University of Salzburg
Institute of Computer Science

PS Software Engineering Part I

InfoWarrior

Online Information System for News Agencies

Personal Documentation

Thomas Aschauer

Date: December 19, 2001
Version: 1.0

Contents

1 Introduction **8**
1.1 Document Structure 8

2 Meeting 1 (October 31, 2001) **9**
2.1 Meeting Tasks 9
2.2 Personal Contribution 9
2.3 Meeting Protocol 9
2.4 Meeting Results 9

3 Meeting 2 (November 7, 2001) **10**
3.1 Meeting Tasks 10
3.2 Personal Contribution 10
3.3 Meeting Protocol 10
3.4 Meeting Results 10

4 Meeting 3 (November 14, 2001) **11**
4.1 Meeting Tasks 11
4.2 Personal Contribution 11
4.3 Meeting Protocol 11
4.4 Meeting Results 12

5 Meeting 4 (November 21, 2001) **13**
5.1 Meeting Tasks 13
5.2 Personal Contribution 13
5.3 Meeting Protocol 13
5.4 Meeting Results 14

6 Meeting 5 (December 5, 2001) **15**
6.1 Meeting Tasks 15
6.2 Personal Contribution 15
6.3 Meeting Protocol 16
6.4 Meeting Results 16

7 Meeting 6 (December 12, 2001) **17**
7.1 Meeting Tasks 17
7.2 Personal Contribution 17
7.3 Meeting Protocol 18
7.4 Meeting Results 18

A Introduction (Task 1) **19**
A.1 Rough System Requirements 19

B The CRC session **20**
B.1 Fundamentals of the CRC Session 20
B.2 Planning the CRC Session 20
B.3 Scenarios 20
B.3.1 Information Grabbing Process 20
B.3.2 Information Presentation Process 22
B.4 Identified Classes 23
B.5 Role Allocation 23
B.6 Goals 23
B.7 Open Questions 24

C The Second CRC Session (Task 3) **25**
C.1 Session Goals 25
C.2 Essential Use Case Modeling 25
C.2.1 Introduction 25
C.3 Essential Use Cases for InfoWarrior 27
C.3.1 Use Case 1 27
C.3.2 Use Case 2 27
C.3.3 Use Case 3 28
C.3.4 Use Case 4 29
C.3.5 Use Case 5 30
C.3.6 Use Case 6 30
C.3.7 Use Case 7 31
C.4 Essential User Interface Prototyping 31
C.4.1 Introduction 31
C.4.2 First thoughts 32

D The Identified Use Case Diagrams **34**
D.1 Competence Management 34
D.2 Information Delivery 35
D.3 Information Presentation 35
D.4 Information Gathering 36

E Workflow System **37**
E.1 Activity Diagram - Competence Management Workflow 37
E.2 Activity Diagram - Story Workflow 38

F Essential User Interface Prototypes **39**
F.1 Login 39
F.2 User Profile 40
F.3 Check Workflow tasks 40
F.4 Story Visualization 41
F.5 Story Editing 41
F.6 Competence Management 42

G Analysis Class Diagrams **43**
G.1 WorkflowSubSystem . . . 43
G.1.1 Description . . . 43
G.1.2 Used Patterns . . . 43
G.1.3 Class Diagram . . . 43
G.2 WorkflowTasks . . . 45
G.2.1 Description . . . 45
G.2.2 Class Diagrams . . . 45
G.3 RequestHandling . . . 45
G.3.1 Description . . . 45
G.3.2 Used Patterns . . . 46
G.3.3 Class Diagrams . . . 46
G.4 Actions . . . 46
G.4.1 Description . . . 46
G.4.2 Class Diagrams . . . 47
G.5 System users . . . 47
G.5.1 Description . . . 47
G.5.2 Class Diagrams . . . 48

H Sequence Diagrams **49**
H.1 Introduction . . . 49
H.2 RequestHandling . . . 49
H.3 WorkflowSubSystem . . . 51
H.3.1 Check Workflow Process . . . 51
H.3.2 Query Stories . . . 51
H.3.3 Enter Vacations . . . 52

I Activity Diagrams **54**
I.1 Introduction . . . 54
I.2 Login action . . . 54
I.3 Save Story action . . . 55

J User Interface Prototypes **56**
J.1 User Interface Flow Diagram . . . 56
J.2 HTML User Interface . . . 57
J.2.1 Login Page . . . 57
J.2.2 New Registration . . . 57
J.2.3 Main Menu . . . 58
J.2.4 Check WorkFlow Tasks . . . 58
J.2.5 Edit/View Stories . . . 59
J.2.6 Calendar . . . 59
J.2.7 Competence Management . . . 60
J.2.8 Visualization (Ordered News) . . . 60
J.2.9 User Customization . . . 61
J.2.10 Billing Statistics . . . 61
J.3 WML User Interface . . . 62
J.3.1 Default Home Page . . . 62

J.3.2 Login Screen 62
J.3.3 New Registration 63
J.3.4 Registered User 63
J.3.5 Reporter or Editor 64
J.3.6 Adding a new Story 64

List of Figures

C.1 User Interface Prototype: Informationgrabber 32
C.2 User Interface Prototype: Informationrequest 33

D.1 Use Case Diagram: Competence Management 34
D.2 Use Case Diagram: Information Delivery 35
D.3 Use Case Diagram: Information Presentation 35
D.4 Use Case Diagram: Information Gathering 36

E.1 Activity Diagram: Competence Management Workflow 37
E.2 Activity Diagram: Story Workflow 38

F.1 Essential User Interface Prototype: Login 39
F.2 Essential User Interface Prototype: User Profile 40
F.3 Essential User Interface Prototype: Workflow task check 40
F.4 Essential User Interface Prototype: Story Visualization 41
F.5 Essential User Interface Prototype: Story Editing 41
F.6 Essential User Interface Prototype: Competence Management 42

G.1 Analysis Class Diagram: WorkflowSubSystem 44
G.2 Analysis Class Diagram: WorkflowTasks 45
G.3 Analysis Class Diagram: RequestHandling 46
G.4 Analysis Class Diagram: Patterns used in RequestHandling 47
G.5 Analysis Class Diagram: Actions 48
G.6 Analysis Class Diagram: System Users overview 48

H.1 Sequence Diagram: Request Handling 50
H.2 Sequence Diagram: Check Workflow Process 51
H.3 Sequence Diagram: Query Stories 52
H.4 Sequence Diagram: Enter Vacations 53

I.1 Activity Diagram: Login Action 54
I.2 Activity Diagram: Save Story Action 55

J.1 User Interface Flow Diagram 56
J.2 HTML user interface: Login Page 57
J.3 HTML user interface: New Registration 57
J.4 HTML user interface: Main Menu 58
J.5 HTML user interface: Check WorkFlow Tasks 58
J.6 HTML user interface: Edit/View Stories 59

J.7 HTML user interface: Calendar 59
J.8 HTML user interface: Competence Management 60
J.9 HTML user interface: Visualization 60
J.10 HTML user interface: User Customization 61
J.11 HTML user interface: Billing Statistics 61
J.12 WML user interface: Default home page 62
J.13 WML user interface: login screen 62
J.14 WML user interface: request new registration 63
J.15 WML user interface: registered user logged in 63
J.16 WML user interface: Reporter or Editor logged in 64
J.17 WML user interface: enter a new story 64

List of Tables

Chapter 1

Introduction

This chapter gives an overview about the personal documentation and its structuring.

1.1 Document Structure

In this personal documentation each chapter represents a meeting. Each chapters sections are the task we had to do, the personal contribution to the group document and a protocol of the corresponding meeting including the results. Due to the fact that really big parts of the group documentation had been done by the whole group (mainly the writing of the first six chapters), the group document is included in the appendix and references are used to point out which parts are the results of my personal preparation. For completion purposes also these parts are included here, that have been done by other project members, for example the user interface prototype that can be seen in appendix J.

Chapter 2

Meeting 1 (October 31, 2001)

2.1 Meeting Tasks

The preparation of the first meeting should be the planning of a first CRC session. This is for clarifying the system requirements.

2.2 Personal Contribution

The first meeting preparation we did together, so it is very difficult to differentiate which part of preparation was done by whom.

2.3 Meeting Protocol

I missed the first CRC session because of a business trip to Sweden, therefore this protocol would be just a transcript of other team member's protocols.
Present persons:

Dr. Roland Schwaiger (customer representative)

Thorsten Engl (project leader, vendor representative)

Christian Zödl (software architect)

Erich Eichberger (chief of protocol)

2.4 Meeting Results

According to the information I received the first CRC session was very successful, although the group did a little misinterpretation of the assigned task. We planned the CRC session in very technical terms, but this first meeting with the ordering customer should have been for identifying the whole business problem instead of identifying classes. But as of the very good session preparation almost all business requirements had been matched.

Chapter 3

Meeting 2 (November 7, 2001)

3.1 Meeting Tasks

For the second meeting, an additional CRC session had to be prepared in order to get deeper thoughts of the target system and its environment.

3.2 Personal Contribution

As with the first meeting's preparation, the documentation for the second meeting we wrote together, therefore a differentiation of each one's contribution is really different. Anyhow, parts that were mainly done by me are the general discussion of an essential user interface prototype (see appendix C.4) and the introduction to essential use case modeling that can be found in appendix C.2.

3.3 Meeting Protocol

Present persons:

Dr. Roland Schwaiger (customer representative)

Thomas Aschauer (project leader, vendor representative)

Erich Eichberger (software architect)

Thorsten Engl (software developer)

Christian Zödl (chief of protocol)

3.4 Meeting Results

The second CRC-Session was very successful, because we clarified further system requirements and identified new use-cases and gui requirements. One of the biggest steps forward to a reasonably complete system specification was the discovery that our system should have some abilities to manage business workflow in the target companies.

Chapter 4

Meeting 3 (November 14, 2001)

4.1 Meeting Tasks

- Essential Use Cases
- Essential User Interface Prototype

4.2 Personal Contribution

In the preparation for our third meeting I was involved with are mainly the workflow subsystem (can be found in appendix E) and the planning of the Essential User Interface Prototype that is included in appendix F.

4.3 Meeting Protocol

Present persons:

Dr. Roland Schwaiger (customer representative)

Thomas Aschauer

Erich Eichberger (project leader)

Thorsten Engl

Christian Zödl

4.4 Meeting Results

Spoken in general, the meeting was successful again. The main points that were suggested as weak were the following:

- The naming of some use cases were not clear
- No clear flow of the essential user interface prototype was given
- There was a little error in the definition of the second workflow (see fig E.2 for the corrected version)

Chapter 5

Meeting 4 (November 21, 2001)

5.1 Meeting Tasks

- Analysis Class Diagrams

5.2 Personal Contribution

In recent years it emphasized that good software architectures are often built or enhanced by using pattern technologies. Using patterns in early stages of the software building process is essential for a good analysis of the system to be built. As I am really well with pattern and their appliance, my main task in the preparation for the fourth meeting was the search for pattern (mostly analysis pattern in that case) that could help us building a good software architecture, and their appliance when building the analysis class diagrams. The analysis class diagrams are included in appendix G.

5.3 Meeting Protocol

Present persons:

Dr. Roland Schwaiger (customer representative)

Thomas Aschauer

Erich Eichberger

Thorsten Engl

Christian Zödl (project leader)

5.4 Meeting Results

Although the meeting could be seen as very successful, there were a few points that needed further work:

- The documentation of the used pattern was detailed as it should be
- The mechanism of session handling was not clear, additional work in clarifying the details are needed

Chapter 6

Meeting 5 (December 5, 2001)

6.1 Meeting Tasks

- User Interface Prototype
- Sequence Diagrams

6.2 Personal Contribution

To give additional documentation to the analysis class diagram that was prepared for Meeting 4, some sequence diagrams had to be drawn and described. That was the part the I was involved with. The sequence diagrams, that can be found in appendix H, are used to document four aspects of the system that are really complex:

The Request Handling Mechanism Request handling and the decision which action to take for a request are really complex processes and are documented in that use case

Check Workflow Process The use of the workflow subsystem in request handlers (called actions) is documented in this diagram

Query Stories The process of querying for a story is shown here including the different cases for general queries and queries that use the workflow subsystem

Enter Vacations This exemplarily case is used to show how a sequence of requests works together

6.3 Meeting Protocol

Present persons:

Dr. Roland Schwaiger (customer representative)

Thomas Aschauer

Erich Eichberger

Thorsten Engl (project leader)

Christian Zödl

6.4 Meeting Results

The meeting on the 5th of December was not as successful as the other meetings. There was a big lack in the presentation of the user interface prototype, and also some unclear things in the sequence diagrams. Therefore, the preparation for the next meeting should contain a new user interface prototype and corrected sequence diagrams.

Chapter 7

Meeting 6 (December 12, 2001)

7.1 Meeting Tasks

- New User Interface Prototype
- Sequence Diagrams
- Activity Diagrams

7.2 Personal Contribution

Because a big job for creating the sequence diagrams had already be done in preparing meeting 5, my main job was the building of the activity diagrams, as shown in figures I.2 and I.1, and because I was the project leader the presentation on the overhead was held by me. Additionally I was involved in designing the user interface flow diagram (see figure J.1).

The Activity Diagrams are not very complicated because the essential processes, like request handling and handling the workflow subsystem, are already documented by sequence diagrams. But the view of activity diagrams matched perfectly the requirements or documenting the internal process of an action, so we picked out two exemplary actions:

Login Action This diagram shows how the logon to InfoWarrior should look like, and it gives an impression how the control path of an action should look like

Save Story Action The save story action is really representative for all actions managing stories (e.g. cancel, release...)

7.3 Meeting Protocol

Present persons:

Dr. Roland Schwaiger (customer representative)

Thomas Aschauer (project leader)

Erich Eichberger

Thorsten Engl

Christian Zödl

7.4 Meeting Results

I think the sixth meeting was really successful, although there are two things that we should think of:

- The naming of out competence management should be changed.
- We did not realize that the difference between HTML user interface (see appendix J.2) and the WML user interface (see appendix J.3) are so wide, that a direct mapping between actions and views is nearly impossible. Therefore an additional mechanism of mapping views to actions has to be introduced.

Appendix A

Introduction (Task 1)

This chapter gives an overview about motivation for an online information system for news agencies.

A.1 Rough System Requirements

A News Agency (called "e-News") wants to redesign their information grabbing and information presentation processes with involvement of IT-Systems.
Potential users of the system are persons that collect news and other that retrieve that information. Beside them, also third party systems should be able to gather information out of InfoWarrior.
Therefore we can identify two possible scenarios:

- InfoWarrior will deliver the gathered information to other or proprietary systems by using XML-Technologies.
- Discrete persons will retrieve the gathered information from InfoWarrior. As a prerequisite for this task the information has to be prepared in an ergonomic manner.

Another special requirement is easy maintenance of InfoWarrior, which presumes that there will be no local system installations.
InfoWarrior should also be able to prepare information for the need of individual persons that are accessing the data.

Appendix B

The CRC session

This chapter describes the fundamentals till the finished CRC session.

B.1 Fundamentals of the CRC Session

In effort to get the whole information from a problem the crc card system is a very good invention because information from a customer is mostly incomplete and not very precise.
The CRC session is in general like a "role game". A "player" (developer) represents a class. By stepping through predefined scenarios correlations between the different classes are quite simple found.
A customer joining this session sees the process of the program and is able to request some modifications.

B.2 Planning the CRC Session

The first task in a crc session is to prepare scenarios, which cover all possible steps of the problem.
By stepping through the scenarios possible classes are identified.

B.3 Scenarios

B.3.1 Information Grabbing Process

News Entry with Review

- Reporter Harry Hirsch was in New York when the terror attack happened. He wants to add a headline to InfoWarrior.
- Logging on.
- Input of the headline (plus category, priority and state). Then submit.
- Logging off.

- Hansl vom Dienst, the responsible editor, is logging on.
- He gets a message from InfoWarrior that Harry Hirsch added a headline that is ready for release.
- HvD reviews the headline and releases it.
- Logging off.

News Entry with Post Processing

- Reporter Harry Hirsch heard from an intelligencer that Miss Levinsky bought a cigar shop called "Big Bill".
- Input of the story (plus category, priority and state). Then submit.
- Hansl vom Dienst is logging on.
- He gets a message from InfoWarrior that Harry Hirsch added a story that is ready for release.
- HvD reviews the story and recognizes that there is not enough "sex and crime" in it. He decides that further information gathering is needed.
- Send story for post processing back to Harry.
- Harry gets a message to post process the story.
- After editing the story he sends it to HvD.
- When HvD gets the control message, he checks the story again and releases it.

News Entry with Cancellation

- Reporter Harry Hirsch is in Afghanistan and writes a report about shaved afghan women.
- Input of the report (plus category, priority and state). Then submit.
- HvD gets a message from InfoWarrior that Harry Hirsch added a report that is ready for release.
- HvD reviews the report and cancels it.
- A message is delivered to Harry.

News Entry split to two Reporters

- Reporter Harry Hirsch writes the following headline to InfoWarrior: "Boris Becker and Anna Kournikova in flagranti?"
- HvD gets a message from InfoWarrior that Harry Hirsch added a headline that is ready for release.
- HvD reviews the headline and decides that a report has to be written.
- He sends the headline to Detlef Hirseklau with a comment that the report has highest priority.
- Detlef Hirseklau gets HvDs message, writes the report and sends it for release.
- HvD is very happy about the report and releases it.

B.3.2 Information Presentation Process

News Retrieval by Unknown User

- Wilma Willig surfs to the E-News homepage and requests the latest sport news.
- InfoWarrior gathers the information and sends the result page to Wilma

News Retrieval by Registered User

- Richi Rich is a registered user of E-News and wants to personalize his stock reports.
- Logging on.
- Downloading the XSLT template.
- After modifying the template to meet his own requirements he‘s making an upload of this file.
- He‘s requesting the new stock reports.
- InfoWarrior does the customized visualization of the reports.

News Retrieval by Third Party System

- A third party software installed at ÖKM (Österreichisches Kraftfahrer Magazin) needs different news from e-news, so they send an XML request to InfoWarrior.
- The received XML file is processed by the program, and the gathered news are converted to XML format again.
- This results are sent back to ÖKM.

B.4 Identified Classes

When preparing the CRC session we identified the following classes.

- InfoWarrior Request Handler
- InfoWarrior Submit Handler
- InfoWarrior Third Party System Handler
- Database
- News
- Message
- User
- XML engine
- XSLT engine
- File

B.5 Role Allocation

Engl Thorsten Role: Project Leader, Sales representative
Classes: InfoWarrior - Request Handler, InfoWarrior - Submit Handler, InfoWarrior - Third Party System Handler

Eichberger Erich Role: Software Engineer, System Analyzer
Classes: Database, XML engine, XSLT engine

Zödl Christian Role: Software Engineer, Protocol
Classes: News, Message, File, User

Schwaiger Roland Role: Customer

B.6 Goals

- Full problem definition
- Requirement clarification
- Delimit of requirements
- Minimize risk for misunderstandings
- Class identification
- First approaches to Use Cases

B.7 Open Questions

- Multiple selection of default XSLT types or user specific?
- User hierarchy?
- How many different categories?
- Processing requests from third party system with only a simple XML interpreter, or various formats?

Appendix C

The Second CRC Session (Task 3)

A second CRC session is of advantage to reach a higher state of clarification for the problem. Another point for a second session is to eliminate misapprehensions between the customer and the developers of the problem.

Another way to accomplish the elimination of such disaccordings is the usage of Essential User Interface Prototyping.

C.1 Session Goals

The primary goal of the second CRC session will be the identification of essential use cases with its corresponding actors. Beside, an essential user interface prototype should be discussed.

C.2 Essential Use Case Modeling

C.2.1 Introduction

A use case is a sequence of actions that provide a measurable value to an actor. Another way to look at it is that a use case describes a way in which a real-world actor interacts with the system. An essential use-case, sometimes called a business use case, is a simplified, abstract, generalized use case that captures the intentions of a user in a technology and implementation independent manner. An essential use case is a structured narrative, expressed in the language of the application domain and of users, comprising a simplified, generalized, abstract, technology free and implementation independent description of one task or interaction. An essential use case is complete, meaningful, and well designed from the point of view of users in some role or roles in relation to a system and that embodies the purpose or intentions underlying the interaction.

There are two basic flavors of use case models:

- Essential use case models
- System use case models

There are some basic differences between essential use case modeling and system use case modeling. First, system use cases contains many implementation details embedded in them, in contrast to essential use cases that are limited to describe the interaction of users with the system, not the system details. The writer of system use cases is analyzing and describing requirements imposed by the problem intermingled with implicit decisions about what the user interface is going to be like, whereas the writer of an essential use case does not. The second thing to notice is that the system use case makes references to screen and reports, and the essential use case does not. This reflects implementation details, someone has decided that the system will be implemented as screens, as opposed to HTML pages perhaps, and printed reports. However, the essential use case could just as easily have referred to major user interface elements, the essential version of screens and reports. Third, the system use case has more steps than the essential use case version. This in fact reflects the preferred style of writing use cases: Each use case step should reflect one step and one step only. There are several advantages to this approach: the use case becomes easier to test because each statement is easier to understand and to validate; alternate courses are easier to write because it is easier to branch from a statement when it does one thing only. Fourth, the use case steps are written in the active voice. For example, the statement "The registrar informs the student of the fees" is in active voice whereas "The student is informed of the fees by the registrar" is in passive voice. Writing in the active voice leads to sentences that are succinct.

Essential Use Case Models

An essential use case model, often referred to as a business or abstract use case model, models a technology-independent view of your behavioral requirements.

System Use Case Models

System use case models, also known as concrete use case models or detailed use case models, model your analysis of your behavioral requirements, describing in detail how users will work with your system including references to its user-interface aspects.

Conclusion

A use case is a sequence of actions that provide a measurable value to an actor. Another way to look at it is that a use case describes a way in which a real-world actor interacts with the system.

An essential use-case is a simplified, abstract, generalized use case that captures the intentions of a user in a technology- and implementation-independent manner. It is complete, meaningful, and well designed from the point of view of users in some role or roles in relation to a system and that embodies the purpose or intentions underlying the interaction.

C.3 Essential Use Cases for InfoWarrior

C.3.1 Use Case 1

Name: News Entry with Review

Category: Information grabbing

Preconditions: A reporter wants to add a story to InfoWarrior.

Postconditions: The story will be released.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Reporter Harry Hirsch was in New York when the terror attack happened. He wants to add a headline to InfoWarrior.
3. Logging on.
4. Input of the headline (plus category, priority and state).
5. Submit.
6. Logging off.
7. Hansl vom Dienst, the responsible editor, is logging on.
8. He retrieves his messages.
9. HvD reads a system generated message that a new story is waiting for release.
10. HvD reviews the headline and releases it.
11. Logging off.
12. The use case ends.

C.3.2 Use Case 2

Name: News Entry with Post Processing

Category: Information grabbing

Preconditions: A reporter wants to add a story to InfoWarrior.

Postconditions: The corrected story will be released.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Reporter Harry Hirsch heard from an intelligencer that Miss Levinsky bought a cigar shop.
3. Logging on.
4. Input of the story (plus category, priority and state).

5. Submit.
6. Hansl vom Dienst is logging on.
7. He retrieves his messages.
8. HvD reads a system generated message that a new story is waiting for release.
9. He searches the story.
10. HvD reviews the story and recognizes that there is not enough "sex and crime" in it. He decides that further information gathering is needed.
11. HvD sets the story's state to "post process".
12. Harry retrieves his messages.
13. He reads the message to post process the story.
14. HH searches the story.
15. After editing the story he sends it for release again.
16. HH logging off.
17. HvD reads a system generated message that the story is waiting for release again.
18. HvD reviews and releases the story.
19. HvD logging off.
20. The use case ends.

C.3.3 Use Case 3

Name: News Entry with Cancellation

Category: Information grabbing

Preconditions: A reporter wants to add a story to InfoWarrior.

Postconditions: The story will be canceled.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Reporter Harry Hirsch is in Afghanistan and writes a report about afghan women.
3. Logging on.
4. Input of the report (plus category, priority and state).
5. Submit.
6. Logging off.
7. Hansl vom Dienst, the responsible editor, is logging on.
8. He retrieves his messages.

9. HvD gets a message from InfoWarrior that Harry Hirsch added a report that is ready for release.
10. HvD reviews the report and cancels it.
11. InfoWarrior delivers a message to Harry.
12. Logging off.
13. The use case ends.

C.3.4 Use Case 4

Name: News Entry split to two Reporters

Category: Information grabbing

Preconditions: A reporter wants to add a story to InfoWarrior.

Postconditions: The story is released, written by two different reporters.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Reporter Harry Hirsch logging on.
3. Reporter Harry Hirsch writes the following headline to InfoWarrior: "Boris Becker and Anna Kournikova in flagranti?"
4. Logging off.
5. Hansl vom Dienst, the responsible editor, is logging on.
6. He retrieves his messages.
7. HvD gets a message from InfoWarrior that Harry Hirsch added a report that is ready for release.
8. HvD is searching for the headline.
9. HvD reviews the headline and decides that a report has to be written.
10. He sends the headline to Detlef Hirseklau with a comment that the report has highest priority.
11. Detlef Hirseklau is logging on.
12. He retrieves his messages.
13. Detlef Hirseklau reads HvDs message.
14. He writes the report and sends it for release.
15. Logging off.
16. HvD retrieves his messages.
17. HvD gets a message from InfoWarrior that the story is ready for release.
18. He is searching for the story.
19. Hansl releases the story.
20. Logging off.
21. The use case ends.

C.3.5 Use Case 5

Name: News Retrieval by Unknown User

Category: Information presentation

Preconditions: A user wants to retrieve information.

Postconditions: He gets a info presentation.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Wilma Willig surfs to the E-News homepage and requests the latest sport news.
3. InfoWarrior gathers the information and sends the result page to Wilma.
4. The use case ends.

C.3.6 Use Case 6

Name: News Retrieval by Registered User

Category: Information presentation

Preconditions: A user wants to retrieve information with personalized visualization.

Postconditions: The users visualization will be permanently changed and all the information will be presented in the personal manner.

Basic Course of Action: Use case sequence

1. Start of use case.
2. Richi Rich is a registered user of E-News and wants to personalize his stock reports.
3. Logging on.
4. Richi is going to personal configuration.
5. RR chooses another presentation template.
6. He‘s requesting the new stock reports.
7. InfoWarrior does the customized visualization of the reports.
8. Logging off.
9. The use case ends.

C.3.7 Use Case 7

Name: News Retrieval by Third Party System

Category: Information presentation

Preconditions: A third party system tries to gather information from InfoWarrior by using XML technologies.

Postconditions: The requested information is delivered.

Basic Course of Action: Use case sequence

1. Start of use case.
2. A third party software installed at ÖKM (Österreichisches Kraftfahrer Magazin) needs different news from e-news.
3. The third party software (TPS) sends authentification information.
4. InfoWarrior grants access.
5. TPS sends an XML request to InfoWarrior.
6. The received XML file is processed by the program, and the gathered news are converted to XML format again.
7. This results are sent back to ÖKM.
8. TPS is logging off.
9. The use case ends.

C.4 Essential User Interface Prototyping

C.4.1 Introduction

The user interface (UI) is the portion of software that a user directly interacts with. An essential user interface prototype is a low-fidelity model, or prototype, of the UI for your system it represents the general ideas behind the UI but not the exact details. Essential UI prototypes represent user interface requirements in a technology independent manner, just as essential use case models do for behavioral requirements. An essential user interface prototype is effectively the initial state, the beginning point, of the user interface prototype for your system. It models user interface requirements, requirements which are evolved through analysis and design to result in the final user interface design for your system.

There are two basic differences between essential user interface prototyping and traditional UI prototyping. First, the goal is to focus on users and their usage of the system, not system features. This is one of the reasons why essential use case modeling and essential user interface prototyping should be performed in tandem: they each focus on usage. Second, prototyping tools are very simple, including white boards, flip chart paper, and sticky notes. The minute that electronic technology is introduced to prototyping efforts a design decision about the implementation technology may habe been made. If an HTML development tool is used to build a user interface prototype then

immediately the design space to the functionality supported is narrowed within browsers. If Java development environment is chosen, then the design space has been narrowed to Java, the same occures if a Windows-based prototyping tool will be used. Understand the problem first, then solve it.

C.4.2 First thoughts

An Essential User Interface Prototype for the information grabbing process may result in this:

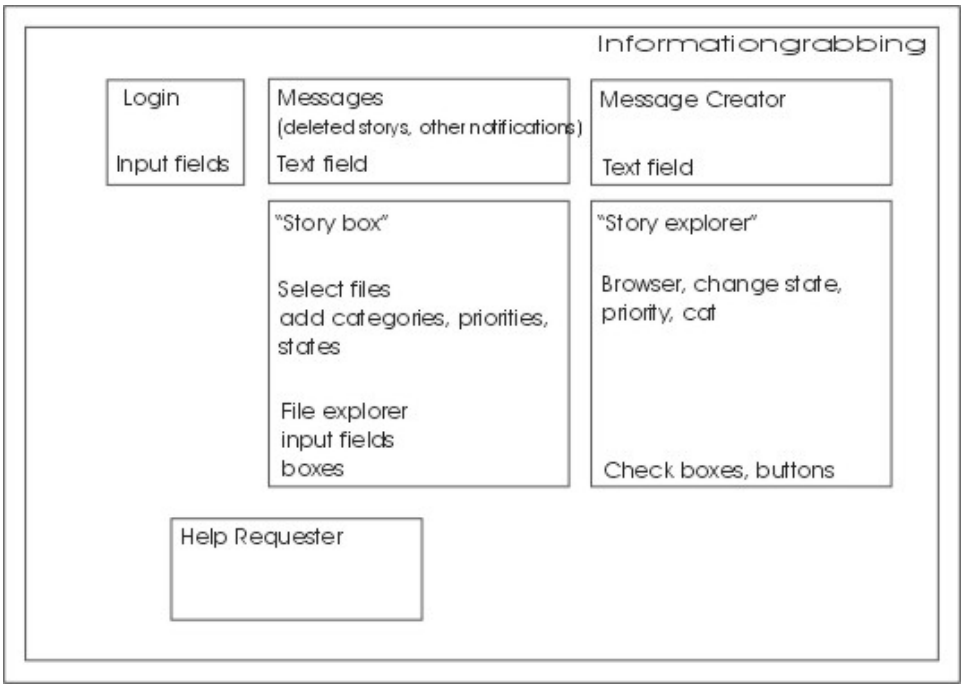

Figure C.1: User Interface Prototype: Informationgrabber

Another prototype for the information request process may look like this:

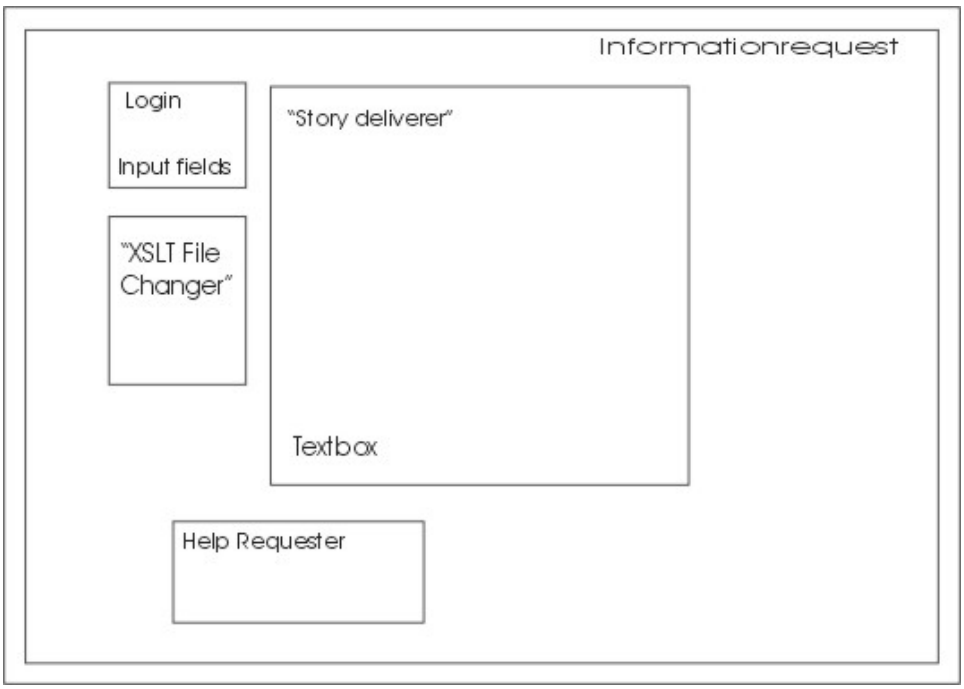

Figure C.2: User Interface Prototype: Informationrequest

Appendix D

The Identified Use Case Diagrams

By working through the scenarios and the essential use cases the following use cases were identified.

Due to the new software requirement that some workflow system functions should be included, we had to add a new method for handling such cases. Therefore the old message driven quasi-workflow machine will be replaced by a template driven full enhanced workflow engine.

D.1 Competence Management

Competence management enables a secretary to manage illness or vacation of employees. This is a very important part of the new workflow system because it enables the software to automatically decide the responsibility of persons for open workflow tasks, even when the original responsible person is not available.

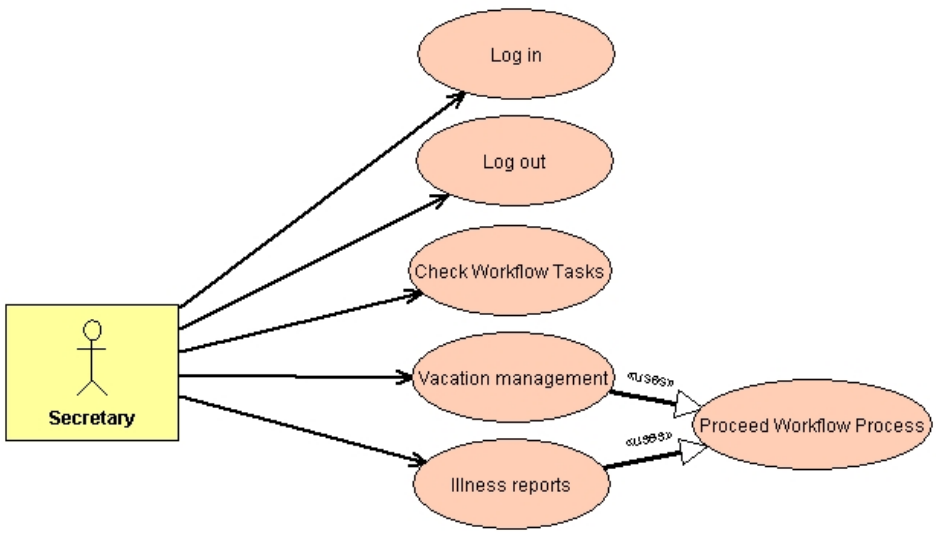

Figure D.1: Use Case Diagram: Competence Management

D.2 Information Delivery

Third party systems are able to subscribe to InfoWarrior, to get the latest news on demand.

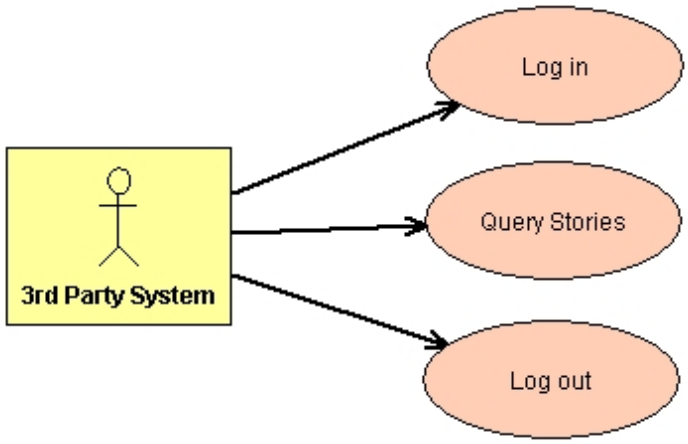

Figure D.2: Use Case Diagram: Information Delivery

D.3 Information Presentation

Why collect stories if nobody can read them? Presentation of stories can be found here.

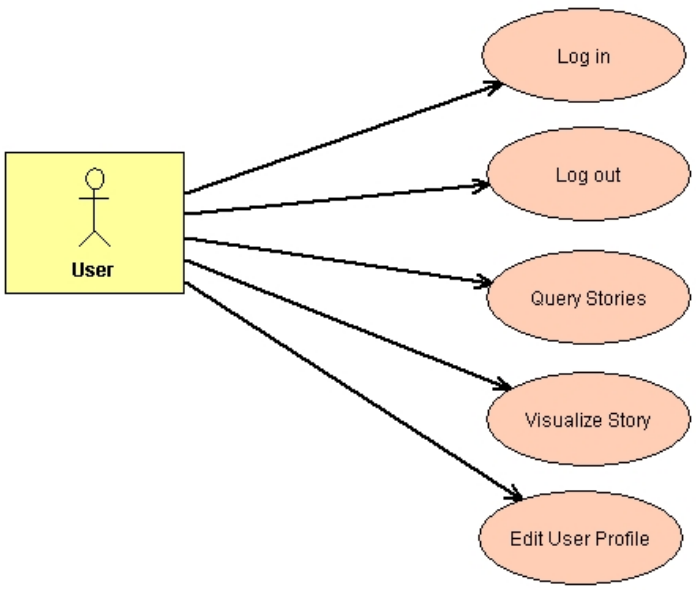

Figure D.3: Use Case Diagram: Information Presentation

D.4 Information Gathering

The most complex use case diagram, can be found here. It shows the creation of a story until its release or cancellation.

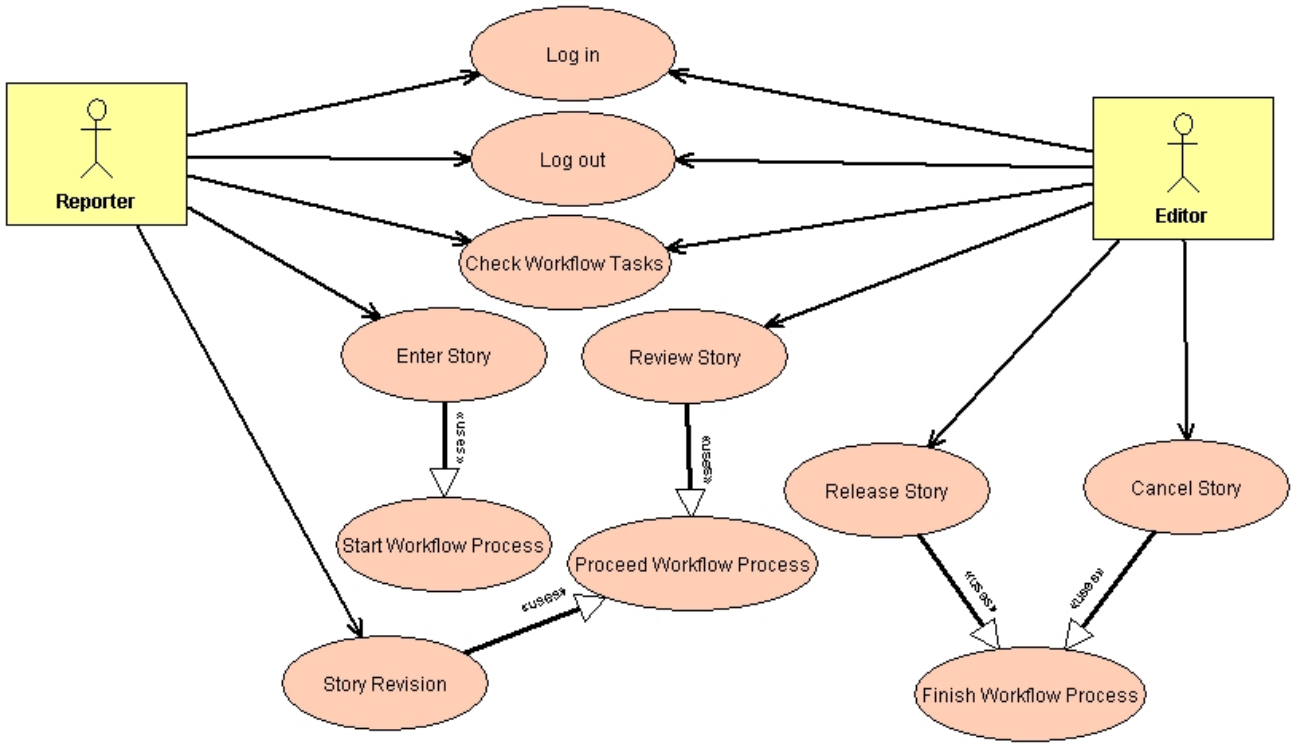

Figure D.4: Use Case Diagram: Information Gathering

Appendix E

Workflow System

The workflow system of InfoWarrior is a template driven system that is capable of managing different workflow processes at the same time, independent of the initiating user. The change of the responsibility of a task is also a very common used function.
The different workflow processes we have identified are modeled in UML by using activity diagrams, as show below:

E.1 Activity Diagram - Competence Management Workflow

The handling of illness reports and vacation planing of epmloyees is show in this activity diagram.

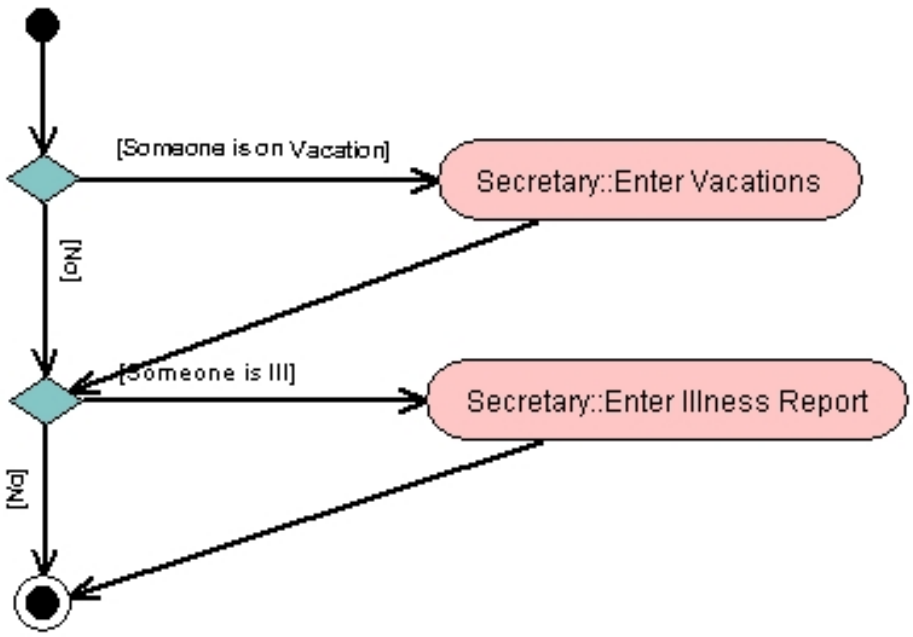

Figure E.1: Activity Diagram: Competence Management Workflow

E.2 Activity Diagram - Story Workflow

Workflow from the input of the story in the system, until its completetation and release, or due the lack of 1/Entropy its cancellation.

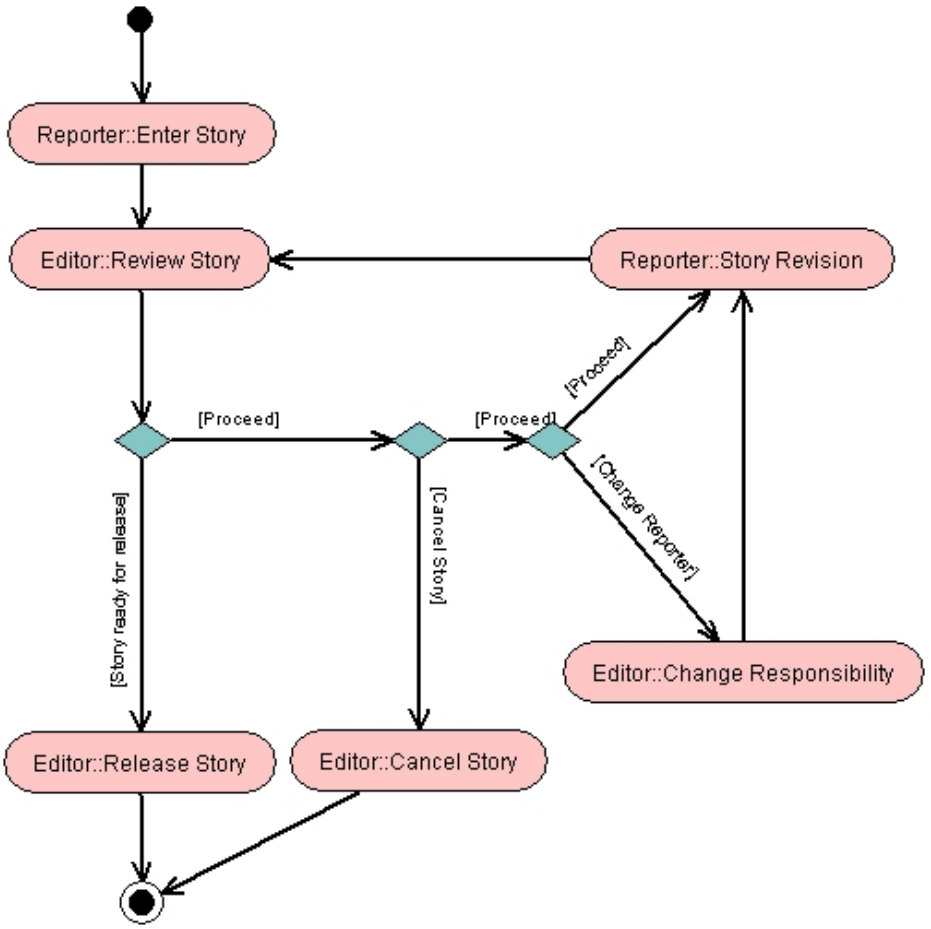

Figure E.2: Activity Diagram: Story Workflow

Appendix F

Essential User Interface Prototypes

F.1 Login

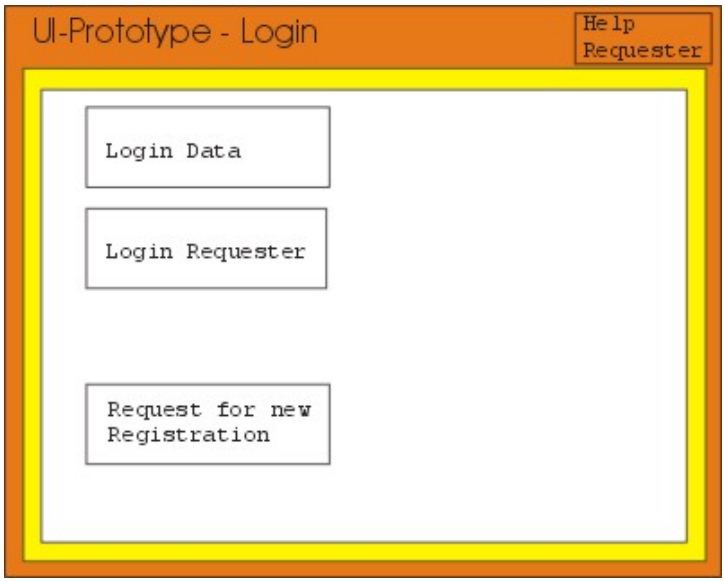

Figure F.1: Essential User Interface Prototype: Login

F.2 User Profile

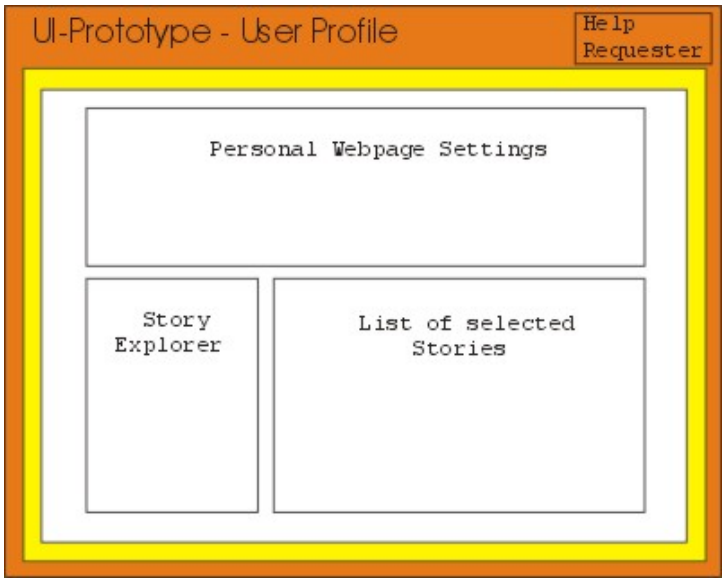

Figure F.2: Essential User Interface Prototype: User Profile

F.3 Check Workflow tasks

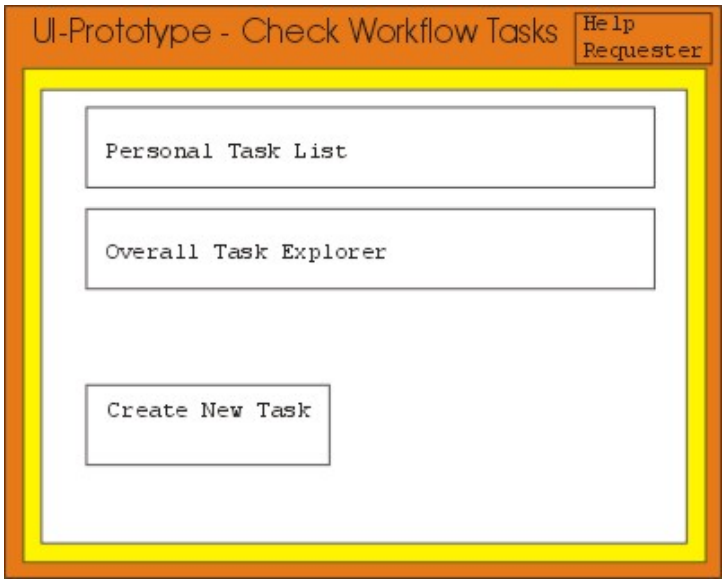

Figure F.3: Essential User Interface Prototype: Workflow task check

F.4 Story Visualization

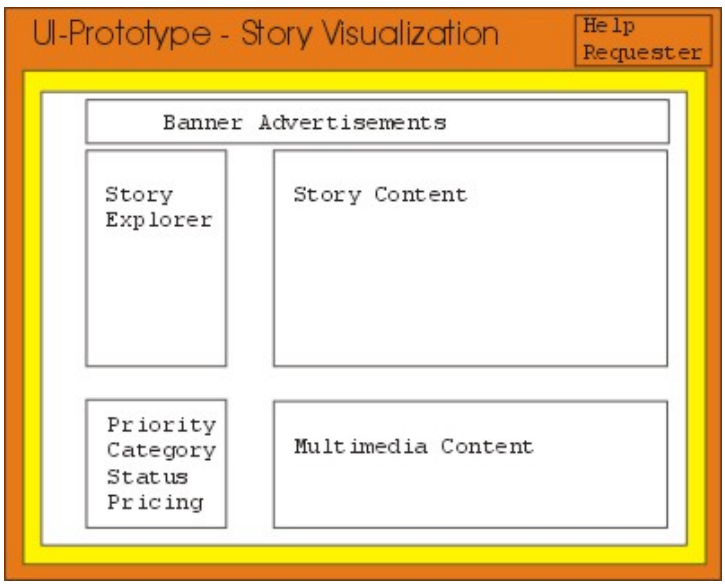

Figure F.4: Essential User Interface Prototype: Story Visualization

F.5 Story Editing

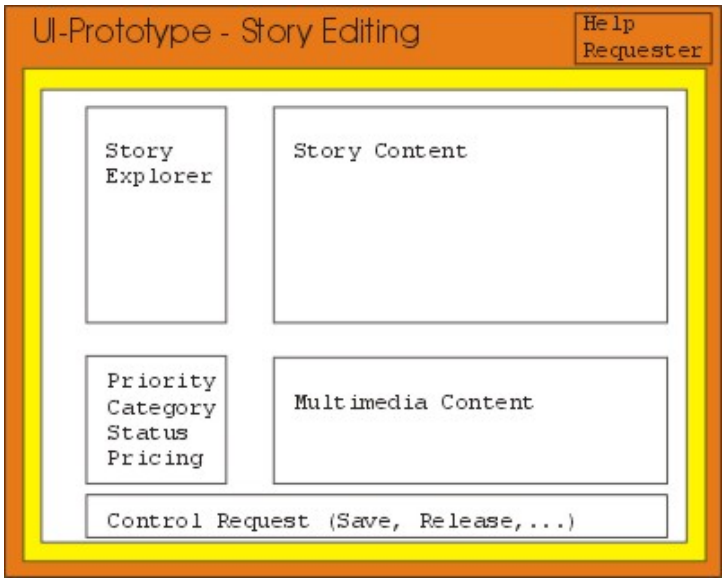

Figure F.5: Essential User Interface Prototype: Story Editing

F.6 Competence Management

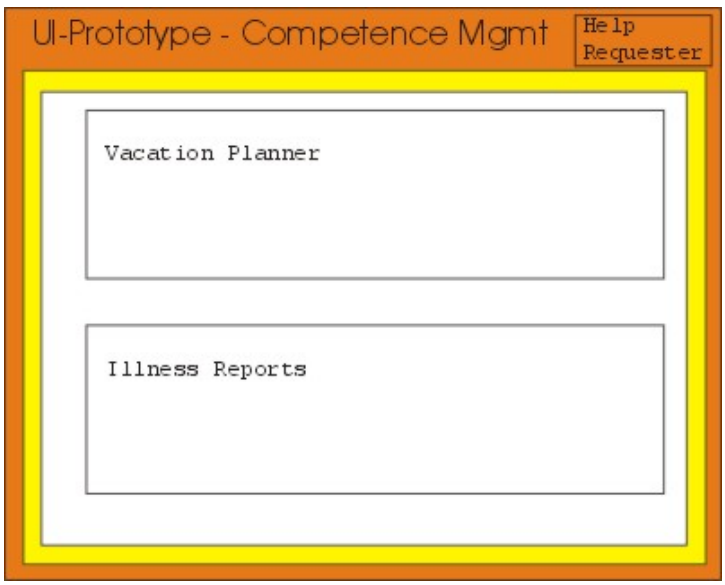

Figure F.6: Essential User Interface Prototype: Competence Management

Appendix G

Analysis Class Diagrams

G.1 WorkflowSubSystem

G.1.1 Description

Each workflow consists of a process which can have an arbitrary amount of tasks. This workflow also has a protocol, consisting of each tasks state which was running during the lifetime of the workflow. This is where the Memento design pattern takes place.

Each task has specific groups of users assigned which are responsible for the further attention handling, but only one user out of this groups will actual process the current task.

Every user owns exact one calendar, which holds the information for vacation,illness and if the user is currently available . This is needed for the workflow subsystem to automatically determine which users of the groups are available for further tasks.

G.1.2 Used Patterns

Memento Design Pattern The memento pattern is used to record and export the inner state of an object, without loose of its inner encapsulation, to obtain the ability to return to this state later on. Taken from [1][Pages 317-332].

G.1.3 Class Diagram

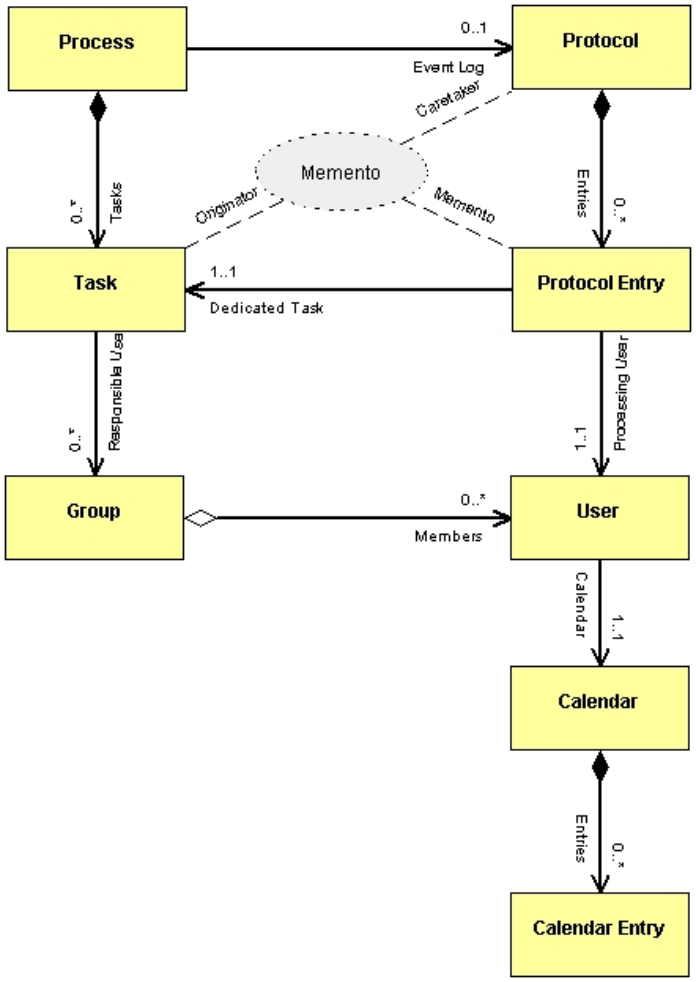

Figure G.1: Analysis Class Diagram: WorkflowSubSystem

G.2 WorkflowTasks

G.2.1 Description

Currently three different tasks exist in our system, and each of these tasks has a different amount of successors.

WorkTask Only one successor can occure in this specific type of a task. The actual work on a story is done in this type of task.

DecisionTask Here a decision takes place what the next worktask will be. Each decision task has two possible successors.

StartStopTask For starting and stopping a workflow.

G.2.2 Class Diagrams

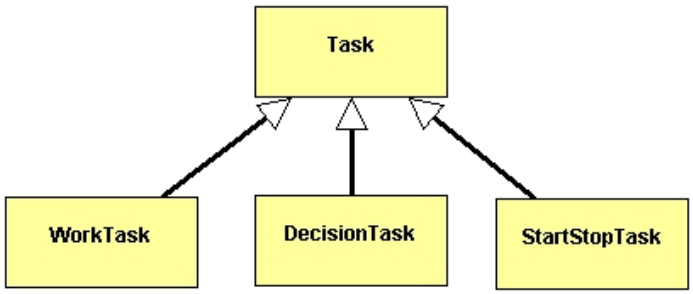

Figure G.2: Analysis Class Diagram: WorkflowTasks

G.3 RequestHandling

G.3.1 Description

A client causes many different requests to the system, these requests have to be processed by the Front Controller to get the users proper permission and to check the requests validity. This introduces an additional security level, to keep unauthorized accesses to the system away from the inner processing of such requests.

After a request is authorized to be processed the dispatcher, in our case the Action Mapping, creates the corresponding Action. An Action is realized by implementing the Command Pattern. The client invokes the execution of such an action, undoing the last action is automatically received by using this pattern.

Triggering actions causes the creation of views respective to the specific request. Such a view could be HTML pages for browsers, or XML data for third party systems. Which will be afterwards sent back to the client.

G.3.2 Used Patterns

FrontController The request handling mechanism must control and coordinate processing of each user across multiple requests. Such control mechanism my be managed in either a centralized or decentralized manner. Taken from [2][Pages 172-185].

Command Encapsulate a command as an object. To obtain the ability, to vary the parameters of clients with different requests, to put operations into a queue, to create a log and to undo operations. Taken from [1][Pages 245-256]

G.3.3 Class Diagrams

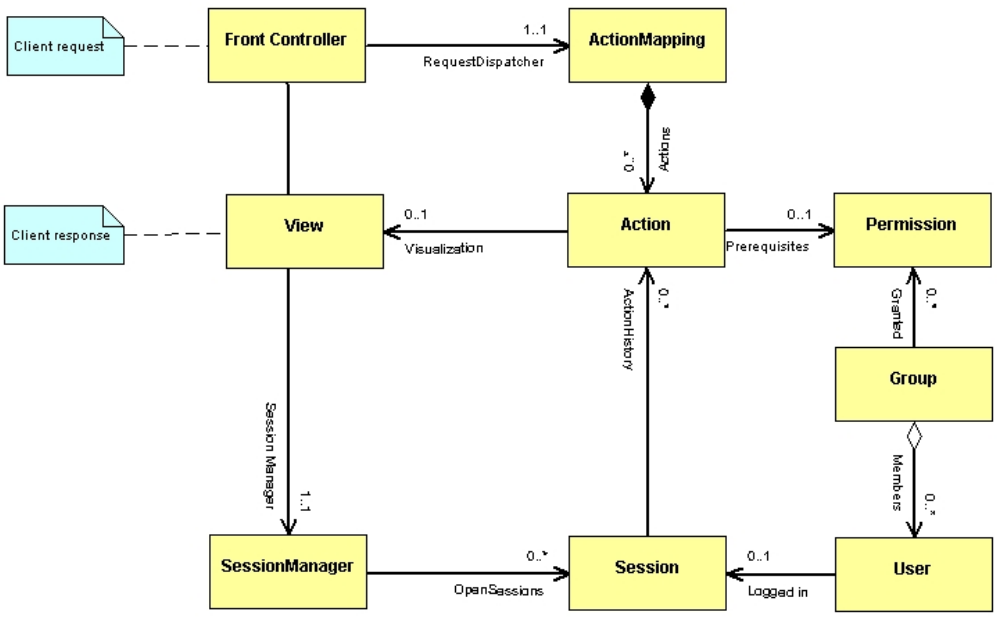

Figure G.3: Analysis Class Diagram: RequestHandling

G.4 Actions

G.4.1 Description

All so far known needed actions are listed below.

acLogin

acLogout

acCheckWorkFlowTasks

acQueryStories

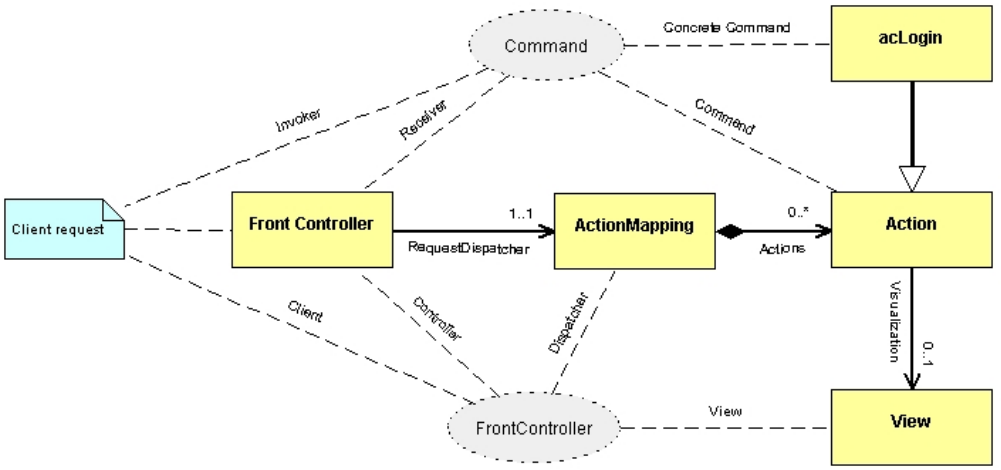

Figure G.4: Analysis Class Diagram: Patterns used in RequestHandling

acQueryStory

acSaveStory

acReleaseStory

acCancelStory

acGetUserSettings

acSetUserSettings

acSetCalendar

acGetCalender

G.4.2 Class Diagrams

G.5 System users

G.5.1 Description

Here all so far known different type of users can be found here.

Editor

3rd Party System

Secretary

Reporter

Viewing User

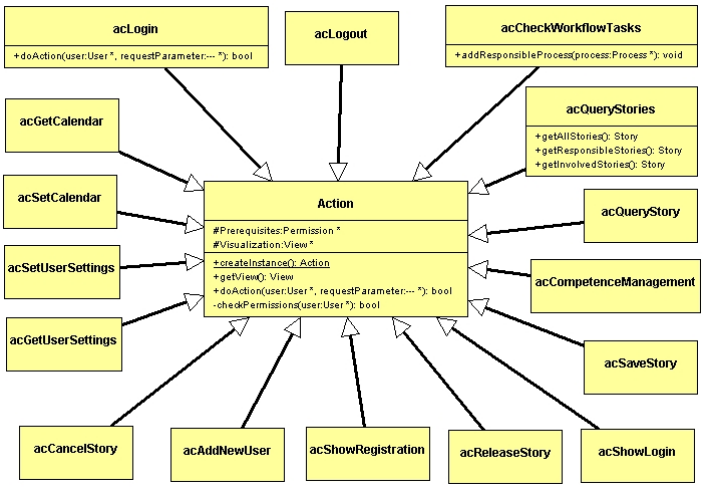

Figure G.5: Analysis Class Diagram: Actions

G.5.2 Class Diagrams

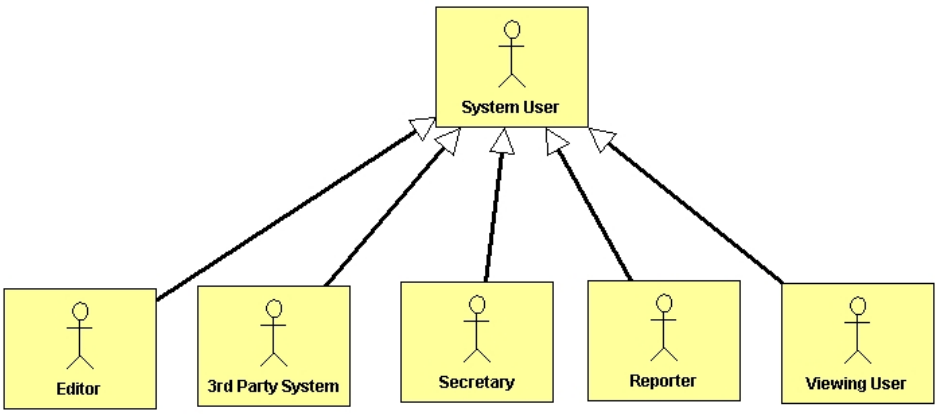

Figure G.6: Analysis Class Diagram: System Users overview

Appendix H

Sequence Diagrams

H.1 Introduction

A sequence diagram shows the program sequence of a method and considers the interaction with other objects. It is a form of visualizing the interaction model of an object oriented modeled target system. In doing so it shows the object relations and the chronology of method calls.

H.2 RequestHandling

This sequence diagram documents the basic technique for processing requests from any possible client.
Special design decisions:

- For every incoming request, if not yet existent, a new session will be created.
- The doAction method of the action class decides whether to delete the session or not.
- The users permission on the requested action is checked by the action itself.
- The action class has to have an internal state in order to know which view will be returned by the getView method. This method has to be called after the doAction method.

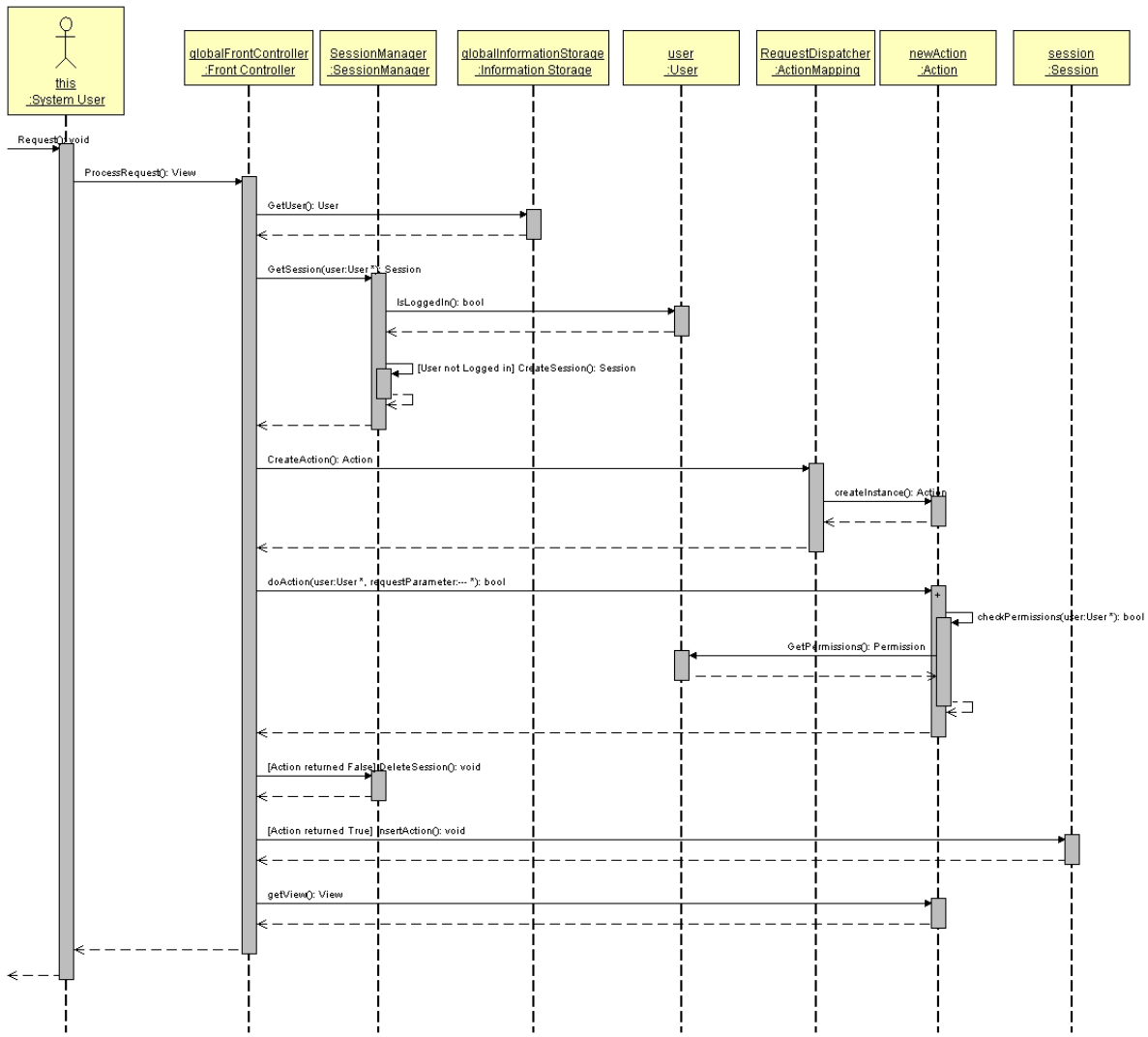

Figure H.1: Sequence Diagram: Request Handling

H.3 WorkflowSubSystem

H.3.1 Check Workflow Process

The sequence diagram for checking the workflow process gives an overview how the responsibility for a workflow task is determined.

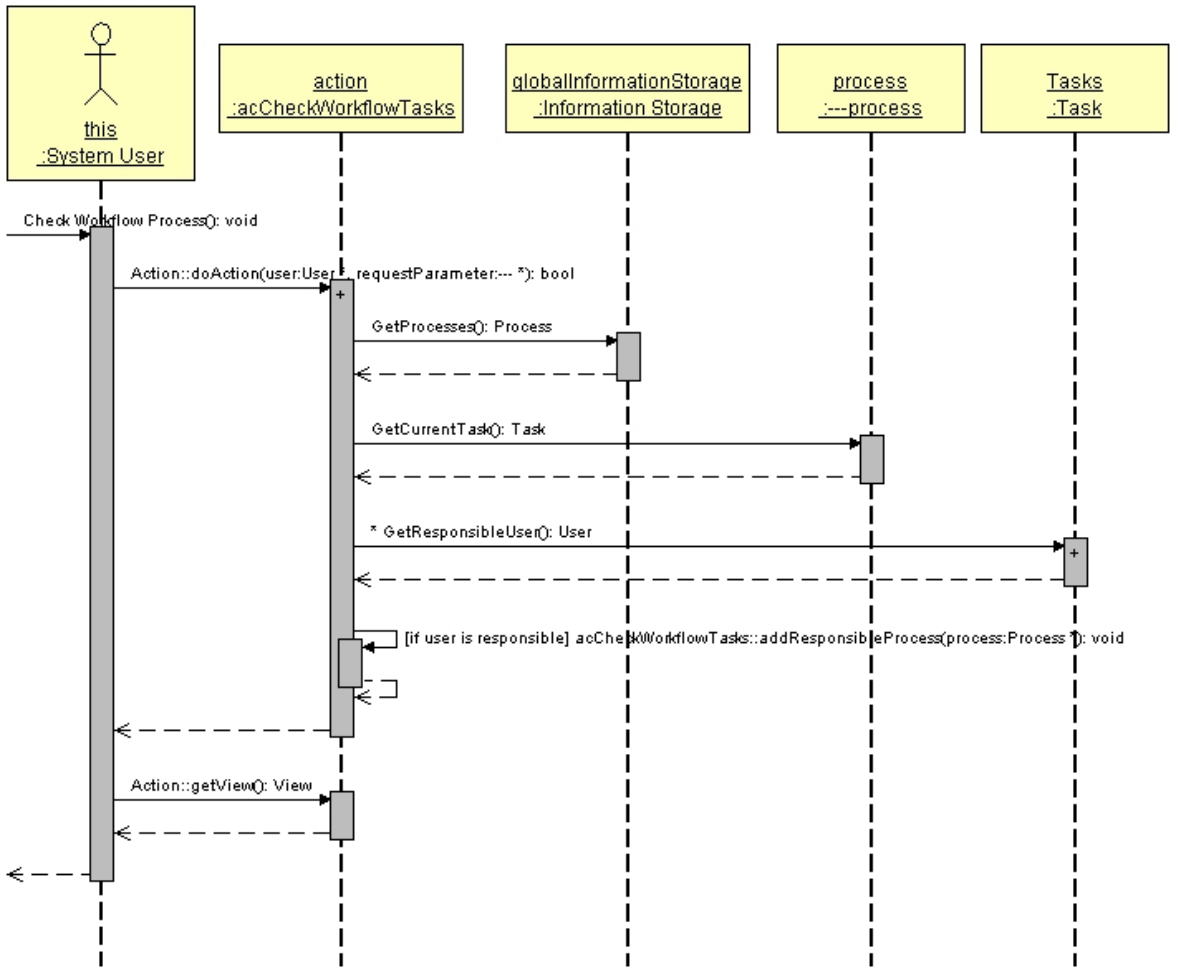

Figure H.2: Sequence Diagram: Check Workflow Process

H.3.2 Query Stories

This really complex diagram illustrates all possibilities of querying stories saved in the system. There are four ways for querying stories:

1. Query all stories
2. Query all stories with constraints (e.g. only sport stories)
3. Query stories the user is responsible for

4. Query stories the user is involved with

All these cases are modeled in the sequence diagram, but case one and two are almost the same, the only differ in the query parameter given to the GetStories method of the information storage.

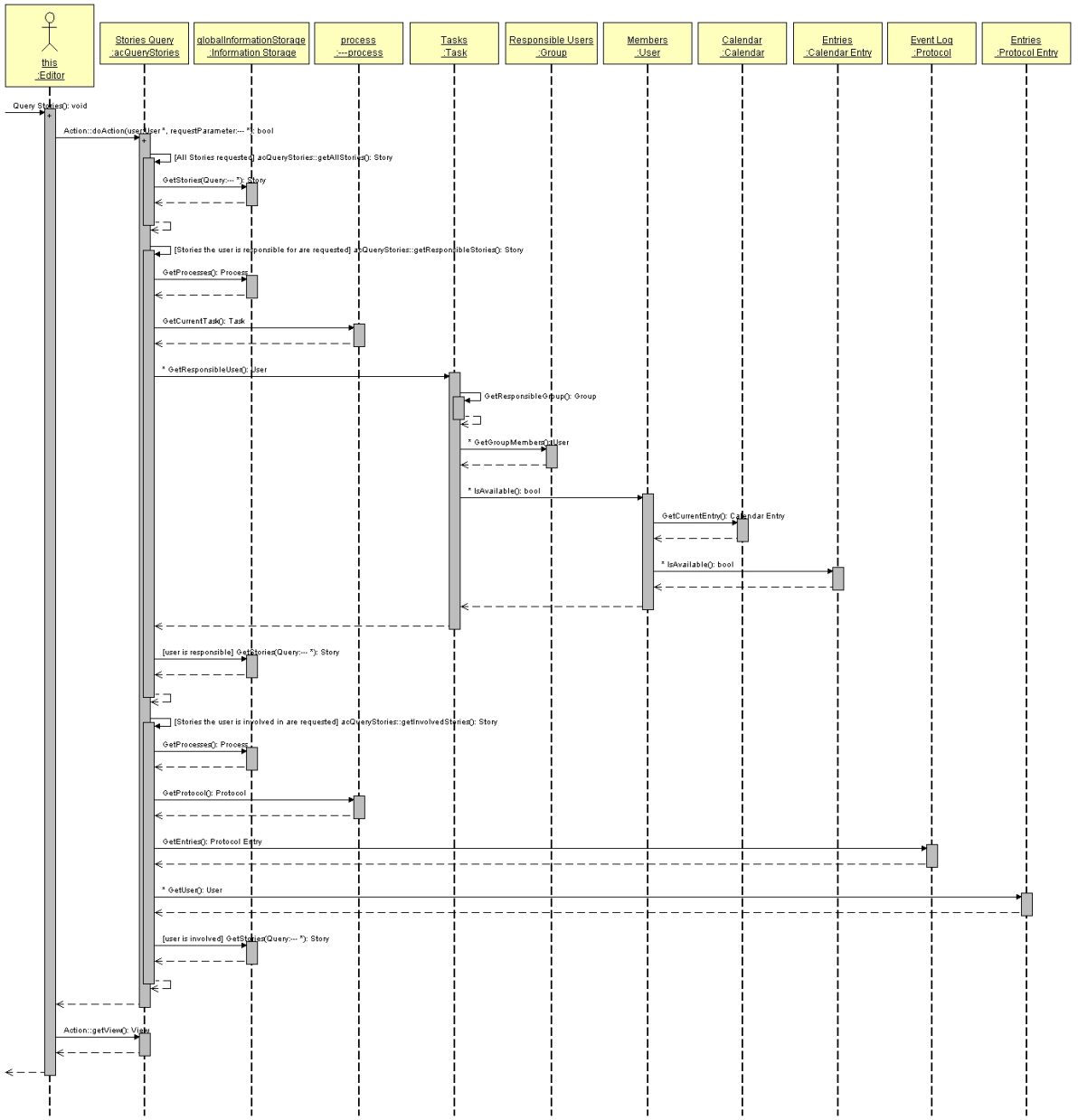

Figure H.3: Sequence Diagram: Query Stories

H.3.3 Enter Vacations

This sequence diagram can be seen as a higher level model because it illustrates the interaction of a sequence of requested actions.

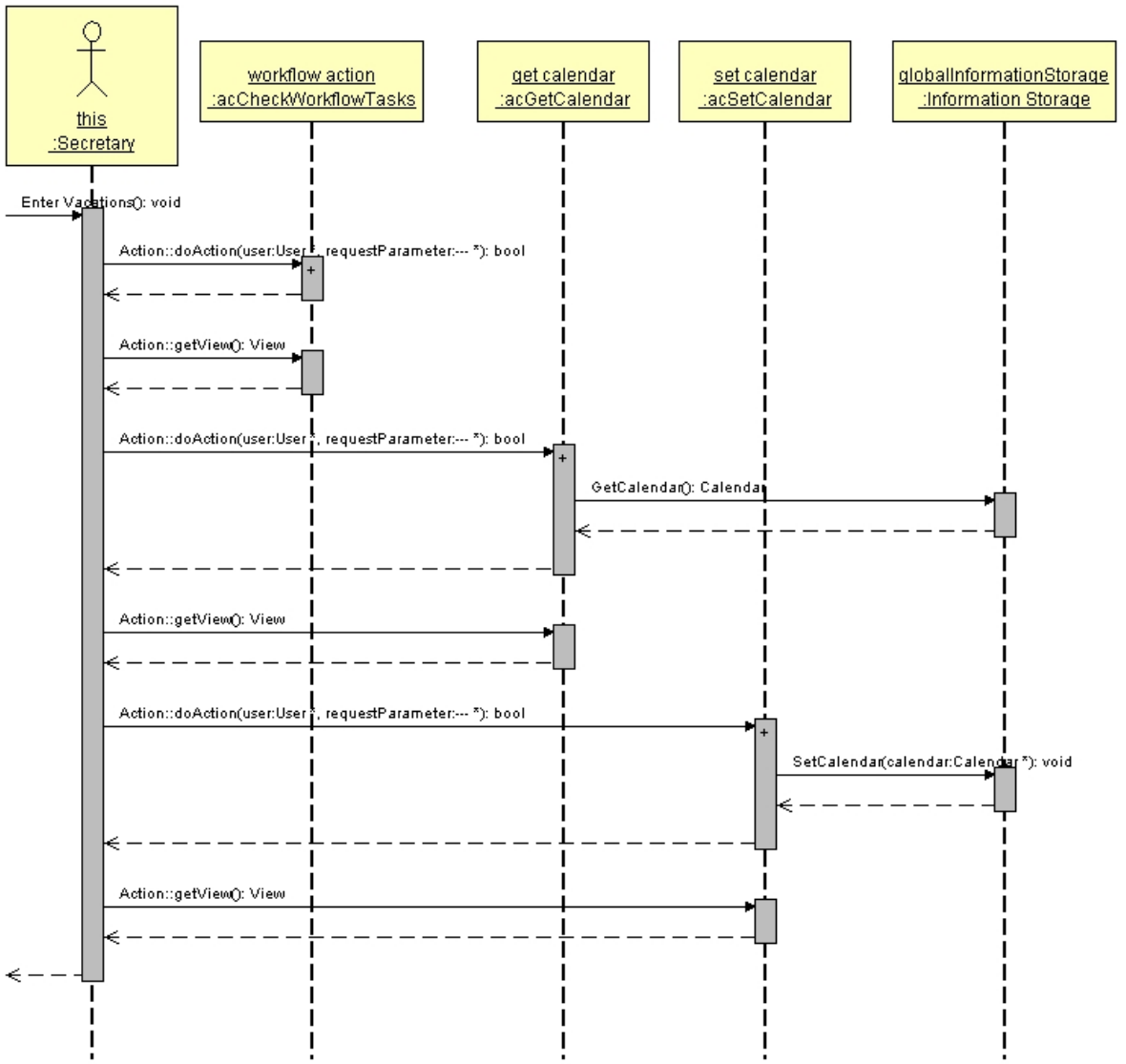

Figure H.4: Sequence Diagram: Enter Vacations

Appendix I

Activity Diagrams

I.1 Introduction

The activity diagram is an alternative to a state diagram. It shows the interference of states and their relations.

Main goal of an activity diagram is the description of a methods algorithm, which in this context is called an activity.

I.2 Login action

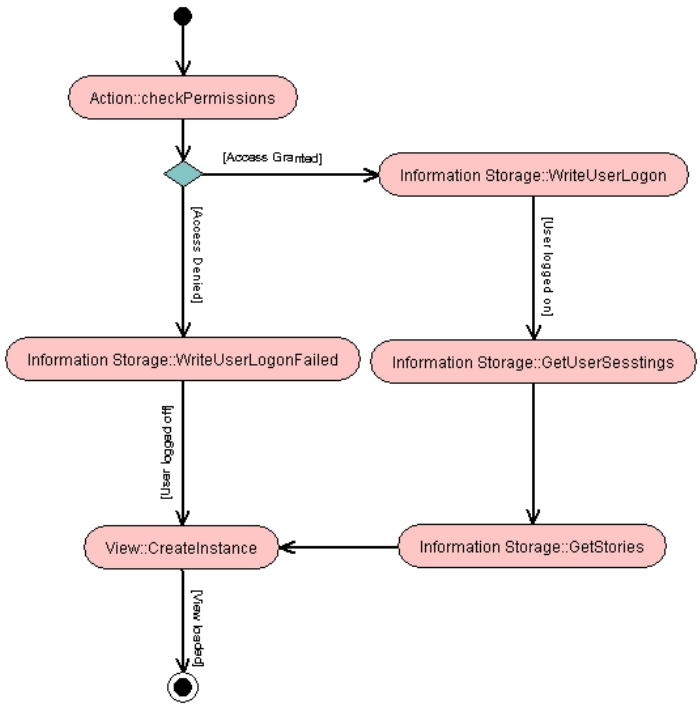

Figure I.1: Activity Diagram: Login Action

I.3 Save Story action

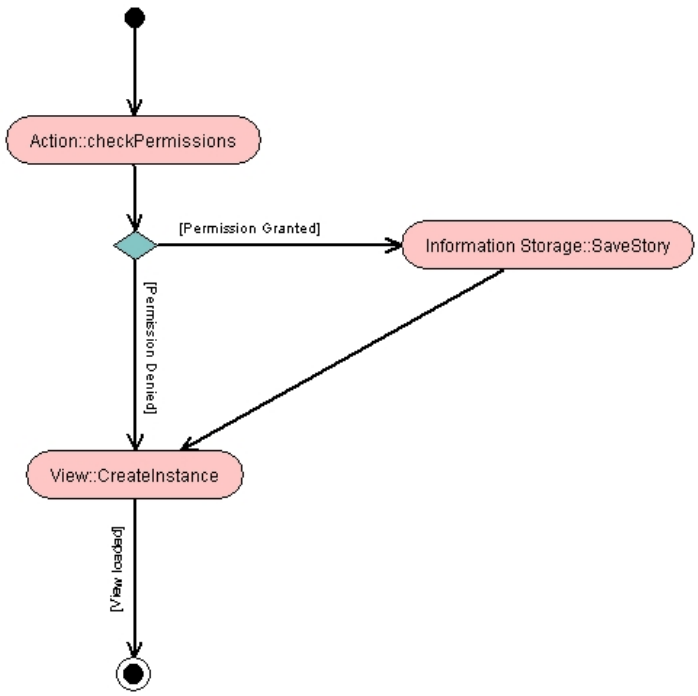

Figure I.2: Activity Diagram: Save Story Action

Appendix J

User Interface Prototypes

J.1 User Interface Flow Diagram

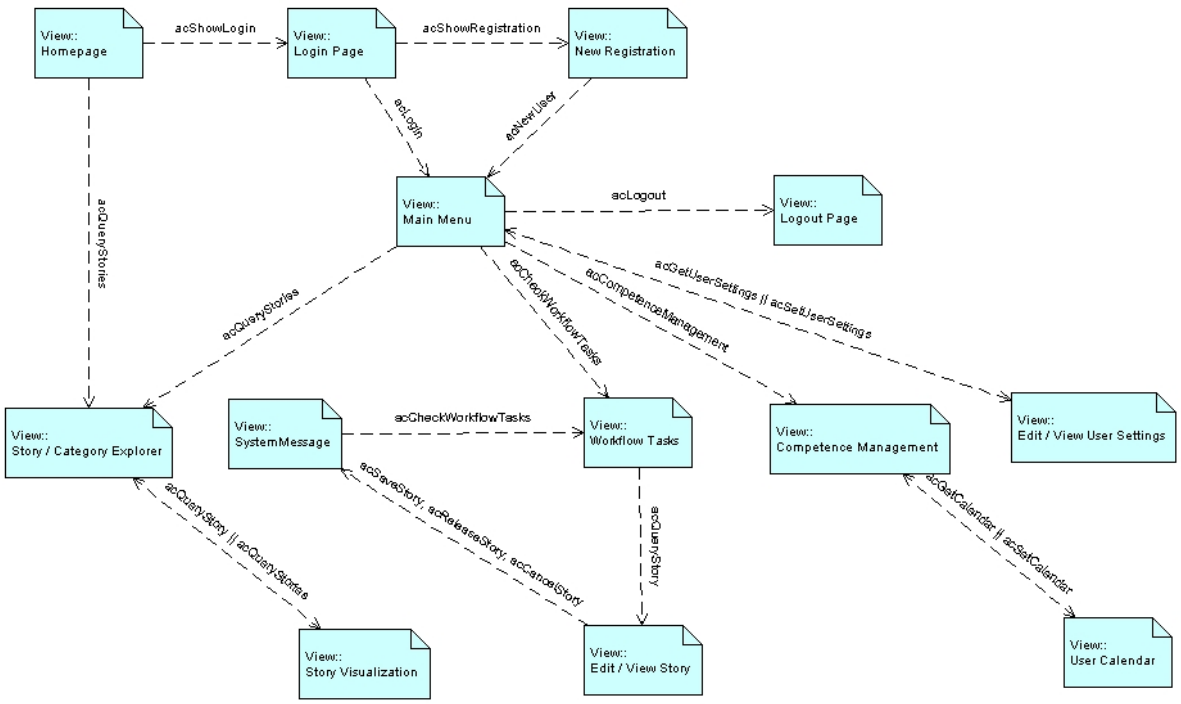

Figure J.1: User Interface Flow Diagram

J.2 HTML User Interface

J.2.1 Login Page

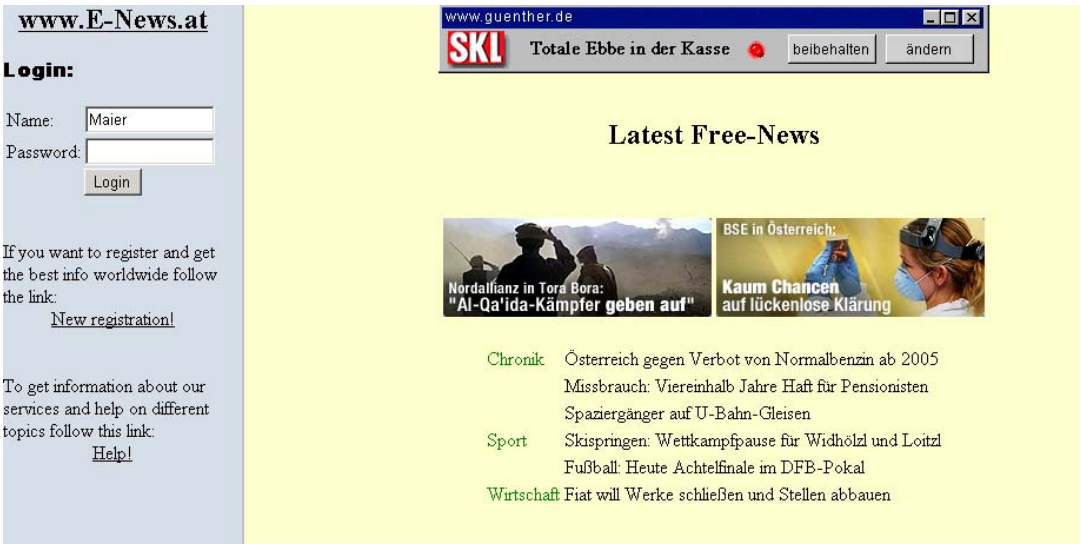

Figure J.2: HTML user interface: Login Page

J.2.2 New Registration

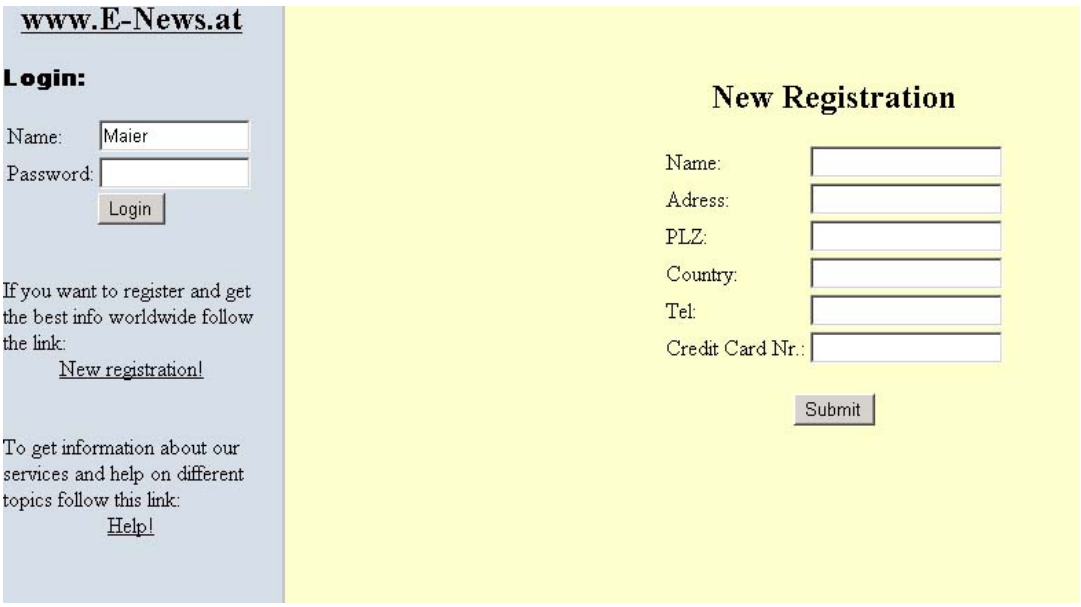

Figure J.3: HTML user interface: New Registration

J.2.3 Main Menu

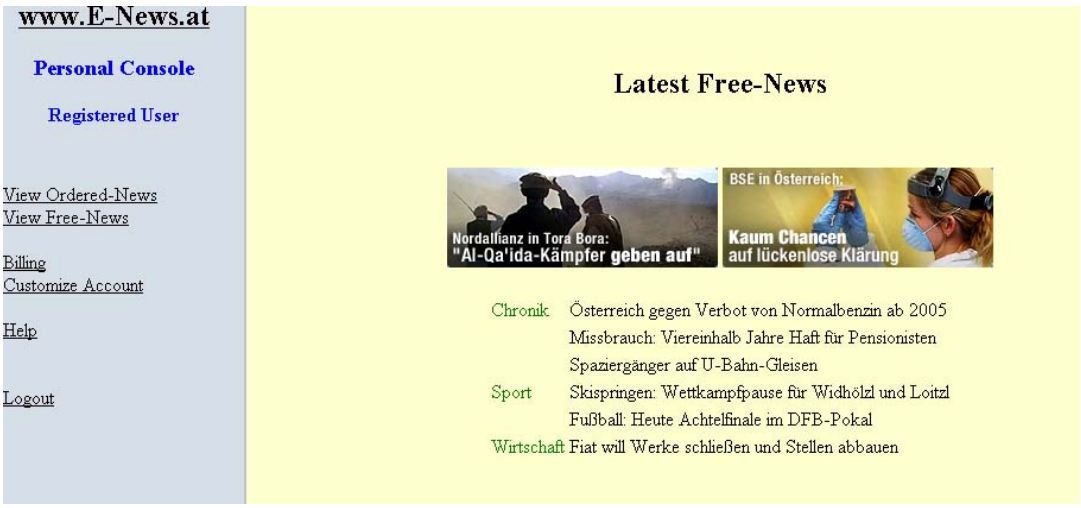

Figure J.4: HTML user interface: Main Menu

J.2.4 Check WorkFlow Tasks

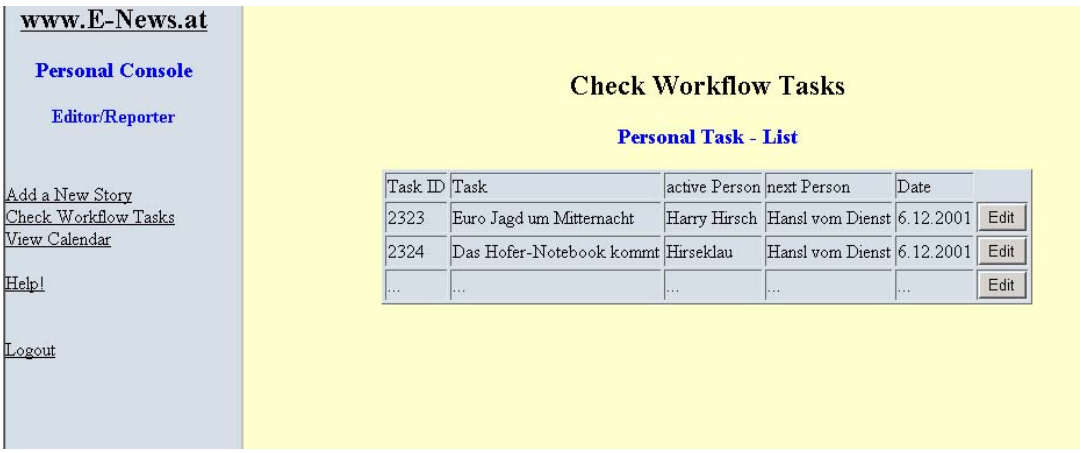

Figure J.5: HTML user interface: Check WorkFlow Tasks

J.2.5 Edit/View Stories

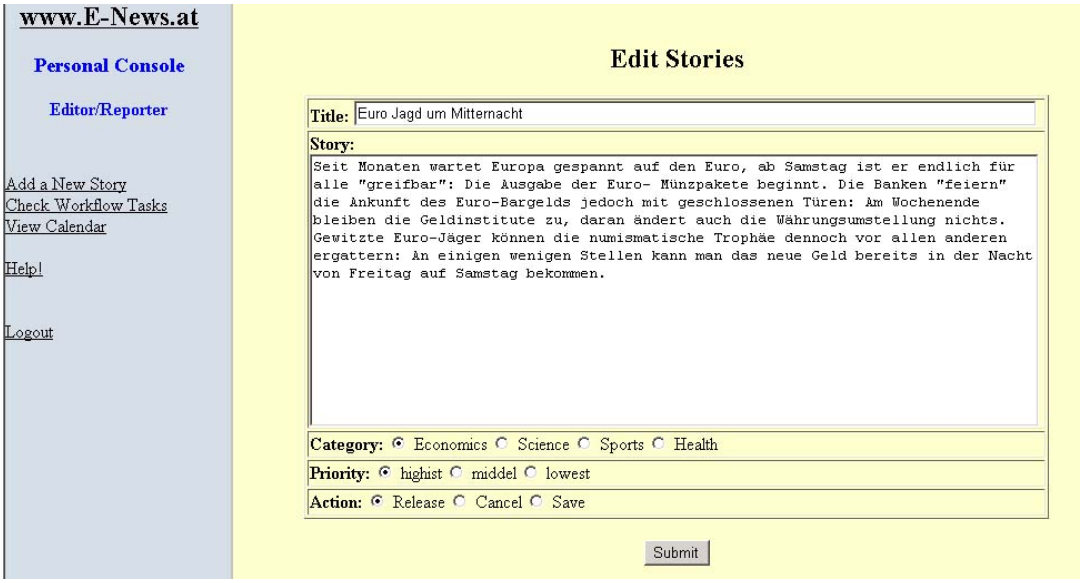

Figure J.6: HTML user interface: Edit/View Stories

J.2.6 Calendar

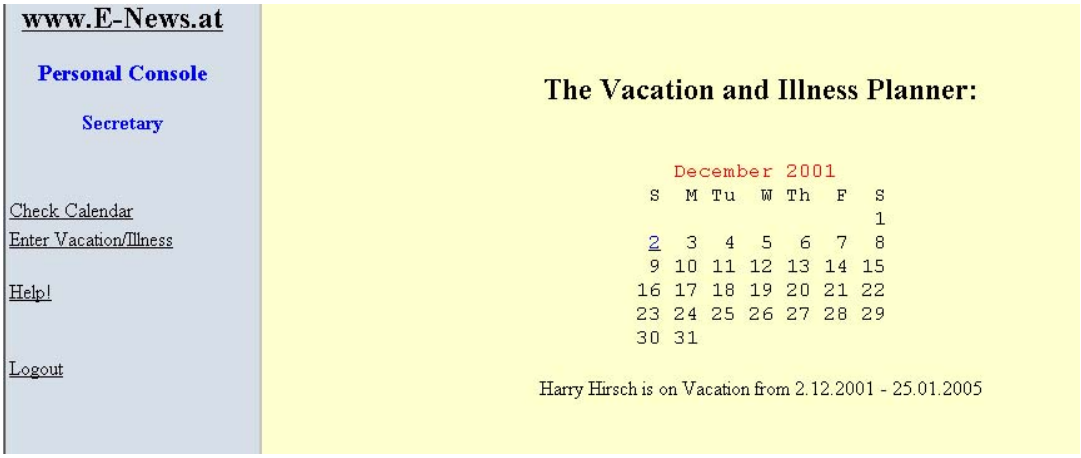

Figure J.7: HTML user interface: Calendar

J.2.7 Competence Management

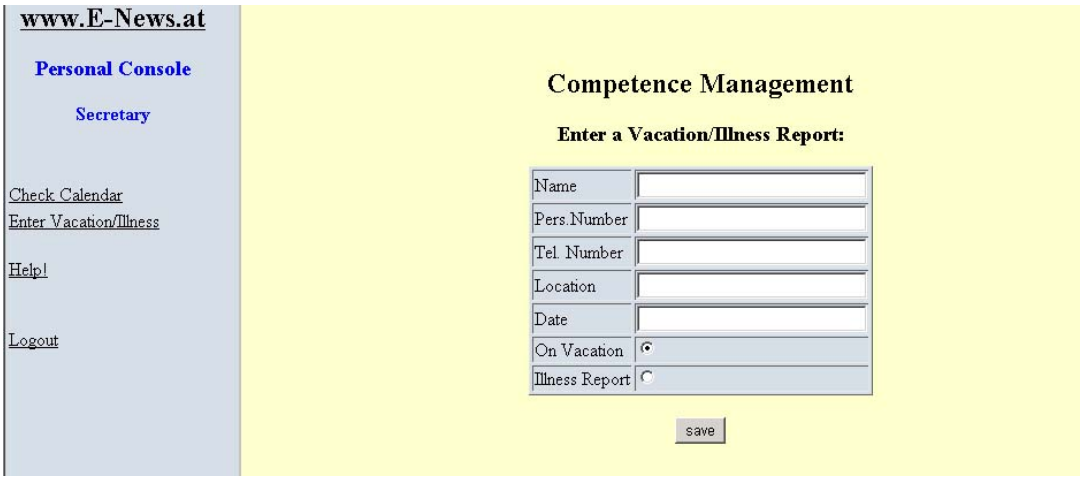

Figure J.8: HTML user interface: Competence Management

J.2.8 Visualization (Ordered News)

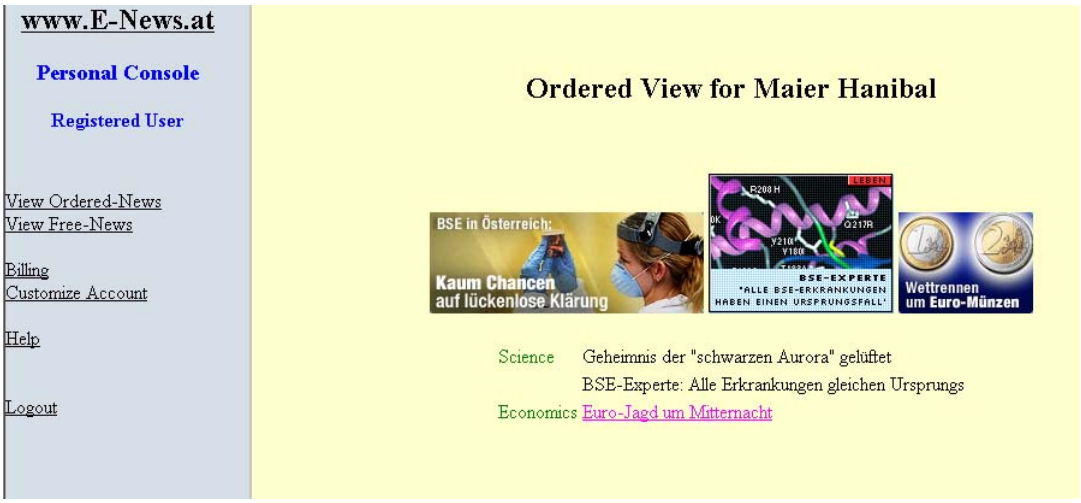

Figure J.9: HTML user interface: Visualization

J.2.9 User Customization

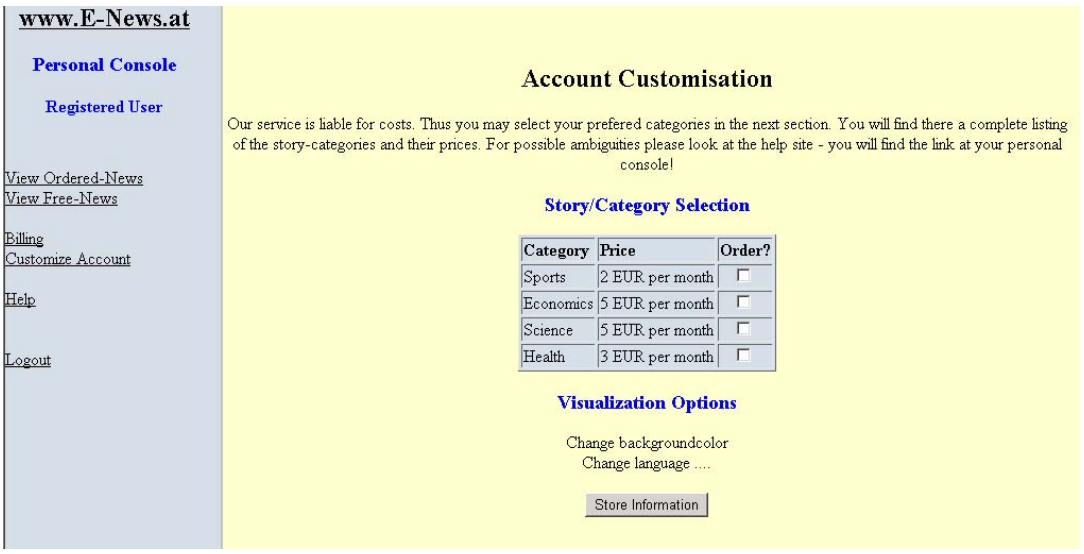

Figure J.10: HTML user interface: User Customization

J.2.10 Billing Statistics

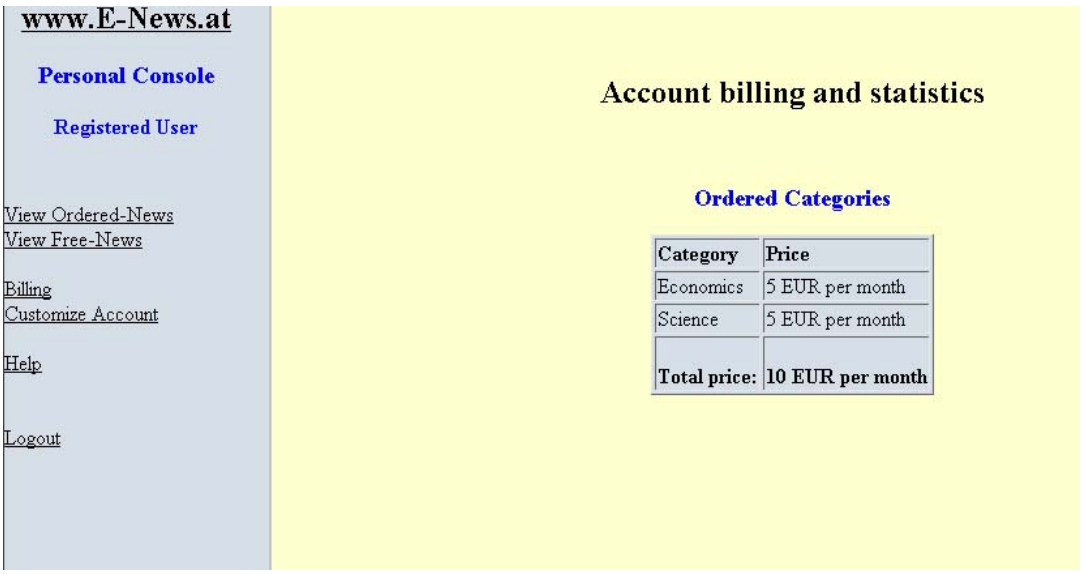

Figure J.11: HTML user interface: Billing Statistics

J.3 WML User Interface

J.3.1 Default Home Page

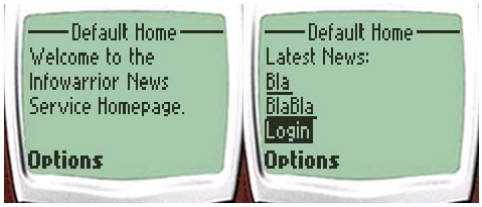

Figure J.12: WML user interface: Default home page

J.3.2 Login Screen

Figure J.13: WML user interface: login screen

J.3.3 New Registration

Figure J.14: WML user interface: request new registration

J.3.4 Registered User

Figure J.15: WML user interface: registered user logged in

J.3.5 Reporter or Editor

Figure J.16: WML user interface: Reporter or Editor logged in

J.3.6 Adding a new Story

Figure J.17: WML user interface: enter a new story

Bibliography

[1] Erich Gamma; Richard Helm; Ralph Johnson; John Vlissides. *Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, 1st edition, 1996. ISBN 3-89319-950-0.

[2] Deepak Alur; John Crupi; Dan Malks. *Core J2EE Patterns - Best Practices and Design Strategies*. Sun Microsystems, Inc., 1st edition, 2001. ISBN 0-13-064884-1.