor-, Nachname vom Editor und Email vom Reviewer. Das Makro RAF2RUserSelect erlaubt für die ausgewählten Artikel neue Reviewer mit Hilfe von RAFE2DLookupCreateUser hinzuzufügen und bereits vorhandene Reviewer mit Hilfe des SIBs RAFE2DRemoveSelectedUsers zu löschen. Der SIB RAFE2DSaveData bekommt aus dem Call Context eine Hashtable und bearbeitet ihre externe Liste, d.h. für jeden Artikel bekommt der SIB eine Liste von Reviewern. Dann wird diese Liste an den SIB LoopOverArray übergeben. Dieser SIB extrahiert jedes Element, also jeden Reviewer, und gibt es an den SIB RAFE2DSaveCreateUser weiter. Der SIB RAFE2DSaveCreateUser prüft, ob der ankommende Reviewer vorhanden ist. Falls ja, dann werden einfach für diesen Reviewer die zugeteilten Artikel in der Datenbanktabelle 'articletoreviewer' gespeichert. Andersfalls wird ein neuer Reviewer mit „incomplete Profile“ erzeugt und dementsprechend in der Datenbank gespeichert.
- Der Schritt 3 enthält folgende neue SIBs: RAFSave2Delegation, RAFPrepareDelegationAckn, RAFCheckRegisterUser, RAFPrepareDelegationData, RAFGen2DelegationData und ein Makro RAFRegisterUser. Der SIB RAFGen2DelegationData stellt bestimmte Daten bereit, die für diesen Schritt benötigt werden: Artikel ID, Vor-, Nachname von Reviewern, Deadline AcceptReject und Deadline Review. Der SIB CheckRequestParameter\_save prüft, ob das Editorial Office ausgewählte Artikel delegieren oder nur speichern möchte. Im letzteren Fall geht man zurück zum SIB RAFGen2DelegationData. Sonst landet man auf dem SIB RAFPrepareDelegationData. Dieser SIB bekommt aus dem Call Context eine

Hashtable und bearbeitet ihre externen Listen, d.h. für jeden Artikel bekommt der SIB eine Liste von Reviewern. Dann wird diese Liste an den SIB LoopOverArray übergeben. Dieser SIB extrahiert jedes Element (jeden Reviewer) und gibt es den SIB RAFCheckRegisterUser weiter und das Makro RAFRegisterUser, mit deren Hilfe für jeden Reviewer überprüft wird, ob sein Profile „incomplete“ ist. Falls ja, dann wird dem Reviewer eine Email mit dem Kennwort gesendet. Dafür bereitet der SIB RAFRegisterUser die benötigten Informationen vor. Das Makro ArticleDelegation führt die Delegation durch. Der SIB RAFPrepareDelegationAckn generiert eine Bestätigung der Delegationsoperation.

### 6.2.4 Testphase

Vor Beginn der Testphase sollte man sich als Editorial Office einloggen und überprüfen, ob für den Editor folgende Berechtigungen eingeschaltet sind:

- F-Role-04 – assign a role to user
- F-User-04 – create new user
- F-User-08 – register new user (full profile)
- F-User-09 – register new user (empty profile)

Falls es nicht erlaubt ist, sollten die Permissions hinzugefügt werden, da es für Schritt 2 notwendig ist.

Nachdem man den Button Revieweralloc betätigt, wird die Hauptseite des Features angezeigt (siehe Abbildung 6.19).

## Reviewerallocation

**This feature allows:**

<b>To pass an article to an editor for reviewer suggestion, just click:</b>	<a href="#">assign editors</a>
<b>To do new delegations, just click:</b>	<a href="#">delegate article</a>

Abbildung 6.19: Reviewer Allocation – Editorial Office – Startseite

Nach dem Klicken auf den Link `assign editors` sieht man eine Tabelle Article-Editor Assignment Matrix. Als default werden noch nicht delegierte Artikel in der Tabelle dargestellt. Wenn die Checkbox mit dem Häkchen gekennzeichnet ist, bedeutet es, dass der zugehörige Editor zu diesem Artikel schon eine Liste von Reviewern

## Assignment of Editors

This matrix gives an overview of assignments between articles and editors

### Article-Editor Assignment Matrix

This matrix shows the assignments between articles and Editors.

An "assignment" between an Editor and a paper implies that the Editor have to propose a list of Reviewers for this article.

ArtID	FF	DD	aa
008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

all not delegated

### Legend of Editors Initials

Initials	First Name	Last Name	Email
FF	FF	FF	powoloz05@ls05.cs.uni-dortmund.de
DD	D	D	powoloz04@ls05.cs.uni-dortmund.de
aa	a	a	powoloz01@ls05.cs.uni-dortmund.de

Submit Close Window

Abbildung 6.20: Article-Editor Assignment Matrix – „not delegated“ Filter

ArtID	FF	DD	aa
001	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
002	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
003	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
004	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
005	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
006	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
007	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

all not delegated

Abbildung 6.21: Article-Editor Assignment Matrix – „all“ Filter

angegeben hat (Abbildung 6.20). Wenn man auf den Button `all` klickt, dann werden alle vorhandenen Artikel angezeigt (Abbildung 6.21).

Nach dem Klicken auf den Button `Submit` wird eine Bestätigungsseite, in der alle vorgenommenen Änderungen dargestellt sind, angezeigt (Abb. 6.22).

## Assignment of Editors

**Following entries are to be revoked**

ArtID	Editors
005	FF,
003	DD,
001	DD,

**Following entries are to be invoked**

ArtID	Editors
005	FF,
003	DD,
001	DD,

If you want to accept the changes, then click the "Submit" button.



Abbildung 6.22: Assignment of Editors – Bestätigungsseite

Nach dem Klicken auf den Button `Submit` wird eine Acknowledgementseite dargestellt, die den erfolgreichen Vorgang bestätigt (Abbildung 6.23).

## Acknowledgement of operations for Assignment of Editors

The operations "to define a new assignment" and/or "to release an existing assignment" were/was successfully fulfilled.

**Following entries are to be revoked**

ArtID	Editors
005	FF,
003	DD,
001	DD,

**Following entries are to be invoked**

ArtID	Editors
005	FF,
003	DD,
001	DD,



Abbildung 6.23: Assignment of Editors – Acknowledgementseite

Jetzt loggt man sich aus dem System aus und als Editor, dem die neu zugewiesenen Artikel mitgeteilt wurden, wieder ein (siehe Abbildung 6.24).

## Reviewerallocation

This feature allows:



Abbildung 6.24: Reviewer Allocation – Editor – Startseite

Nach dem Klicken auf den Button `propose reviewers` wird eine Liste von Artikeln dargestellt, für die eine Liste von möglichen Reviewern zusammengestellt wird (Abbildung 6.25).

## Suggestion of Reviewers

Hier you can propose reviewers for each new article

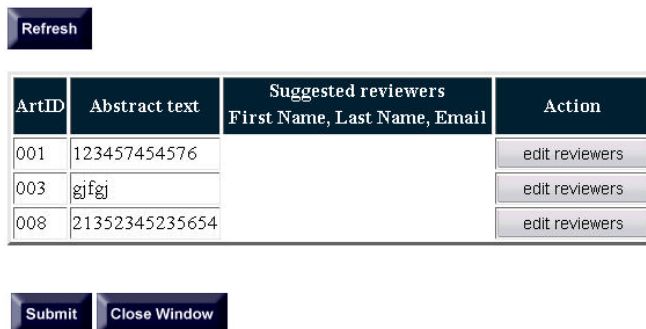


Abbildung 6.25: Reviewer Allocation – Vorschlag von Reviewers

Nach dem Klicken auf den Button `edit reviewers` wird das Formular in Abbildung 6.26 angezeigt. Mit Hilfe des Formulars können einerseits neue Reviewer hinzugefügt und andererseits existierende Reviewer gelöscht werden.

Hier füllt man das Formular aus und drückt den Button `Add user`. Weiterhin findet eine Überprüfungsphase der angegebenen Daten statt. Zuerst wird ein Vorhandensein der Email überprüft. Falls es fehlt (Abb. 6.27) oder fehlerhaft geschrieben ist (Abb. 6.28), wird eine Fehlermeldung ausgegeben.

## Suggestion of Reviewers

Please enter email adress, first name and last name for the user you want to insert

First name:

Last name:

Email:

Add user

### List of Reviewers

This matrix gives an overview of suggested reviewers.

To remove one or more reviewers from the list, just select the appropriate cell or cells within the matrix. Then click the "Remove users from list" button below the matrix.

	First Name	Last Name	Email
<input type="checkbox"/>	aaa	bbb	powoloz09@ls05.cs.uni-dortmund.de

Remove users from list

Submit

Close Window

Abbildung 6.26: Reviewer Allocation – das Vorschlagsformular

The following internal error(s) appeared:

Email for user not set

## Suggestion of Reviewers

Please enter email adress, first name and last name for the user you want to insert

First name:

Last name:

Email:

Add user

Abbildung 6.27: Suggestion of Reviewers – fehlende Email-Adresse

The following internal error(s) appeared:  
The entered value has not a valid e-mail adress format

---

## Suggestion of Reviewers

Please enter email adress, first name and last name for the user you want to insert

First name:

Last name:

Email:

Add user

Abbildung 6.28: Suggestion of Reviewers – falsches Email-Format

Falls die angegebenen Email richtig ist aber Vor- oder Nachname des Reviewers fällt und im System kein Benutzer mit solcher Email vorhanden ist, dann wird eine Fehlermeldung angezeigt (Abbildung 6.29).

The following internal error(s) appeared:  
Please enter Firstname for the new user,  
Please enter Lastname for the new user

---

## Suggestion of Reviewers

Please enter email adress, first name and last name for the user you want to insert

First name:

Last name:

Email:

Add user

Abbildung 6.29: Suggestion of Reviewers – das Fehlen von Vor- und Nachname

Falls alle Angaben korrekt waren, wird eine Bestätigungsseite angezeigt (Abb. 6.30).

Hier wird den Button `Submit` angeklickt. Damit bestätigt man die vorgeschlagenen Reviewer. Jetzt loggt man sich als Editorial Office ein. Dann drückt man den Button `Revieweralloc` und in der erschienenen Seite des Features drückt man den Link `'delegate article'`. Es wird eine Matrix der Delegation dargestellt. Durch die Editoren vorgeschlagene Reviewer werden mit Häkchen gekennzeichnet. Wenn der gegebene Artikel schon an den gegebenen Reviewer verteilt ist, wird die entsprechende Checkbox geblockt, so dass keine Verteilung mehr vorgenommen werden kann (Abbildung 6.31).

Nach dem Klicken auf den Button `Delegate` beginnt die Überprüfungsphase der Daten. Falls ein Fehler auftritt, wird eine Fehlerliste angezeigt (Abb. 6.32).

The following reviewers are suggested:

ArtID	Suggested reviewers	
	First Name, Last Name, Email	
001	aaa, bbb,	powoloz09@ls05.es.uni-dortmund.de
	123, 567,	powoloz08@ls05.es.uni-dortmund.de
	Martin, Schmidt,	powoloz99@ls05.es.uni-dortmund.de

If you want to accept the suggestion, then click the "Submit" button.

**Submit**

Abbildung 6.30: Suggestion of Reviewers – Bestätigungsseite

## List of Delegated Reviews

This matrix gives an overview of all delegated reviews.

To define a new review, just select the appropriate cell within the matrix. Then click

- the "Delegate" button, if you want to save and to delegate this review,
- the "Save" button, if you want only to save this review.

To release an existing review, just deselect the cell.

ArtID	15	16	37	CC	DD	bb	ab
001	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
002	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Delegate** **Save**

Deadline Accept/Reject (dd.mm.yyyy)	Deadline Review (dd.mm.yyyy)
05.07.2005	01.01.2006

**Close Window**

Abbildung 6.31: Delegate article – Liste der delegierten Reviews

**Close Window**

## Delegation Matrix Acknowledgement

**Following errors occurred during delegation:**

ArtID	Error
002	desired_delegations_reached
001	desired_delegations_reached

Abbildung 6.32: Delegate article – Liste der aufgetretenen Fehler

Wiederum kann eine Verteilung von Artikeln vorgenommen werden (Abb. 6.33).

## List of Delegated Reviews

This matrix gives an overview of all delegated reviews.

To define a new review, just select the appropriate cell within the matrix. Then click

- the "Delegate" button, if you want to save and to delegate this review,
- the "Save" button, if you want only to save this review.

To release an existing review, just deselect the cell.

ArtID	15	16	37	CC	DD	bb	ab
001	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
002	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
009	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Deadline Accept/Reject (dd.mm.yyyy)	Deadline Review (dd.mm.yyyy)
05.07.2005	01.01.2006

Abbildung 6.33: Delegate article – neue Delegation

Falls keine Fehler eingetreten sind, wird eine Acknowledgementseite dargestellt (siehe Abbildung 6.34).

## Delegation Matrix Acknowledgement

Article ID	Delegated		Status(Since)	Decision Deadline	Task Deadline
	from	to			
009	a a	C C	waiting for decision(2005-07-05)	2005-07-05	2006-01-01
009	a a	123 567	waiting for decision(2005-07-05)	2005-07-05	2006-01-01
009	a a	aaa bbb	waiting for decision(2005-07-05)	2005-07-05	2006-01-01

Abbildung 6.34: Delegation article – Acknowledgementseite

Hier wird überprüft, ob die neuen Delegationen in der Delegation Matrix eingetragen sind (Abbildung 6.35).

Die Testphase von Reviewer Allocation Feature ist erfolgreich durchgeführt worden.

009	aa	C C	waiting for decision(since 2005-07-05)	2005-07-05	2006-01-01	<ul style="list-style-type: none"> <li>extend decision • extend task</li> <li>remove delegation</li> </ul>	please accept following article for reviewing	Regular Paper	009
009	aa	123 567	waiting for decision(since 2005-07-05)	2005-07-05	2006-01-01	<ul style="list-style-type: none"> <li>extend decision • extend task</li> <li>remove delegation</li> </ul>	please accept following article for reviewing	Regular Paper	009
009	aa	aaa bbb	waiting for decision(since 2005-07-05)	2005-07-05	2006-01-01	<ul style="list-style-type: none"> <li>extend decision • extend task</li> <li>remove delegation</li> </ul>	please accept following article for reviewing	Regular Paper	009

Abbildung 6.35: Delegation Feature – Ausschnitt aus der Delegation Matrix

## 6.3 Distribution Feature

### 6.3.1 Beschreibung der Aufgabe

Die Aufgabe bestand darin, den Workflow des Features **Bidding Matrix** dahingehend zu erweitern, dass der PC Chair die zusätzliche Möglichkeit bekommt, die Delegation der Artikel an die Reviewer durch ein externes Programm optimal zu lösen.

Hierzu soll eine lineare Lösung der Verteilung mit Hilfe des Programms: **lp\_solve** erfolgen. Die Einbindung und die Verwendung des externen Programms soll weitestgehend modular realisiert werden, so dass es leicht an verschiedenen Stellen im Service einzupflegen ist.

Für die Eingabedaten des Solvers wird eine Input Datei generiert, welche eine Zielfunktion und mehrere Restriktionen enthält. Das Ergebnis des Solvers wird wiederum in eine output Datei geschrieben.

### 6.3.2 Modellierungsphase

Mittels Anwendungs- und Aktivitätsdiagrammen wird versucht, die Erweiterungen des Workflows im Feature **Bidding Matrix** zu verdeutlichen.

Das Use Case Diagramm in der Abbildung 6.36 zeigt die Erweiterung des Workflows des Features **Bidding Matrix** um die neue Funktion: `Distribute Delegations with Solver`. Kommt der Solver zu einem Ergebnis, kann diese Verteilung mittels der Funktion `Apply Solver Output in` die **Bidding Matrix** übernommen werden. Die Funktionen des alten Workflows sind wie bisher anwendbar.

Das Aktivitätsdiagramm in der Abbildung 6.37 beschreibt den Ablauf der Verwendung des externen Solvers. Im Modul **Bidding Matrix** sind verschiedene Funktionen zur Bearbeitung der Matrixeinträge enthalten. Neu hinzugekommen ist die Funktion `Use Solver`. Abhängig von den Einträgen der Matrix wird eine Datei erzeugt, die als Input des externen Programms benötigt wird. Diese Datei enthält einerseits die Zielfunktion, die minimiert werden soll, andererseits die Restriktionen, die die Grenzwerte einzelner Variablen festlegen.

Nachdem der Solver zu einem Ergebnis gekommen ist, kann sich der Benutzer die (optimale) Lösung in der Matrix anzeigen lassen. Die neu berechneten Delegationen erscheinen in der selben Form wie die von Hand selektierten Checkboxen in der Matrix. Nun ist der Benutzer in der Lage, weitere Änderungen an der Verteilung der Delegationen vorzunehmen. Daraufhin kann die Verteilung verworfen, gespeichert oder gespeichert und delegiert werden.

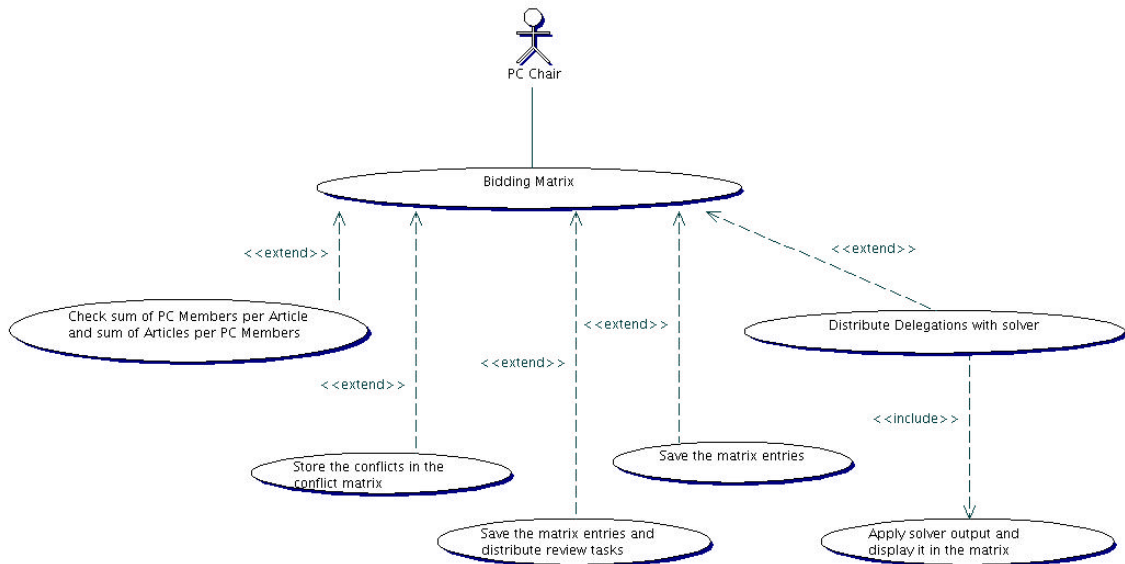


Abbildung 6.36: Use Case

### 6.3.3 Implementierungsphase

In diesem Abschnitt wird die Implementierung des Workflows im Feature **Bidding Matrix** dargestellt. Nach einer kurzen Übersicht über die neu implementierten SIBs wird auf deren Spezifikationen eingegangen. Die SIBs lassen sich in zwei Kategorien gliedern: spezielle und universelle.

#### Übersicht

Die folgende Liste zeigt die speziellen und universellen SIBs (siehe Abbildung 6.38).

- UseSolver (spezial)
- GenerateSolverInput (universell)
- ExecuteSolver (universell)
- ParseSolverOutput (universell)
- SolverOutput2BiddingMatrix (spezial)
- TerminateSolver (universell)

Zusätzlich wurden folgende Einträge in die Datei `web.xml` eingetragen.

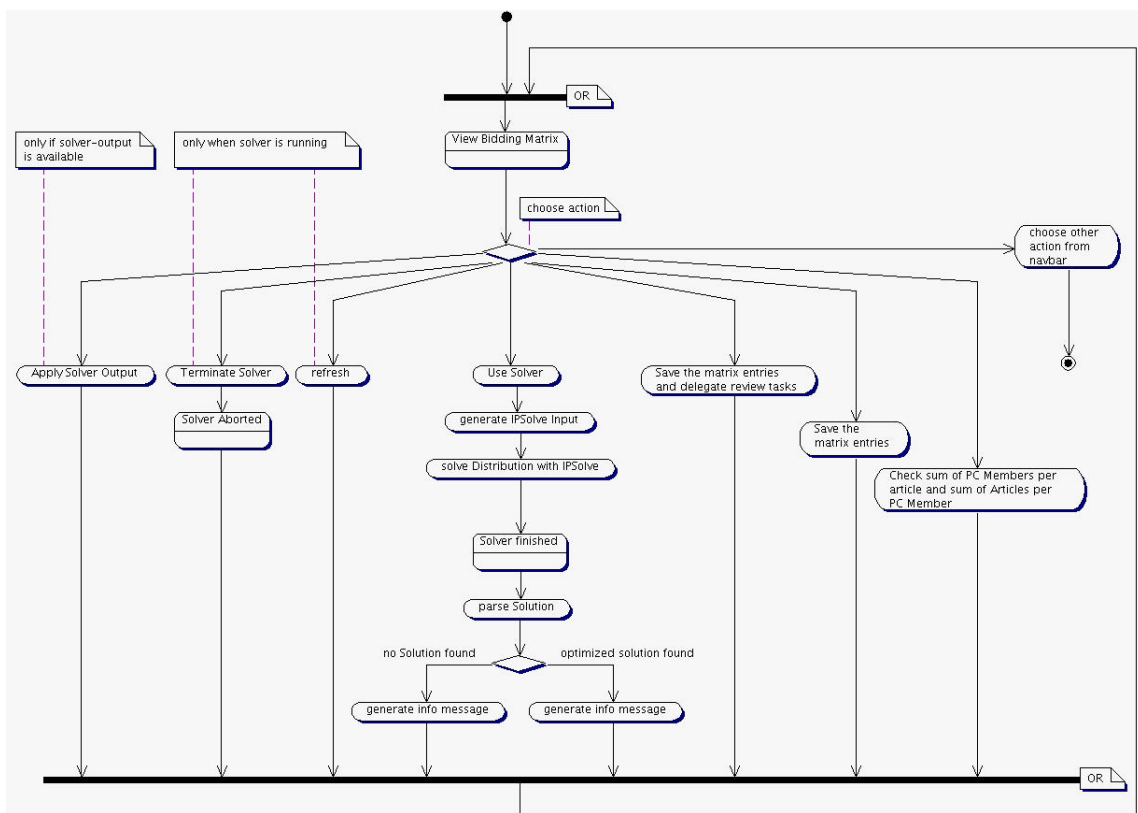


Abbildung 6.37: Activity Chart

- InputFilePath
- OutputFilePath
- SolverPath
- SolverExecutable
- PidofPath

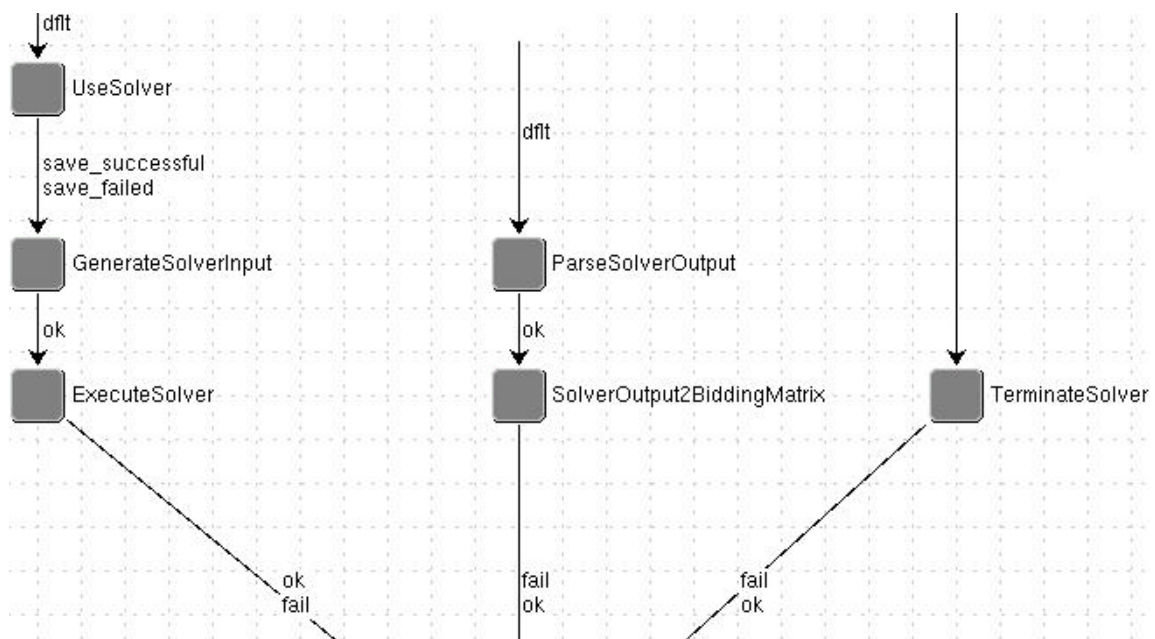


Abbildung 6.38: Übersicht über die neu implementierten SIBs im Makro Bidding Matrix

### Ablauf des Solver-Prozesses

Der PC Chair kann die gewünschte Anzahl der Reviews für jeden Artikel mittels eines drop-down Menüs festlegen. Danach kann er die maximale Abweichung der durchschnittlichen Last definieren. Beim Benutzen von **Use Solver** wird die Lösung in mehreren Schritten generiert.

Zuerst wird die aktuelle **Delegation Matrix** analysiert und die durchschnittliche Last der Reviewer berechnet. Außerdem wird die maximal mögliche Anzahl neu zu verteiler Delegationen für jeden PC Member berechnet. Konflikte und bereits abgeschlossene Delegationen werden dabei berücksichtigt. Anhand dieser Informationen und den vom PC Chair selektierten Einträgen in der Bidding Matrix werden die Zielfunktion und die Restriktionen erstellt. Die Zielfunktion spiegelt die Konflikte und die aktuelle Last der Reviewer wieder. Diese gilt es dann zu minimieren:

$$MIN = \sum_{i,j}^{Articles,Users} conflict_{i,j} * bid_{i,j} * workload_j * X_{i,j}$$

Der zweite Teil des Solver Inputs besteht aus einer Menge von Restriktionen. Diese Restriktionen setzen sich aus drei verschiedenen Arten zusammen. Zum Einen wird festgelegt, dass eine Variable  $X_{i,j}$  nur die Werte 0 oder 1 annehmen kann. Dabei steht die 0 dafür, dass keine Delegation zwischen Artikel  $i$  und Reviewer  $j$  generiert wird, die 1 für eine Delegation.

Ein weiterer Restriktionstyp sorgt dafür, dass ein Artikel genau an die gewünschte Anzahl von Reviewern delegiert wird. Hierzu wird für jeden Artikel folgende Ungleichung benutzt:

$$\sum_j^{Users} X_{article_j} < AnzahlderReviewsproArtikel.$$

Weiter werden für jeden User zwei Restriktionen erzeugt, welche jeweils eine untere und obere Grenze für die jeweilige Last setzen.

$$\sum_i^{Artikel} X_{user_i} > untereGrenzederLast.$$

$$\sum_i^{Artikel} X_{user_i} < obereGrenzederLast.$$

Auftretende Sonderfälle, wie z.B. User, deren Anzahl der Konflikte größer ist als die untere Grenze der Last, werden besonders behandelt.

Im nächsten Schritt wird eine Input-Datei, die die oben beschriebene Zielfunktion und Restriktionen als ASCII-Text enthält, erzeugt.

### Beschreibung und Anwendung der SIBs

Die neu implementierten SIBs unterteilen sich in zwei Kategorien. Während die SIBs der ersten Kategorie universell einsetzbar sind, lassen sich die SIBs der zweiten Kategorie nur im Zusammenhang mit dem Makro Bidding Matrix verwenden.

- Der SIB `UseSolver` generiert die Zielfunktion und die Restriktionen aus den in der Bidding Matrix gesetzten Daten. Die Zielfunktion wird als String gespeichert und in den Call Context unter dem Schlüssel 'SolverFunction' geschrieben. Die Restriktionen werden einzeln als String in einem Vektor gehalten und ebenfalls in den Call Context unter dem Schlüssel 'SolverFunction' geschrieben.
- `GenerateSolverInput` ist ein universeller SIB und wird dazu benutzt den String und den Vektor in eine Input-Datei zu schreiben. In der `web.xml` wird der Pfad, in dem diese Datei gespeichert wird, festgelegt. Der Name der Datei, an den die `ServiceID` automatisch angehängt wird, wird als SIB-Parameter eingegeben.
- `ExecuteSolver` ist ein universeller SIB, der die generierte Input-Datei benutzt, um mit dieser den Solver als Thread zu starten. Der Pfad der ausführbaren Datei des Solvers wird wiederum in der `web.xml` festgelegt. Die `ServiceID` wird automatisch an den Input- bzw. Output-Dateinamen angehängt.

Nachdem der Thread gestartet wurde, wird die ProzessID im ServiceContext gespeichert. Diese ProzessID (pid) wird als SIB-Parameter gesetzt um Überlappungen vorzubeugen. Außerdem wird die pid benötigt um den Solver bei Bedarf zu terminieren.

Zusätzlich wird der Solver-Status im Service Context gespeichert, so dass der Benutzer auch nach erneutem Einloggen im Dienst den Status abrufen kann. Nach der erfolgreichen Berechnung der Lösung wird diese in eine Output-Datei geschrieben. Name und Pfad der Datei können wiederum individuell festgelegt werden, wobei die ServiceId an den Namen angehängt wird.

- Der universelle `SIB ParseSolverOutput` liest die oben beschriebene Output-Datei und parst deren Inhalt. Die Zeilen werden als Strings in einen Vektor zur weiteren Verwendung gespeichert. Der SIB benötigt die Eingabe des Pfads und des Namen der Output-Datei sowie die Eingabe des Vektor-Namens. Der Name des Vektors wird im Graphen festgelegt.
- `SolverOutput2BiddingMatrix` parst den Output-Vektor und filtert die Variable-Wert-Paare mittels regulärer Ausdrücke heraus. Die Benutzer-Artikel-Kombinationen werden auf Gültigkeit überprüft und im Falle eines positiven Ergebnisses auf die Bidding Matrix abgebildet. Die neue Verteilung wird mit der bereits vorhandenen Verteilung verrechnet, so dass die bereits selektierten Checkboxen in der Bidding Matrix erhalten bleiben.
- `TerminateSolver` wird benutzt um den laufenden Solver-Prozess bei Bedarf zu beenden. Hierzu wird die im Call Context gespeicherte ProzessID verwendet. Dieser SIB benutzt das `/bin/kill`-Kommando und ist dementsprechend nur auf Unix-Systemen einsetzbar.

### 6.3.4 Testphase

Die Testphase wurde manuell durchgeführt, indem im neuen Solver-Workflow alle erdenklichen Situationen durchgespielt wurden.

Das Hauptaugenmerk beim Testen lag in der Verteilung der Reviews an die PC Member. Die Verteilung sollte so ausgeglichen wie möglich sein. Spezialfälle, wie extrem hohe Lasten einzelner PC Member sollten durch die Verteilung angepasst werden, so dass diese nur sehr wenige neue Reviews zugewiesen bekommen, wohingegen PC Member mit geringer Last mehr neue Delegationen bekommen. In der Testphase wurde festgestellt, dass dieses gewünschte Verhalten erreicht wird.

Für weitere Spezialfälle, wie PC Member, die sehr viele Konflikte haben, werden spezielle Restriktionen geschrieben, so dass auch diese korrekt in die Verteilung eingerechnet werden. Auch der Fall, dass die maximale Anzahl der Reviews pro Artikel, die im Setup festgelegt ist, vom Benutzer überschrieben wird, bestand die Testphase.

Für einige sehr spezielle Fälle ist es allerdings nicht möglich, eine ausgeglichene

Verteilung zu erhalten. Ist dies der Fall, hat der Benutzer die Möglichkeit die prozentuale Abweichung zu erhöhen. Dadurch wird erreicht, dass in jedem Fall eine Verteilung berechnet wird, was allerdings Auswirkungen auf die Qualität der Lösung zur Folge hat. Abbildung 6.39 zeigt den Zustand der Matrix ohne die Lösung des Solvers. In der letzten Zeile ist die Last der einzelnen Benutzer abzulesen. Die PC Member **BS** und **MKa** haben hier eine Last von '0'.

ID	BS	IS	JB	JS	LR	MG	MJ	MK	MKa	PW	SB	SS	TF	TS	YL	ID	Sum:()
044	⌋	I ⌋	I ⌋	I	I ⌋	I ⌋	I ⌋	I ⌋	⌋	I ⌋	I ⌋	I ⌋	B	c I	I	044	3
045	⌋	I ⌋	I	I ⌋	I ⌋	I ⌋	I	I ⌋	⌋	I	I ⌋	I ⌋	I ⌋	c I	I ⌋	045	3
046	⌋	I	I ⌋	I ⌋	I ⌋	I	I	I ⌋	⌋	I ⌋	I ⌋	I ⌋	I ⌋	c C	I ⌋	046	3
047	⌋	I ⌋	B	I ⌋	B ⌋	I	I	U	⌋	I ⌋	I	I ⌋	I ⌋	c C	I ⌋	047	3
048	⌋	I ⌋	I ⌋	B	B	I	I	U	⌋	U	I	I ⌋	I ⌋	c C	I ⌋	048	3
049	⌋	I ⌋	B	U	B	I	I	U	⌋	U	I	I ⌋	I ⌋	I	I ⌋	049	3
050	⌋	B	I ⌋	I ⌋	I ⌋	B	B	B	⌋	U	B	I ⌋	I ⌋	c I	R	050	2
051	⌋	I ⌋	I ⌋	I	I ⌋	B	I	B	⌋	U	I ⌋	I ⌋	B	c I	I ⌋	051	3
052	⌋	c C	B	I ⌋	I ⌋	I ⌋	I ⌋	I ⌋	⌋	U	B	I ⌋	c C	c U	B	052	3
053	⌋	⌋	I	⌋		⌋	⌋	⌋	⌋	⌋		⌋		c	⌋	053	4
054	⌋	⌋	B	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	c	⌋	054	
ID	BS	IS	JB	JS	LR	MG	MJ	MK	MKa	PW	SB	SS	TF	TS	YL	ID	Sum:()
055	⌋	⌋	I	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	c	⌋	055	
056	⌋	⌋	U	⌋	⌋		⌋	⌋	⌋	⌋		⌋	⌋	c	⌋	056	3
057	⌋	⌋	I	⌋	⌋	⌋	⌋	⌋	⌋	⌋		⌋		c	⌋	057	3
058	⌋	⌋	I	⌋	⌋	⌋	⌋	⌋	⌋	⌋		⌋	⌋	c	⌋	058	1
059	⌋	⌋	I	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	⌋	059	
<b>Sum:</b>		<b>14</b>	<b>15</b>	<b>12</b>	<b>13</b>	<b>13</b>	<b>12</b>	<b>12</b>		<b>12</b>	<b>15</b>	<b>12</b>	<b>14</b>	<b>12</b>	<b>11</b>		

Abbildung 6.39: Bidding matrix vor der Verwendung des Solvers

Abbildung 6.40 zeigt den Zustand, der die Ausgabe des Solvers enthält. Die vom Solver vorgeschlagenen Delegationen sind in diesem Fall die selektierten Checkboxes.

Der Solver wurde mit 4 Reviews pro Artikel und einer Abweichung von 10% angestoßen. Wie sich in der letzten Zeile ablesen läßt, ist eine gleichmäßige Verteilung dabei herausgekommen.

ID	BS	IS	JB	JS	LR	MG	MJ	MK	MKa	PW	SB	SS	TF	TS	YL	ID	Sum:()	
044	<input checked="" type="checkbox"/>	I	I	I	I	I	I	I	I	I	I	I	I	B	c I	I	044	4
045	I	I	I	I	I	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	I	c I	I	045	4
046	I	I	I	I	I	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	I	c C	I	046	4
047	I	I	B	I	B	I	I	U	<input checked="" type="checkbox"/>	I	I	I	I	I	c C	I	047	4
048	I	I	I	B	B	I	I	U	<input checked="" type="checkbox"/>	U	I	I	I	I	c C	I	048	4
049	I	I	B	U	B	I	I	U	I	U	I	I	I	I	I	I	049	4
050	<input checked="" type="checkbox"/>	B	I	I	I	B	I	B	<input checked="" type="checkbox"/>	U	B	I	I	I	c I	R	050	4
051	I	I	I	I	I	B	I	B	<input checked="" type="checkbox"/>	U	I	I	B	I	c I	I	051	4
052	I	c C	B	I	I	I	I	I	I	U	B	I	<input checked="" type="checkbox"/>	c C	c U	B	052	4
053	I	I	I	I	I	I	I	I	I	I	I	I	I	c	I	I	053	4
054	<input checked="" type="checkbox"/>	I	B	I	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	c	I	054	4
ID	BS	IS	JB	JS	LR	MG	MJ	MK	MKa	PW	SB	SS	TF	TS	YL	ID	Sum:()	
055	I	I	I	I	I	I	<input checked="" type="checkbox"/>	I	I	<input checked="" type="checkbox"/>	I	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	c	I	055	4	
056	I	I	U	I	I	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	c	I	056	4	
057	<input checked="" type="checkbox"/>	I	I	I	I	I	I	I	I	I	I	I	I	c	I	057	4	
058	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	<input checked="" type="checkbox"/>	I	I	I	I	c	<input checked="" type="checkbox"/>	058	4	
059	I	I	I	I	I	I	<input checked="" type="checkbox"/>	I	I	<input checked="" type="checkbox"/>	I	I	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I	059	4	
<b>Sum:</b>	<b>16</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>15</b>	<b>16</b>	<b>16</b>	<b>16</b>	<b>16</b>			

Abbildung 6.40: Bidding matrix nach der Verwendung des solvers

## 6.4 XML-Import/Export für Reports

### 6.4.1 Beschreibung der Aufgabe

Der bestehende Workflow des Report Features stellt die Einreichung eines Reports über ein HTML-Formular zur Verfügung. D.h. die Begutachter können ihren Report entweder online schreiben, oder den in einem Texteditor geschriebenen Report in das Formular einfügen und anschliessend einreichen. Der Vorgang des Einreichens soll dahingehend erweitert werden, dass die Begutachter XML-Templates zum Verfassen der Reports verwenden können, um diese später hochzuladen. Dabei ist das Hochladen von mehreren Reports in einer XML Datei möglich. Der Dienst wird die XML-Datei validieren und auftretende Fehler visualisieren. Des Weiteren soll ein Workflow für den Export von Reports in eine XML-Datei entwickelt werden.

### 6.4.2 Modellierungsphase

Im folgenden Kapitel geht es um die Modellierung des neuen Workflows, bezogen auf die oben genannte Aufgabe. Der neue Workflow wurde mit Hilfe von UML-Diagrammen modelliert. Im Vordergrund standen neue Aktionen, sowie Links, die ein Begutachter für den Import sowie den Export von Reports ausführen kann. Abbildungen 6.41 und 6.42 zeigen den Workflow mit den neuen Aktionen:

- download XML template
- import XML reports
- XML report
- modify XML report
- XMLExport
- CSVExport

Der Begutachter hat zuerst die Möglichkeit das XML Template durch Anklicken des "download XML Template" Links herunterzuladen und dann auszufüllen. Später kann der Begutachter dieses existierende XML-Template mit geschriebenen Reports hochladen und mittels des Dienstes validieren. Nach der Validation kann der Begutachter nochmals seinen Report überarbeiten und schließlich einreichen. Im Falle einer XML Datei mit mehreren Reports, werden diese nacheinander überarbeitet und eingereicht. Um die Reports als "XML " oder "CSV" –Datei zu exportieren, klickt der Benutzer auf den "XMLExport " oder "CSVExport" –Button. Es wird eine CSV- Datei oder XML Datei erzeugt, in der sich alle ausgewählten Reports befinden.

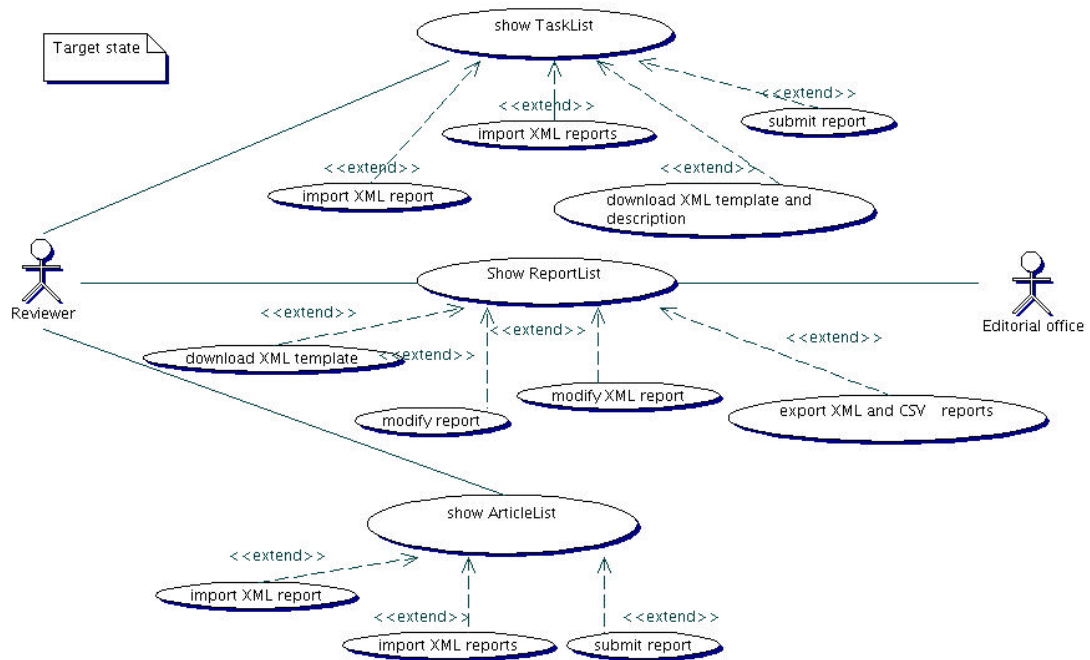


Abbildung 6.41: Use Case Import, Export XML Reports

Das Aktivitätsdiagramm in der Abbildung 6.43 beschreibt die unterschiedlichen Schritte, die bei dem Export von Reports notwendig sind.

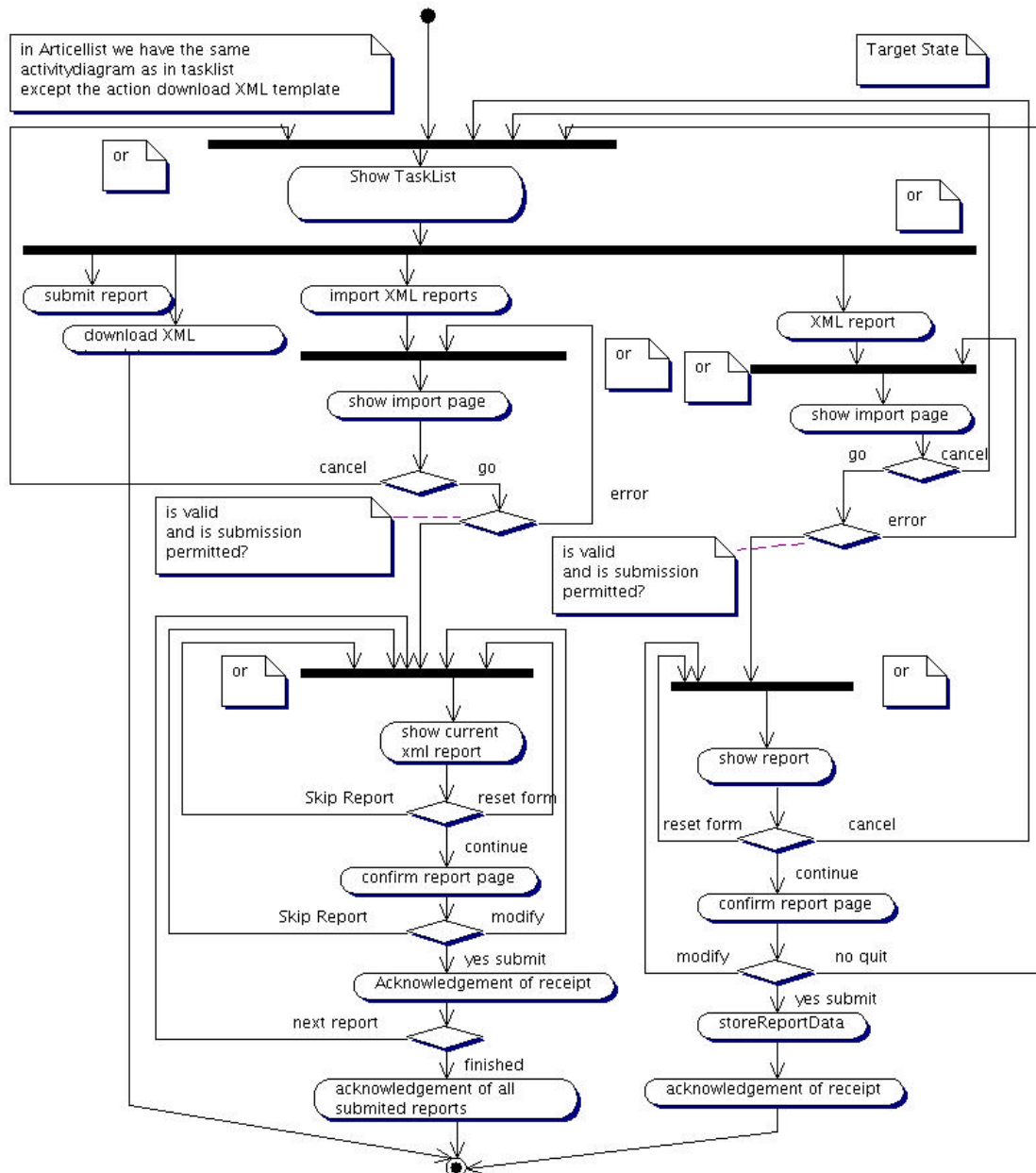


Abbildung 6.42: Activity Chart Import XML Reports

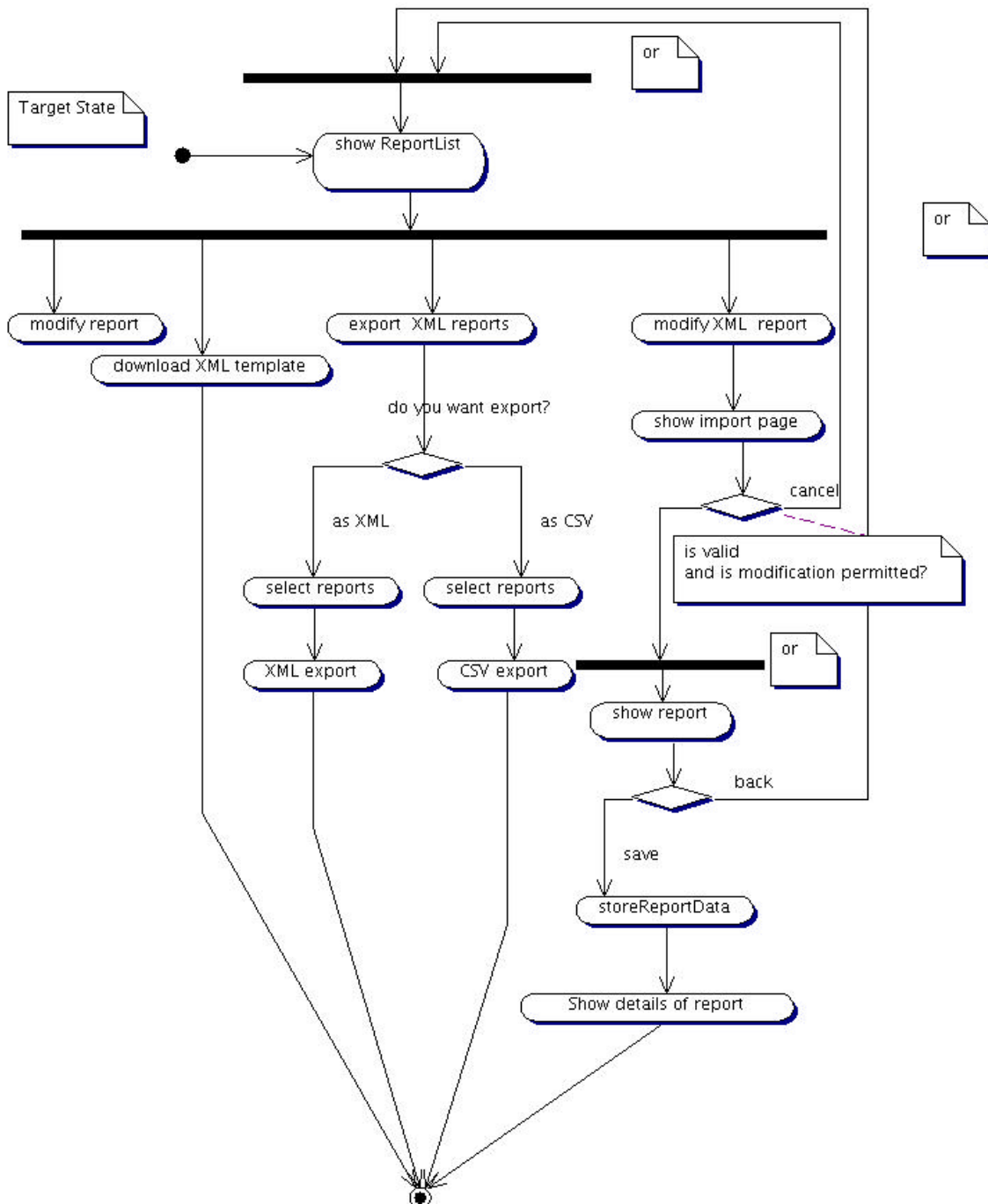


Abbildung 6.43: Activity Chart Modify XML Reports

### 6.4.3 Implementierungsphase

In diesem Abschnitt wird die Implementierung des neuen Workflows beschrieben. Drei existierende Makros wurden erweitert: **Articles**, **Reports** und **Tasklist**, in denen neue SIBs implementiert oder vorhandene SIBs modifiziert wurden. Die folgenden Dateien (SIBs und WML) wurden verändert bzw. neu erstellt.

- ArticleListActions (geändert).
- DownloadXMLReportTemplate (neu).
- ExportReportAction (neu).
- GenerateCSVReports (neu).
- GenerateExportReportList (neu).
- GenerateXMLReports (neu).
- LoopOverReports (neu).
- ModifyXMLImportReport (neu).
- PrepareTemplateKeys (neu).
- ReportListActions (geändert).
- ShowAllSubmittedXMLReport (neu).
- ShowArticleList (geändert).
- ShowReadReport (geändert).
- ShowReportList (geändert).
- ShowStartXMLReportsImport (neu).
- ShowTaskList (geändert).
- TaskListActions (geändert).
- XMLImportReport (neu).
- XMLImportReports (neu).
- allxmlreports-cont.wml (neu).
- articlelist-cont.wml (geändert).

- `exportreports-cont.wml` (neu).
- `reportlist-cont.wml` (geändert).
- `reportsubmitconfirm-cont.wml` (geändert).
- `rolepermissions.inc` (geändert).
- `startmodifyxmlreportimport-cont.wml` (neu).
- `startxmlreportimport-cont.wml` (neu).
- `startxmlreportsimport` (neu).
- `submitreport-cont.wml` (geändert).
- `tasklist-cont.wml` (geändert).

### Beschreibung der beteiligten SIBs

- **ArticleListActions:** Der SIB wurde modifiziert. Es wurde ein zusätzlicher Branch, "xml\_Report", eingefügt.
- **DownloadXMLReportTemplate:** Dieser SIB liest einen String "templatekey" aus dem SIB "FillTemplate".  
Der String "templatekey" wird unter dem Parameter "filledbody" in den Call Context geschrieben. Danach werden die Daten als XML Datei hochgeladen (siehe Abbildung 6.44).

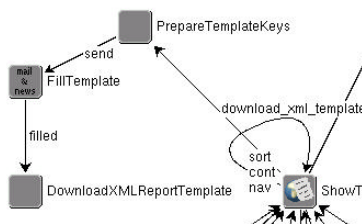


Abbildung 6.44: Download XML Template

- **ExportReportAction:** Dieser SIB ist für die Weiterleitung zu den Aktions-Teilgraphen zuständig, die in der "Export-Reports" Seite verfügbar sind (CSVExport, XMLExport).

- **GenerateCSVReports**: Dieser SIB wurde erstellt, um die gewählten Reports auszulesen und als CSV Datei zu exportieren.
- **GenerateExportReportList**: Abhängig von der Rolle des Benutzers speichert dieser SIB eine Liste von Reports in den Call Context. Diese Liste enthält alle Reports, die der Benutzer geschrieben hat. Sie werden mit Hilfe des SIBs `CallContext2PreShow` für die Anzeige aufbereitet und dann im SIB `ShowFrameSetFile` übernommen und angezeigt.
- **GenerateXMLReports**: Dieser SIB wurde erzeugt, um die selektierten Reports einzulesen und als XML Datei zu speichern (siehe Abbildung 6.45).
- **LoopOverReports**: Dieser SIB durchläuft das Array **ArrayReports** vom Typ `Element` und speichert das aktuell ausgewählte Objekt im Call Context unter einem Schlüssel ab.
- **ModifyXMLImportReport**: Dieser SIB benutzt JDOM und den Parser "Xerces". Er überprüft die Wohlgeformtheit und Validität der XML Datei. Wenn die XML Datei valide und wohlgeformt ist, wird ein angepasster Objektbaum erstellt und im Hauptspeicher gehalten. Die Inhalte der XML Datei werden gelesen und unter bestimmten Schlüsseln in den Call Context geschrieben.
- **PrepareTemplateKeys**: Liest das vorhandene XML Template ein. Das Templateverzeichnis wurde als Properties Schlüssel in dem Parameter "templateDir" angegeben. Der Name des zu ladenden Templates ist über den Parameter "templatename" spezifiziert worden. Das Templateverzeichnis und der Auswertungsbereich "score-interval" wurden in einer Hashtable gespeichert. Diese Hashtable wurde unter einem bestimmten Schlüssel in den Call Context geschrieben.
- **ReportListActions**: Dieser SIB wurde erweitert, so dass er für die Weiterleitung der Aktion "modify XML report" zuständig ist.
- **ShowAllSubmittedXMLReport**: Mit diesem SIB werden dem Begutachter, die über die XML Datei eingereichten Reports, in einer HTML-Seite angezeigt.
- **ShowArticleList**: Dieser SIB wurde um die Aktion "XML Report" erweitert. Diese Aktion ist ein- bzw. ausgeblendet, wenn das Feature "F-REV-13" aktiviert

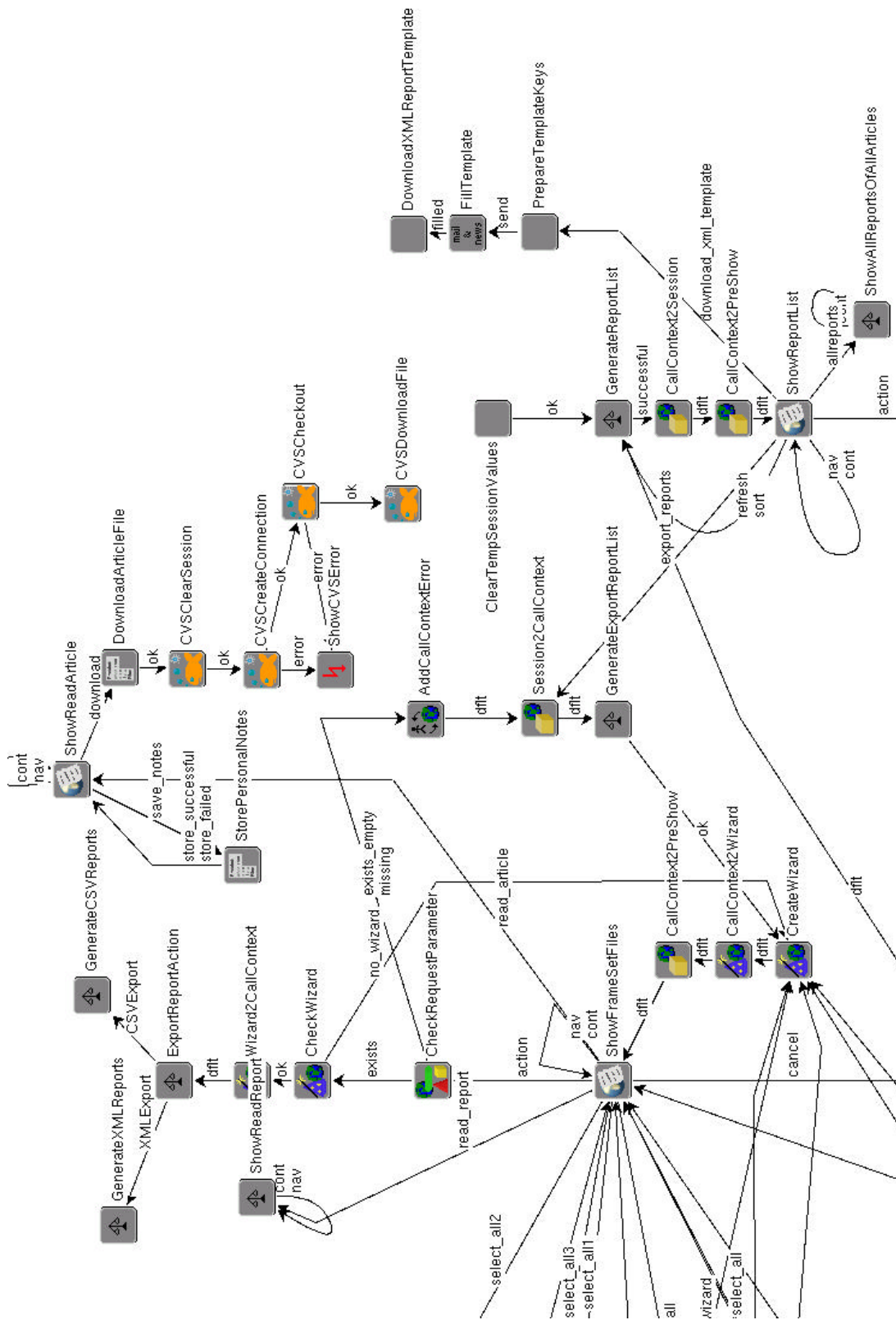


Abbildung 6.45: Export XML Reports

bzw. deaktiviert ist.

- **ShowReadReport**: Dieser SIB prüft zuerst, ob es ein Report Objekt im Call Context gibt. In diesem Fall holt er mittels der entsprechenden Report\_ID den Report aus der Datenbank. Die Daten dieses Reports werden durch die Daten des Report Objekts aus dem Call Context ersetzt und angezeigt.
- **ShowReportList**: Dieser SIB wurde um die Aktion "modify XML report" erweitert. Diese Aktion wird verwendet, um einen Report mittels einer XML Datei zu modifizieren.
- **ShowStartXMLReportsImport**: Dieser SIB ist vom Typ "ShowFrameSetFile". Er zeigt die HTML-Seite "startxmlreportsimport-cont.html" an.
- **ShowTaskList**: Dieser SIB wurde um die Aktion "XML Report" erweitert. Diese Aktion ist ein- bzw. ausgeblendet, wenn das Feature "F-REV-13" aktiviert bzw. deaktiviert ist.
- **TaskListActions**: Der SIB wurde modifiziert. Es wurde ein zusätzlicher Branch, "xml\_Report" eingefügt.
- **XMLImportReport**: In diesem SIB wird JDOM verwendet. JDOM liest die XML Datei unter Verwendung eines SAX-Parsers. Nach dem Parsen der XML Datei wird die JDOM-Dokumentstruktur (Baumstruktur) im Speicher aufgebaut. Die Inhalte der XML Datei stehen in verschachtelten Elementen. Jedes Element kann Inhalte in Form von Text und/oder Unter-Elementen enthalten. Jeder Inhalt wird unter einem bestimmten Schlüssel in den Call Context geschrieben. Der SIB hat als Parameter "parser\_classname" mit dem Wert: org.apache.xerces.parsers.SAXParser. Der Parser überprüft die XML Datei auf Validität (Übereinstimmung mit den Regeln des Schemas).
- **XMLImportReports**: Der SIB liest unter Verwendung des JDOM und des Parsers "Xerces" die eingegebene XML Datei und erstellt einen angepassten Objektbaum. Dieser Objektbaum wird im Hauptspeicher gehalten. Dann werden die Inhalte der XML Datei gelesen und unter bestimmten Schlüsseln in den Call Context geschrieben (siehe Abbildung 6.46).

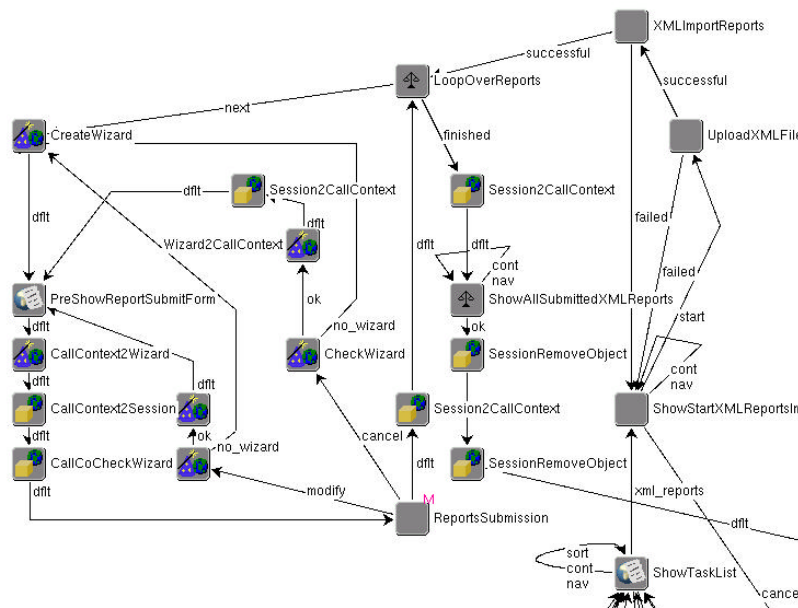


Abbildung 6.46: Import XML Reports

Um die von den oben beschriebenen SIBs generierten Informationen anzuzeigen, wurden dazu noch einige WML-Dateien geschrieben oder modifiziert.

- `allxmlreports-cont.wml`: Diese WML-Seite zeigt alle Informationen über die Reports an, die der Begutachter mittels der XML-Datei eingereicht hat.
- `articlelist-cont.wml`: Diese Datei wurde modifiziert. Es wurden ein neuer Link und eine neue Aktion hinzugefügt. Der Link "Import XML Reports" bzw. die Aktion "XML Report" ermöglichen dem Begutachter mittels einer XML Datei mehrere Reports bzw. einen Report einzureichen.
- `exportreports-cont.wml`: Die WML-Seite liest die ReportIDs und die ArticleIDs aus einem Vektor. Sie zeigt dem Begutachter die Reportliste zum Exportieren an. Durch den "CSVExport" bzw. "XMLReport" Button wird das Hochladen der selektierten Reports als CSV bzw. XML Datei gestartet.
- `reportlist-cont.wml`: In dieser Datei wurden zwei Links und eine Aktion hinzugefügt. Der Link "Download XML Template" führt zum Downloaden des XML-Templates. Die Aktion "modify XML report" ermöglicht dem Begutachter mittels einer XML Datei einen Report zu modifizieren. Der Link "Export Reports" führt zur "Export Report" Seite, in der der Benutzer die selektierten Reports als

CSV oder als XML Datei exportieren kann.

- `reportsubmitconfirm-cont.wml`: Die WML-Seite wurde durch einen Button "Skip Report" erweitert. Dieser Button wird angezeigt, wenn mehrere Reports mittels einer XML Datei eingereicht werden müssen. Es ermöglicht dem Begutachter zum nächsten Report zu springen, wenn er den aktuellen Report nicht einreichen möchte.
- `rolepermissions.inc`: Es wurde ein neues Feature in "Report Management" eingefügt. Es handelt sich um "F-REV-13". Beim Aktivieren bzw. Deaktivieren des Features wird die Aktion "XML Report" sowohl in **TaskList** als auch in **ArticleList** ein- bzw. ausgeblendet.
- `startmodifyxmlreportimport-cont.wml`: Diese WML-Seite stellt ein Eingabefenster dar, in welchem die XML Datei eines modifizierten Reports eingegeben werden soll.
- `startxmlreportimport-cont.wml`: In dieser WML-Seite wird ein Eingabefenster dargestellt, in dem die XML Datei eines einzureichenden Reports eingegeben werden soll.
- `startxmlreportsimport`: Diese WML-Seite stellt ein Eingabefenster dar, in welchem die XML Datei eines einzureichenden Reports eingegeben werden soll.
- `submitreport-cont.wml`: Diese WML-Seite wurde durch einen "Skip Report" Button erweitert, so dass der Begutachter die Möglichkeit hat, ohne den aktuellen Report einzureichen, zum nächsten Report zu springen.
- `tasklist-cont.wml`: Diese WML-Seite wurde um die Links "Download XML Template", "Import XML Reports" und die Aktion "XML Report" erweitert. Um das XML Template zu downloaden, wird der Link "Download XML Template" benutzt. Der Link "Import XML Reports" bzw. die Aktion "XML Report" ermöglichen dem Begutachter mittels XML Datei mehrere Reports bzw. einen Report einzureichen.

#### 6.4.4 Testphase

Der gesamte Workflow wurde getestet. Zu Beginn können die Begutachter das XML-Template für das Verfassen von Reports runterladen. Dies geschieht durch das Anklicken des Links "Download XML Template" von der ArticleList oder TaskList-Seite (siehe

Abbildung 6.47).

### Your Personal Task List

This page shows you the list of all the currently outstanding tasks you are responsible for. It represents your personal view on all articles in the journal workflow showing

1. which reviews you are responsible for, and
2. which next step you have to perform within each paper's workflow

Please

- confirm that you accept the review.

In exceptional cases (e.g. if one of the authors was/is your student or supervisor, or works in the same group, or if there are similar strong indications against your impartial review **reviewing** single papers. In this case you **MUST** explain the reason for not **reviewing** this paper.

- Only once you have **accepted reviewing** a paper you can access the "submit report" or "XML Report" actions.
- There are two possibilities to submit a report: Either online with clicking on "**submit report**" or with help of a xml file by using the "**XML Report**" action from the taskactions.
- If you like to submit several reports with help of a xml file, you should click on "**Import XML Reports**".

Here you can download the xml template: [Download XML Template](#)

In case you want to submit a several reports please click here: [Import XML Reports](#)

Article ID	File to download	Comments	Arrival of Task	Deadline	Task
003	003_jpr.txt (submitted: Tue Jul 05 11:56:17 CEST 2005)	Please review this paper and submit a report about it.	2005-07-07	2006-01-01	• <a href="#">read article</a> • <a href="#">submit report</a> • <a href="#">XML Report</a>
004	004_jpr.txt (submitted: Tue Jul 05 11:57:31 CEST 2005)	Please review this paper and submit a report about it.	2005-07-06	2006-01-01	• <a href="#">read article</a> • <a href="#">accept/reject res</a>

Abbildung 6.47: Der Link Download XML Template

Die Modifizierung der Rollen ist auch erfolgt: Das Feature "XML Report" (F-REV-13) funktioniert einwandfrei. Man wählt das Feature "F-REV-13" an bzw. ab. Die Änderung wird gespeichert und das jeweilige Feature ein bzw. ausgeblendet. Beim Einschalten des Features "F-REV-13" wird der Link "XML Report" innerhalb der Tasktabelle sowie der Artikeltable in der Aktionsspalte angezeigt (Abbildung 6.47). Durch Anklicken des Links "XML Report" gelangt der Benutzer zur "Report Import"-Seite (siehe Abbildung 6.48).

**ArticleID 003**

**Import Report**

Please upload the xml file which you filled with your report.

XML File:

Abbildung 6.48: Die Import XML Report –Seite

Dort kann er seinen XML Report hochladen und validieren. Wenn keine Fehler auftreten, wird der Inhalt der XML-Datei auf die "SubmitReport"-Seite gesetzt. Ansonsten meldet der Dienst diese Fehler und die Art und Weise wie man diese korrigieren kann. Dann kann der Begutachter seinen Report nochmals überarbeiten und einreichen. Im Fall einer

XML-Datei mit mehreren Reports benutzt der Begutachter den Link "Import Reports" in der "TaskList" oder "ArtikelList"-Seite und es wird eine "Reports Import"-Seite angezeigt. Nach dem Hochladen und dem Validieren der XML Datei werden die Reports nacheinander eingereicht. In diesem Fall haben die Begutachter die Möglichkeit Reports der XML-Datei einzeln zu überarbeiten und zum Schluss wird eine Bestätigung angezeigt, die alle gesendeten Reports zusammenfaßt (siehe Abbildung 6.49).

Show all submitted reports in xml file

---

**Article018: new article**

*Report 018-R3*

Reviewer Schmidt, Alexa

SUMMARY OF EVALUATION

Overall Quality Excellent

RECOMMENDATION

Score Strong Rejected (1)  
Confidence Medium(2)

COMMENTS TO BE WITHHELD FROM AUTHOR(S)

comments 012

READER INTEREST

Is the paper of current interest to a reasonable segment of the journal? Perhaps  
Relative to the current level of reader interest in the paper, how is this interest likely to change during the next five years? Diminishing interest  
Within its particular field of specialization, is the topic of the paper considered important? Not really

CONTENT

Is the paper technically sound? Only partially  
How would you describe the technical depth of the paper? Appropriate for someone working in the field  
Does the paper make a tangible contribution to the state-of-the-art in its field? No  
Is the bibliography adequate? No  
To what extent is the material in the paper likely to be used by other researchers and practitioners? Small

PRESENTATION

Is the abstract an appropriate and adequate digest of the work presented? No  
Does the introduction clearly state the background and motivation in terms understandable to the non-specialist? Probably  
How would you rate the overall organization of the paper? Poor  
Relative to the technical content, is the length of the paper appropriate? No, should be shortened  
Is the English satisfactory? No  
How readable is the paper for a computer scientist or engineer who is not a specialist in this particular field? Unreadable  
Disregarding the technical content, how would you regard the quality of presentation? Poor

DETAILED COMMENTS FOR AUTHOR(S)

comments 004

---

**Article019: new article**

*Report 019-R3*

Reviewer Schmidt, Alexa

SUMMARY OF EVALUATION

Overall Quality Fair

Abbildung 6.49: Bestätigungsseite der eingereichten Reports in der XML Datei

Ein anderer Teil der Testphase bestand darin, das Exportieren von Reports als XML sowie als CSV Datei zu überprüfen. Dazu dient der Link "Export Reports" von der ReportList-Seite (siehe Abbildung 6.50).

## List of Reports

To get an overview of all reports click here: [all reports](#)

**Refresh** (last refresh: *Thu Jul 07 16:45:46 CEST 2005*)

Here you can download a xml template: [Download XML Template](#)

In case you want to export one or a several reports as xml or csv file please click here: [Export Reports](#)

ID	Referee	Subreviewer	Score	Confidence	Date	Action
005-R1	Mueller, Bell		2	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
006-R3	Mueller, Bell		2	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
007-R2	Mueller, Bell		2	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
008-R3	Mueller, Bell		2	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
009-R1	Mueller, Bell		0	3	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
010-R3	Mueller, Bell		1	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
011-R3	Mueller, Bell		2	2	2005-07-05	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>
015-R1	Mueller, Bell	kk	4	3	2005-07-06	<a href="#">read article</a> • <a href="#">read report</a> • <a href="#">modify report</a> • <a href="#">modify XML report</a>

Abbildung 6.50: Der Link Export Reports in der ReportList–Seite

Nach dem Anklicken dieses Links wird die "Export Reports"-Seite angezeigt, die alle geschriebenen Reports von einem Begutachter enthält (siehe Abbildung 6.51).



## Export Reports

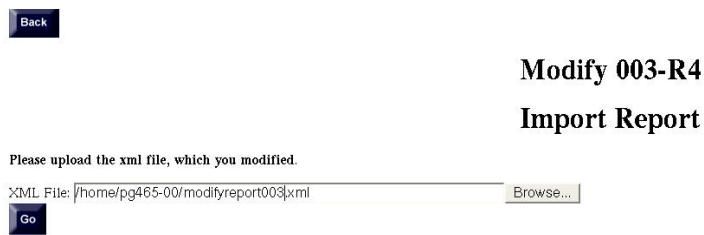
Here You can select one or several reports and export them as xml or csv file.

ArticleID	Report 1	Report 2	Report 3
	<input type="checkbox"/> select 1	<input type="checkbox"/> select 2	<input type="checkbox"/> select 3
003	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
004	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
005	<input type="checkbox"/> 005-R1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
006	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> 006-R3
007	<input checked="" type="checkbox"/>	<input type="checkbox"/> 007-R2	<input checked="" type="checkbox"/>
008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> 008-R3
009	<input type="checkbox"/> 009-R1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
010	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> 010-R3
011	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> 011-R3
013	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
015	<input type="checkbox"/> 015-R1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abbildung 6.51: Die Export–ReportList

Um die Reports als XML oder als CSV zu exportieren, muss der Begutachter mindestens einen Report selektieren. Durch das Anklicken des Buttons "XMLExport" oder "CSVExport" werden die Reports in eine XML–Datei oder CSV–Datei exportiert.

Im letzten Teil der Testphase wurden die geschriebenen Reports mit Hilfe einer XML-Datei modifiziert. Durch Anklicken des Links "Modify XML Report" von der "ReportList"-Seite wird die "Modified XML Report"-Seite angezeigt. Dort kann der Begutachter seinen XML-Report hochladen und validieren (siehe Abbildung 6.52).



The screenshot shows a web interface for modifying an XML report. At the top left, there is a dark blue button labeled "Back". To the right, the text "Modify 003-R4" and "Import Report" is displayed in a bold, black font. Below this, a message reads "Please upload the xml file, which you modified." Underneath the message is a text input field containing the file path "/home/pg465-00/modifyreport003.xml". To the right of the input field is a button labeled "Browse...". Below the input field is a dark blue button labeled "Go".

Abbildung 6.52: Import des modifizierten XML Reports

Wenn keine Fehler auftreten, wird der modifizierte Report angezeigt und daraufhin vom Begutachter gespeichert werden.

# Kapitel 7

## Fazit

Ziel der Projektgruppe war es, den bereits bestehenden OJS um einige Features zu erweitern. Auch wenn diese Änderungen im System für den Benutzer gering wirken, waren es große Teile des Systems, die von uns angepasst wurden. Die gesetzten Ziele wurden wie geplant erreicht.

Während der Arbeit fiel immer wieder auf, dass der OJS ein mächtiges, gewachsenes System ist - mit allen Vor- und Nachteilen, die daraus resultieren. Oft wurde uns die Arbeit durch vorhandene Module und SIBs erleichtert, doch ebenso oft stolperten wir über Teile, die noch auf Technologien von vor über 4 Jahren basierten. Manchmal fanden wir sogar konkurrierende Systeme vor, wo es galt, das neuere der Beiden zu finden, zu nutzen und etwas "aufzuräumen" (z.B. das Logging-System). Doch mit jeder angepassten Klasse vertiefte sich auch unser Wissen über Servlets, das ABC und v.a. Datenbanken.

Bei Fragen und Problemen standen die Betreuer immer zur Verfügung und hatten eine große Geduld mit uns, ob es um Anfänger-Fragen oder später um ausgewachsene Probleme ging. Die Atmosphäre der Gruppe war gut und die Betreuer verstanden es, ohne übermäßigen Druck die Motivation aufrecht zu erhalten und eine Identifikation mit dem Projekt und dem Dienst OJS zu schaffen.

Die Entwicklung des OJS ist noch lange nicht zu Ende und es gibt sicherlich noch viele Dinge, die verbessert werden können. Geplant ist auch eine Rückführung auf eine gemeinsame Linie mit dem Schwesterdienst OCS. Dadurch könnten die Weiterentwicklungen beider Dienste vereinigt und mit weniger Aufwand an beiden gemeinsam gearbeitet werden.



# Literaturverzeichnis

- [1] *JournalDemo: Ein Beispiel des Online Journal Service Dienstes*, <http://sttt.cs.uni-dortmund.de/pg465ojs/servlet/Journal>
- [2] B. Lindner, T. Margaria, B. Steffen. *Ein personalisierter Internetdienst für wissenschaftliche Begutachtungsprozesse*, GI-VOI-BITKOM-OCG-TeleTrusT Konferenz Elektronische Geschäfts-prozesse (eBusiness Processes), Universität Klagenfurt, 24.-25. September 2001, <http://syssec.uni-klu.ac.at/EBP2001/>
- [3] Omar Zaouaoui Elidrissi. *Objektorientiertes Design und Modellierung*, Bommerholz den 10.Oktober 2004
- [4] Marc von Renteln. *Grundlagen der Programmierung in Java*, Bommerholz den 10.Oktober 2004
- [5] Abdelazziz Aic Benhaddou. *Tools für die Software-Entwicklung*, Bommerholz den 10.Oktober 2004
- [6] Mikhail Farnassov. *Ant*, Bommerholz den 10.Oktober 2004
- [7] Dominik Brink. *Erstellen von Dokumenten mittels L<sup>A</sup>T<sub>E</sub>X*, Bommerholz den 10.Oktober 2004
- [8] Abdelmounaim Ramadane. *Tomcat 4.x als Applikations-Server*, Bommerholz den 10.Oktober 2004
- [9] Alexander Powolozki. *Entwicklung von Servlets in Java*, Bommerholz den 10.Oktober 2004
- [10] Markus Korte. *GUI für (dynamische) Webapplikationen*, Bommerholz den 10.Oktober 2004
- [11] Philipp Langer. *Design von Webapplikationen mittels ABC/EWIS*, Bommerholz den 10.Oktober 2004
- [12] Taieb Elfakiri. *Dokument-Management mittels XML*, Bommerholz den 10.Oktober 2004

- [13] Holger Hopf. *Konzepte der Datenbank PostgreSQL*, Bommerholz den 10.Oktober 2004
- [14] Ganna Ostapchuk. *das Entscheidungssystem OJS*, Bommerholz den 10.Oktober 2004
- [15] B. Steffen, M. Karusseit, M. Bajohr 2004 *Projektgruppe 465 Antrag*, Lehrstuhl 5, Universität Dortmund
- [16] PG440 *Endbericht der PG 440 EUREKA*, Lehrstuhl 5, Universität Dortmund