University of Salzburg
Institute of Computer Science

PS Software Engineering Part I

InfoWarrior

Online Information System for News Agencies

Christian Zödl

Date: December 19, 2001
Version: 1.0

Contents

List of Figures

List of Tables

Chapter 1

The CRC session

This chapter describes the fundamentals till the finished CRC session.

1.1 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.

1.2 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.

1.3 Scenarios

1.3.1 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.

1.3.2 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.

1.4 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

1.5 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

1.6 Goals

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

1.7 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?

1.8 Session Protocol for October 31, 2001

Present persons:

Dr. Roland Schwaiger (customer representative)

Thorsten Engl (project leader, vendor representative)

Christian Zödl (software architect)

Erich Eichberger (chief of protocol)

1.9 Session 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.

1.10 Personal Protocol

In this week we worked out the first CRC session. The goal for the CRC session was to get more information from the customer. So it should be possible, that there are no misunderstandings between customer and contractor.

Chapter 2

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.

2.1 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.

2.2 Essential Use Case Modeling

2.2.1 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.

2.3 Essential Use Cases for InfoWarrior

2.3.1 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.

2.3.2 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.

2.3.3 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.

2.3.4 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.

2.3.5 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.

2.3.6 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.

2.3.7 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.

2.4 Session Protocol for November 7, 2001

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)

2.5 Session 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.

2.6 Personal Protocol

At the meeting I was chief of the protocol, and so I documented some important information, what we should do for next week and some modification in our CRC system. This modifications are that the mailing subsystem would be replaced by a workflow subsystem, and the responsibility of a few classes were not clear, so they should be split up. Also we worked out the use cases. They can be seen above in section 2.3.

Chapter 3

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.

3.1 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.

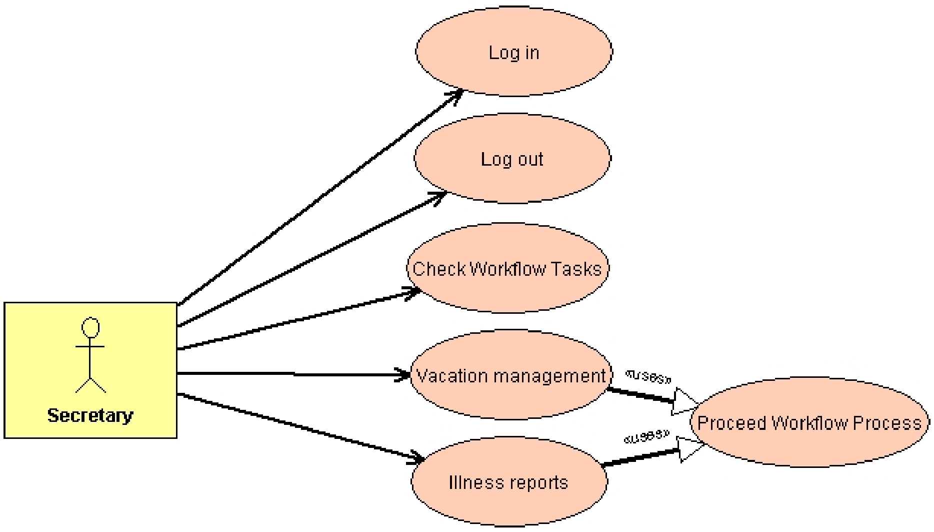

Figure 3.1: Use Case Diagram: Competence Management

3.2 Information Delivery

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

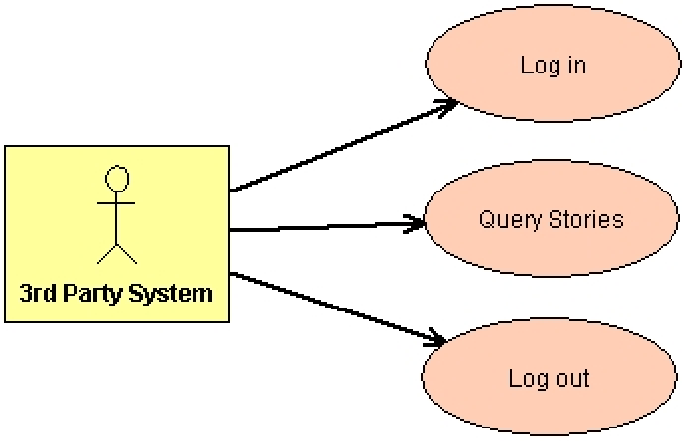

Figure 3.2: Use Case Diagram: Information Delivery

3.3 Information Presentation

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

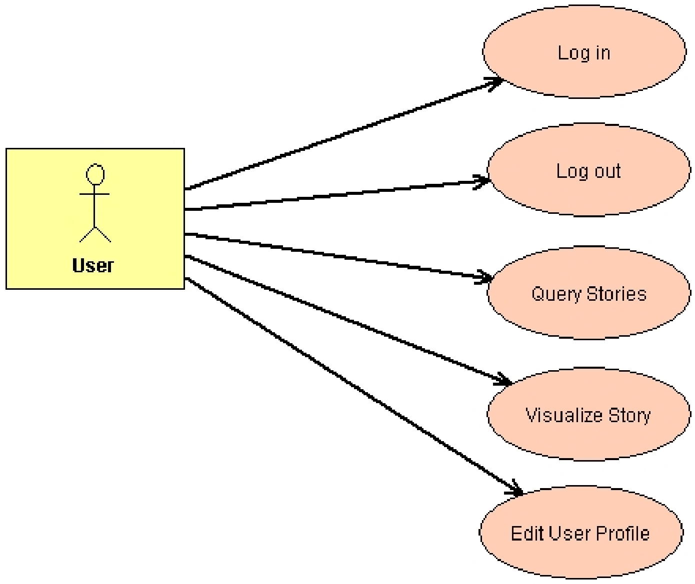

Figure 3.3: Use Case Diagram: Information Presentation

3.4 Information Gathering

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

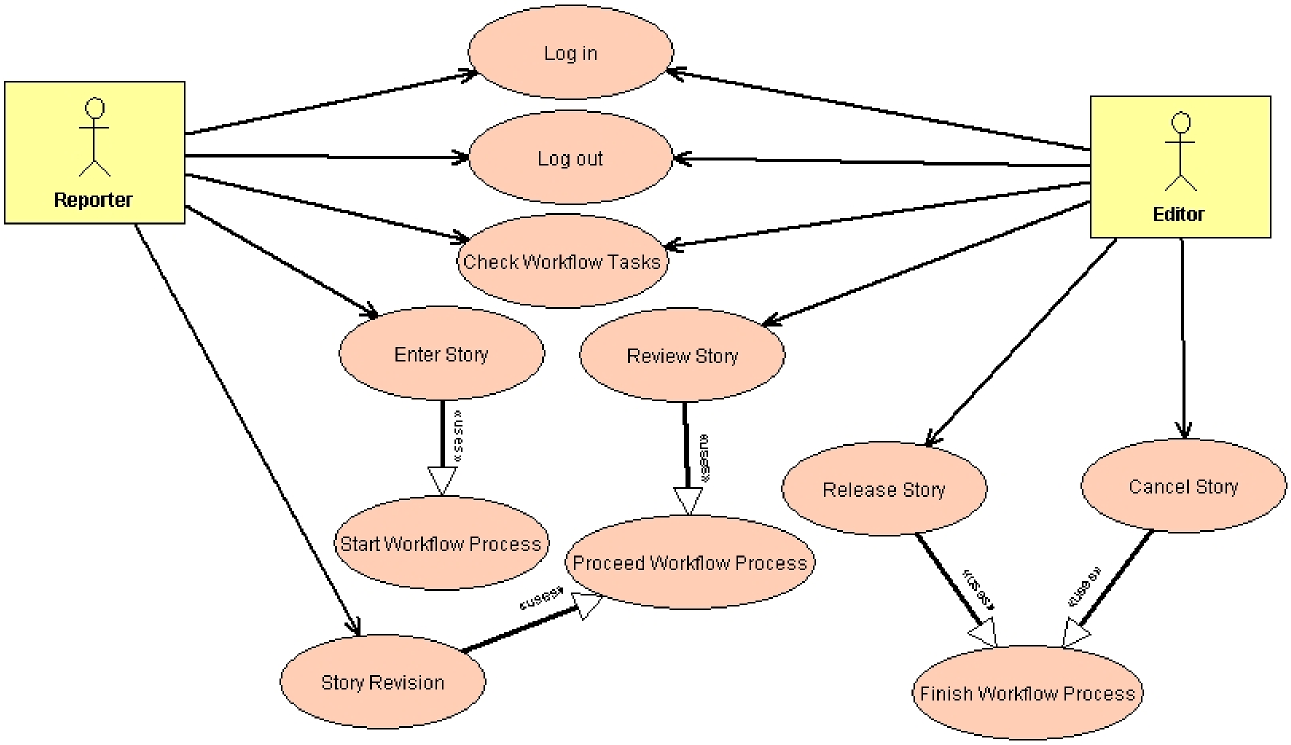

Figure 3.4: Use Case Diagram: Information Gathering

Chapter 4

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:

4.1 Activity Diagram - Competence Management Workflow

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

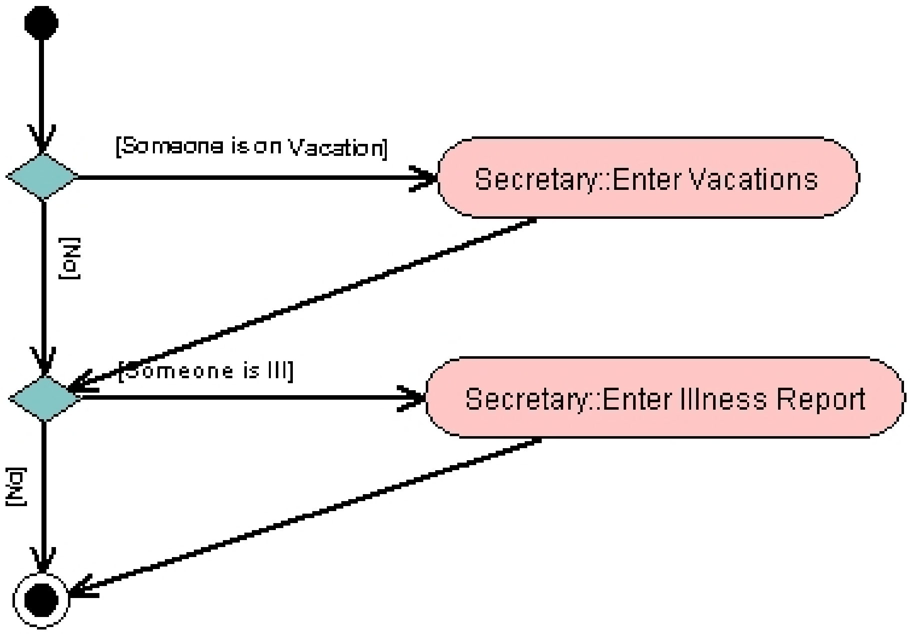

Figure 4.1: Activity Diagram: Competence Management Workflow

4.2 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.

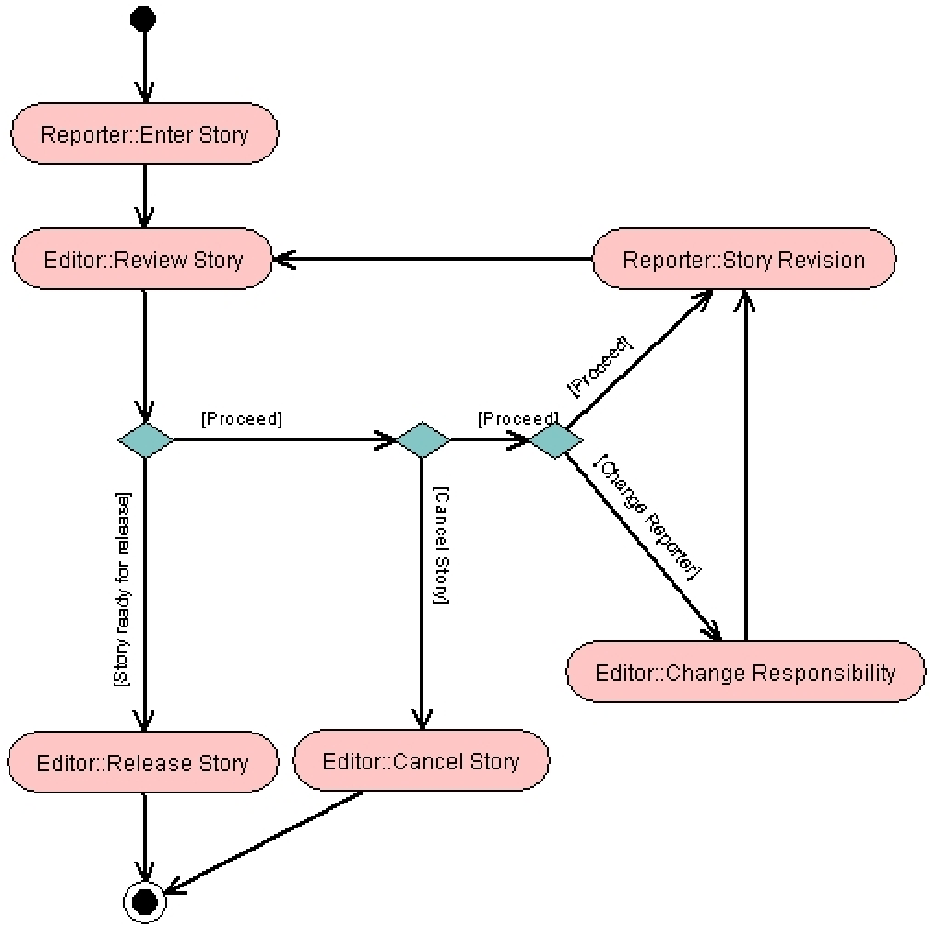

Figure 4.2: Activity Diagram: Story Workflow

Chapter 5

Essential User Interface Prototypes

5.1 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.

5.2 Login

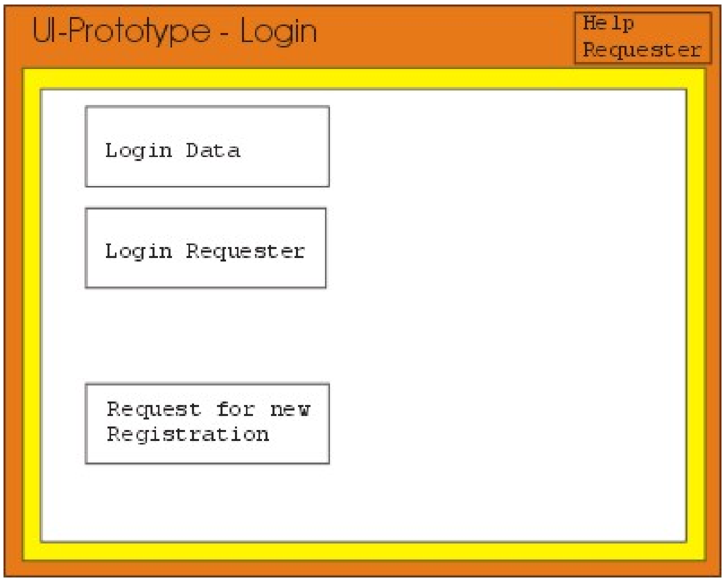

Figure 5.1: Essential User Interface Prototype: Login

5.3 User Profile

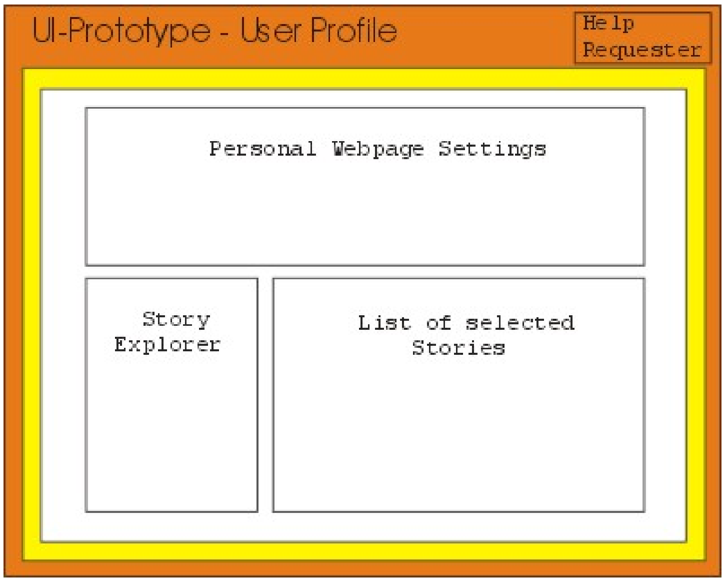

Figure 5.2: Essential User Interface Prototype: User Profile

5.4 Check Workflow tasks

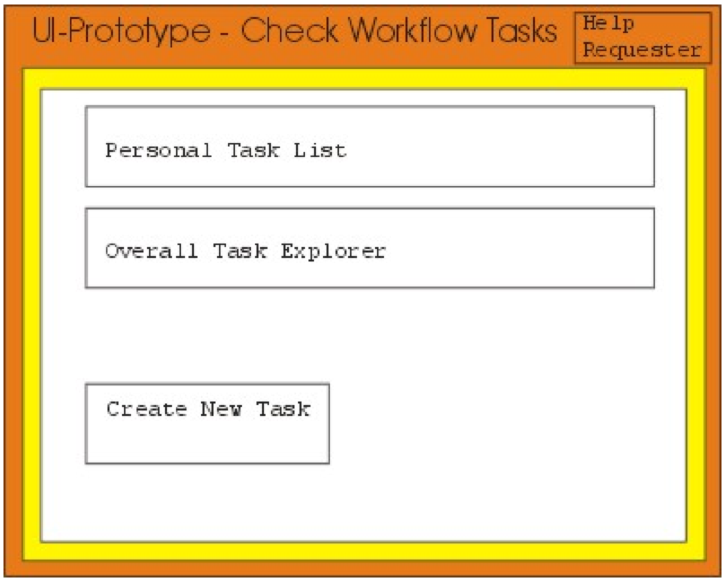

Figure 5.3: Essential User Interface Prototype: Workflow task check

5.5 Story Visualization

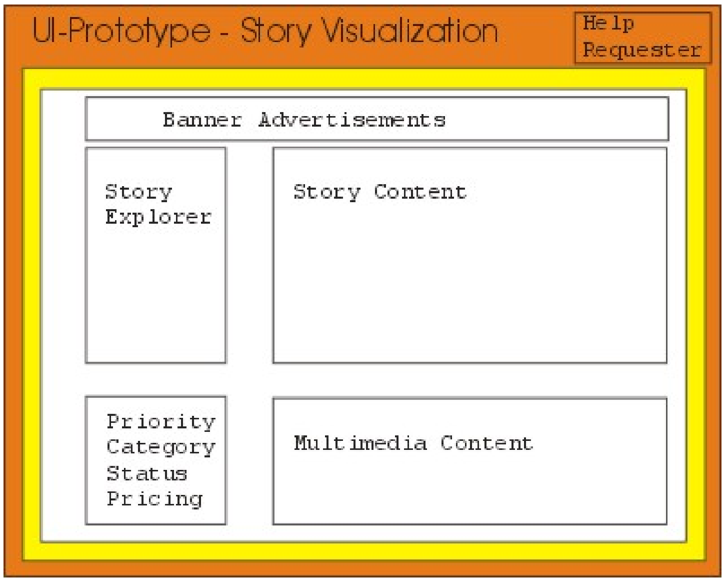

Figure 5.4: Essential User Interface Prototype: Story Visualization

5.6 Story Editing

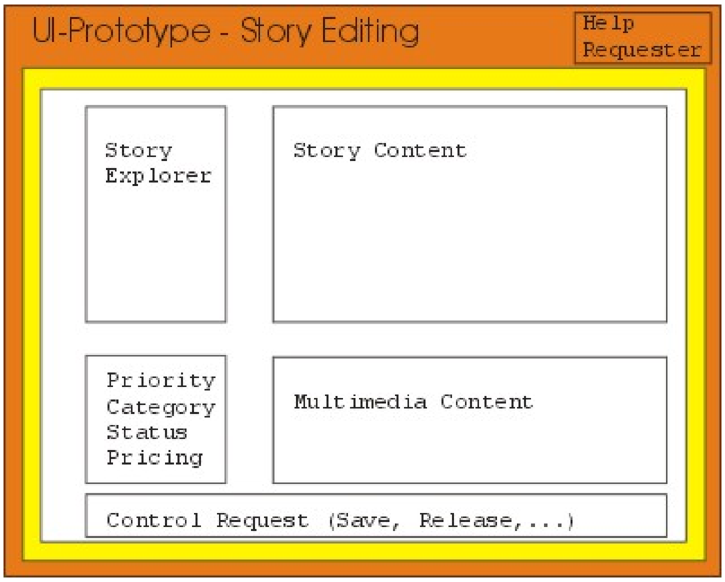

Figure 5.5: Essential User Interface Prototype: Story Editing

5.7 Competence Management

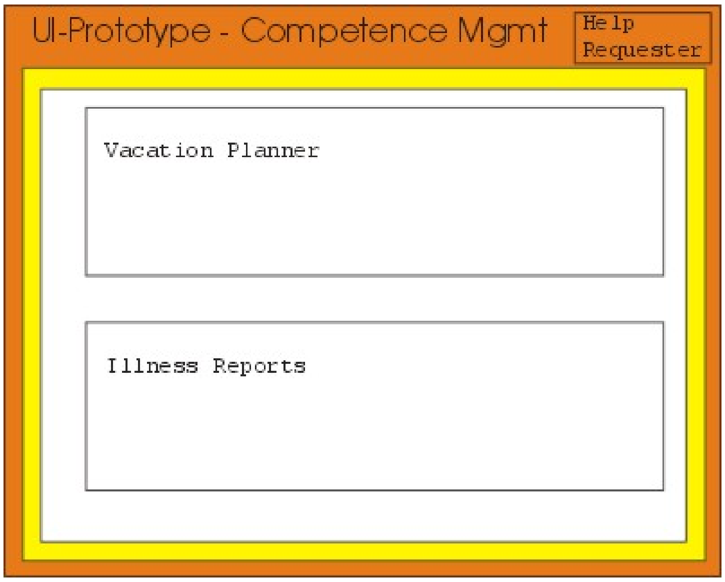

Figure 5.6: Essential User Interface Prototype: Competence Management

5.8 Personal Protocol

In this week we made the use case diagrams and the Essential User Interface Prototype. Also we thought about the workflow problem and worked out some activity diagrams for the workflow process.

The Essential User Interface Prototype gives the customer information how the user interface will look like. It's only a rough copy, but later it will be the base for further user interface prototypes. At the meeting the customer had the possibility to innovate the user interface and announce requests for changes.

Chapter 6

Analysis Class Diagrams

6.1 WorkflowSubSystem

6.1.1 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.

6.1.2 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].

6.1.3 Class Diagram

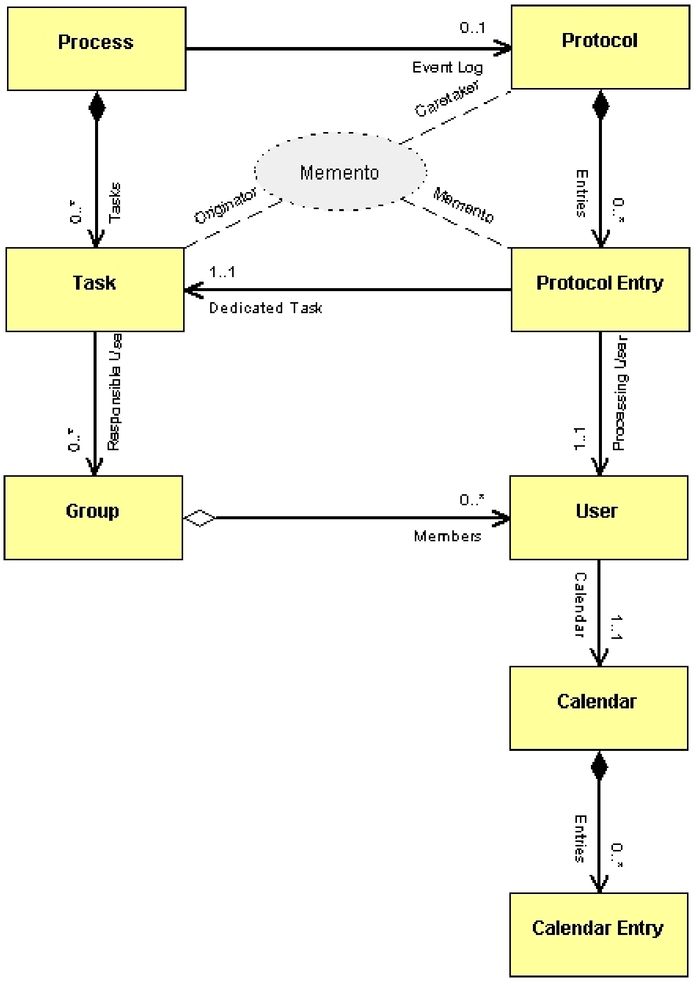

Figure 6.1: Analysis Class Diagram: WorkflowSubSystem

6.2 WorkflowTasks

6.2.1 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.

6.2.2 Class Diagrams

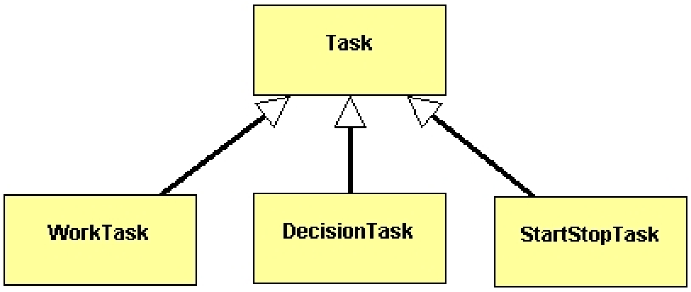

Figure 6.2: Analysis Class Diagram: WorkflowTasks

6.3 RequestHandling

6.3.1 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.

6.3.2 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]

6.3.3 Class Diagrams

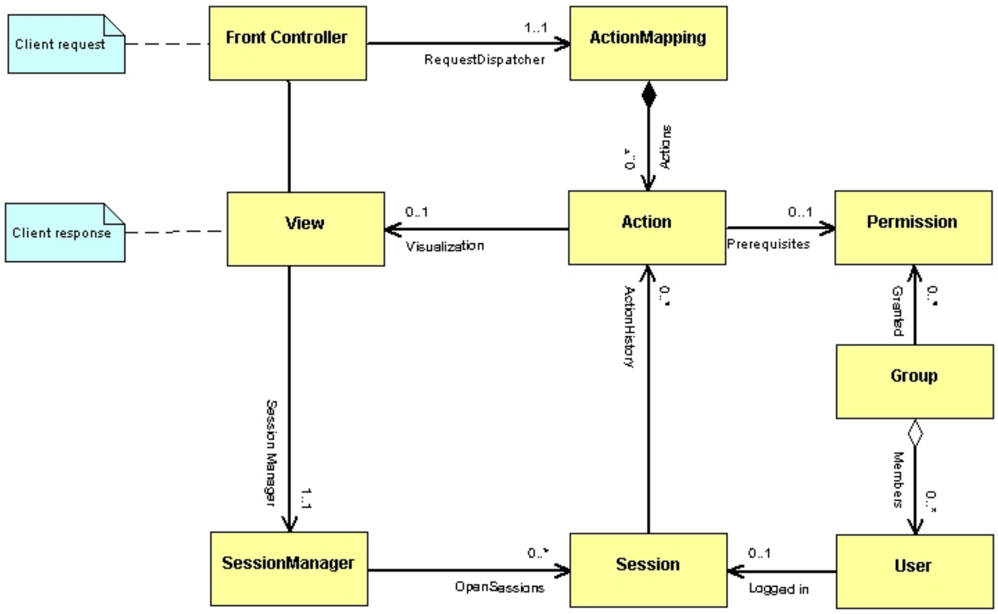

Figure 6.3: Analysis Class Diagram: RequestHandling

6.4 Actions

6.4.1 Description

All so far known needed actions are listed below.

acLogin

acLogout

acCheckWorkFlowTasks

acQueryStories

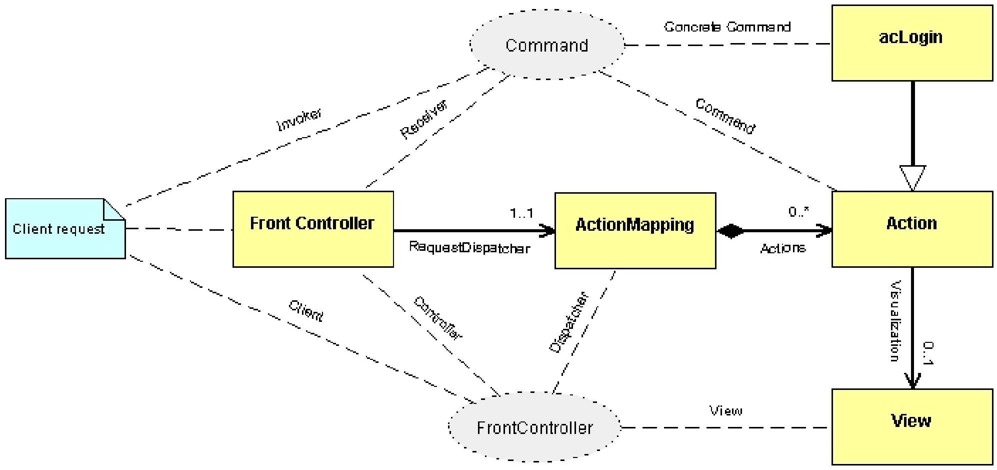

Figure 6.4: Analysis Class Diagram: Patterns used in RequestHandling

acQueryStory

acSaveStory

acReleaseStory

acCancelStory

acGetUserSettings

acSetUserSettings

acSetCalendar

acGetCalender

6.4.2 Class Diagrams

6.5 System users

6.5.1 Description

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

Editor

3rd Party System

Secretary

Reporter

Viewing User

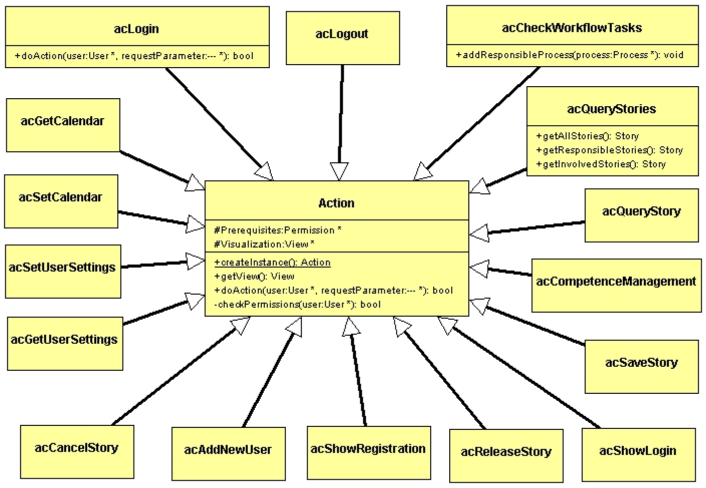

Figure 6.5: Analysis Class Diagram: Actions

6.5.2 Class Diagrams

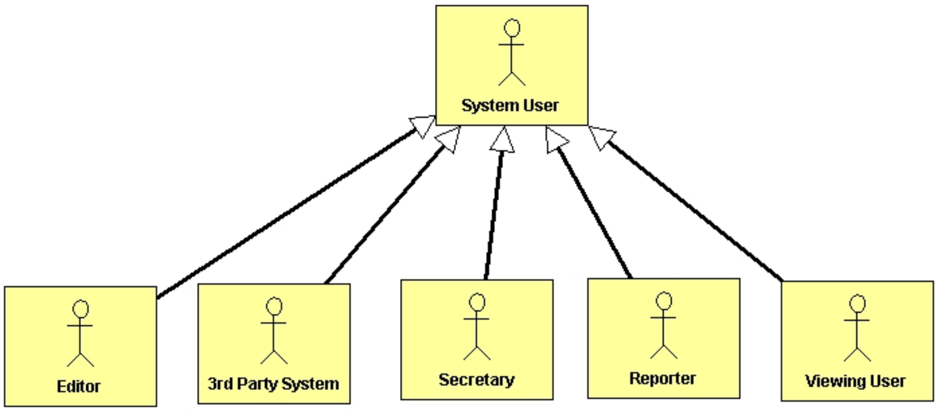

Figure 6.6: Analysis Class Diagram: System Users overview

6.6 Personal Protocol

By this meeting I was the project leader. We had to build some analysis class diagrams that are based on the discussion we had so far, on the two preceding crc sessions, and the essential use cases. These diagrams give a first impression how the different parts of the system work together and build the basic software architecture for InfoWarrior. As project leader I had to present the analysis class diagrams. They are shown in figures 6.1, 6.2, 6.3, 6.5 and 6.6.

Chapter 7

User Interface Prototypes

7.1 User Interface Flow Diagram

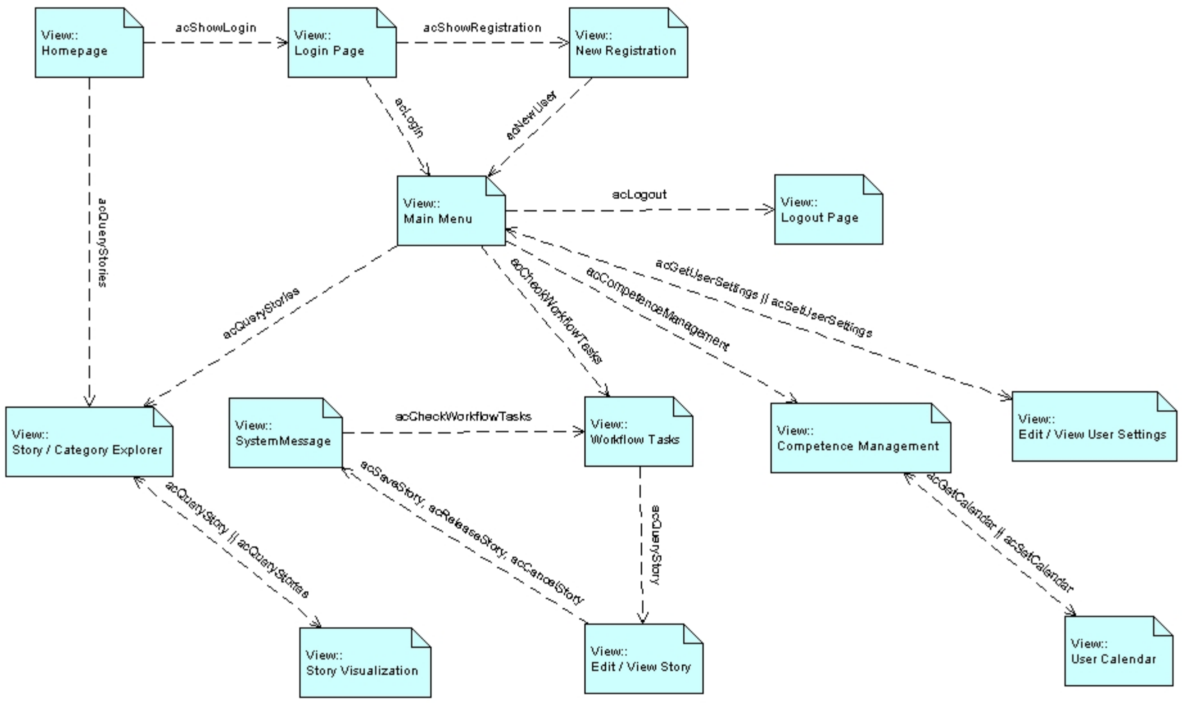

Figure 7.1: User Interface Flow Diagram

7.2 HTML User Interface

7.2.1 Login Page

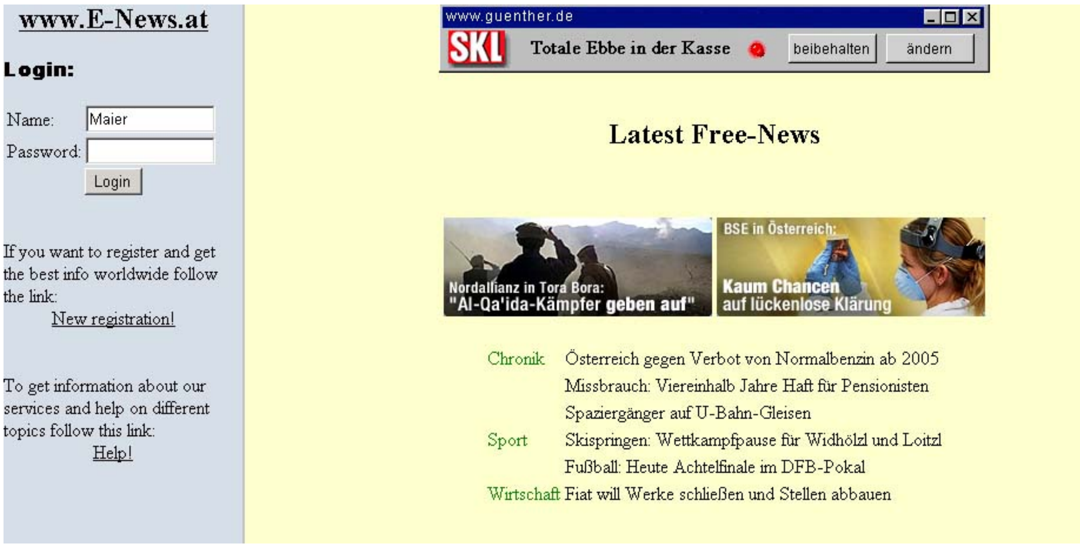

Figure 7.2: HTML user interface: Login Page

7.2.2 New Registration

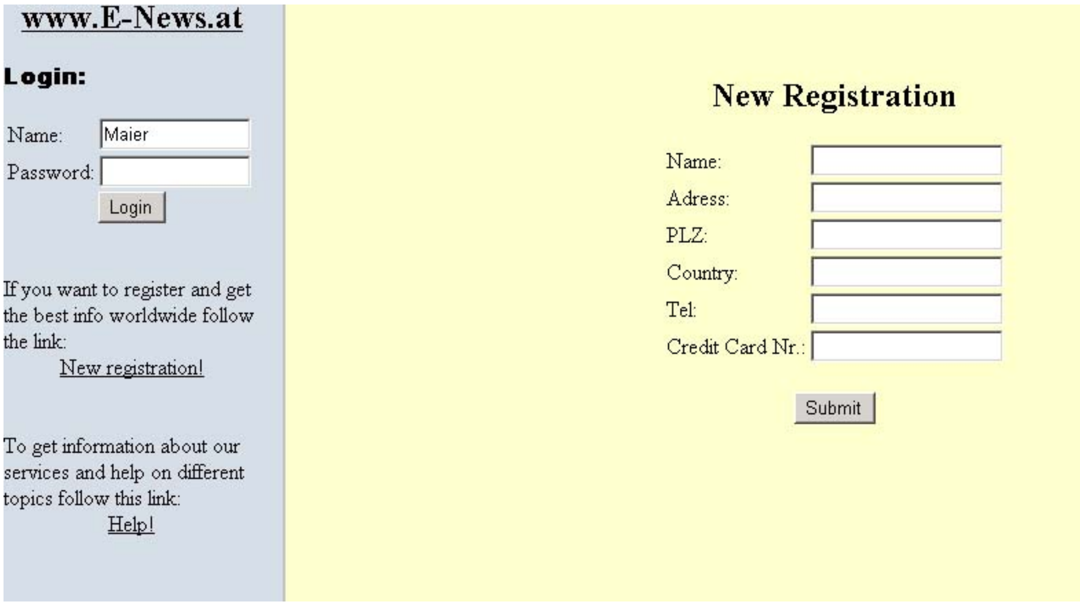

Figure 7.3: HTML user interface: New Registration

7.2.3 Main Menu

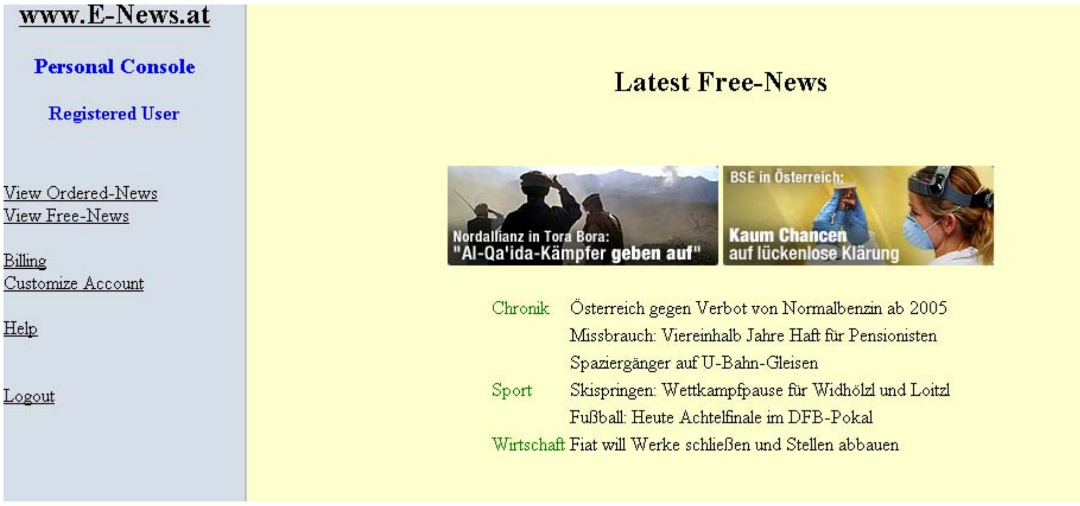

Figure 7.4: HTML user interface: Main Menu

7.2.4 Check WorkFlow Tasks

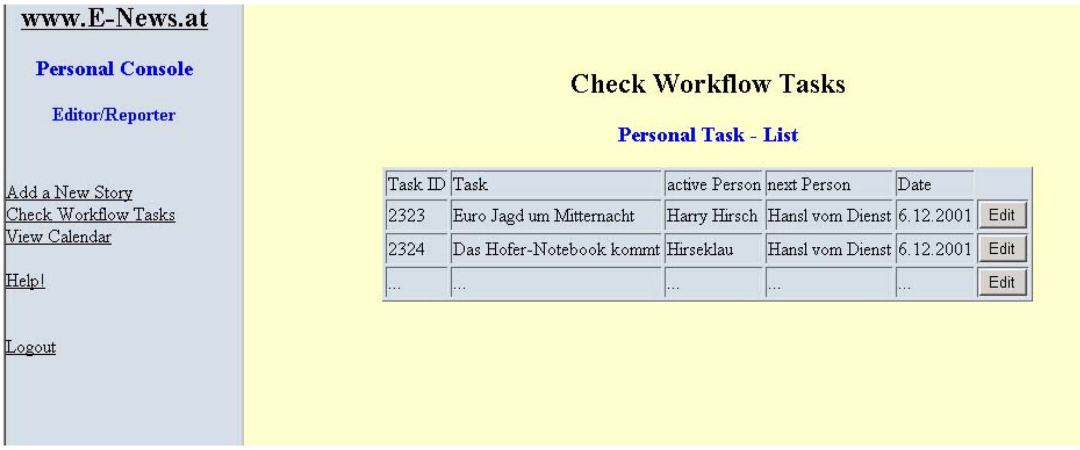

Figure 7.5: HTML user interface: Check WorkFlow Tasks

7.2.5 Edit/View Stories

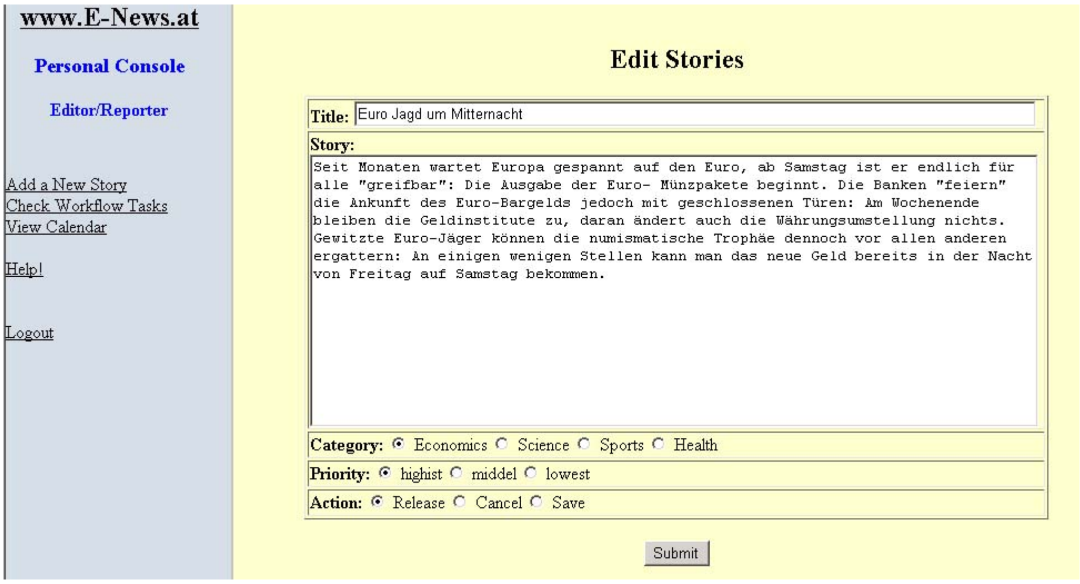

Figure 7.6: HTML user interface: Edit/View Stories

7.2.6 Calendar

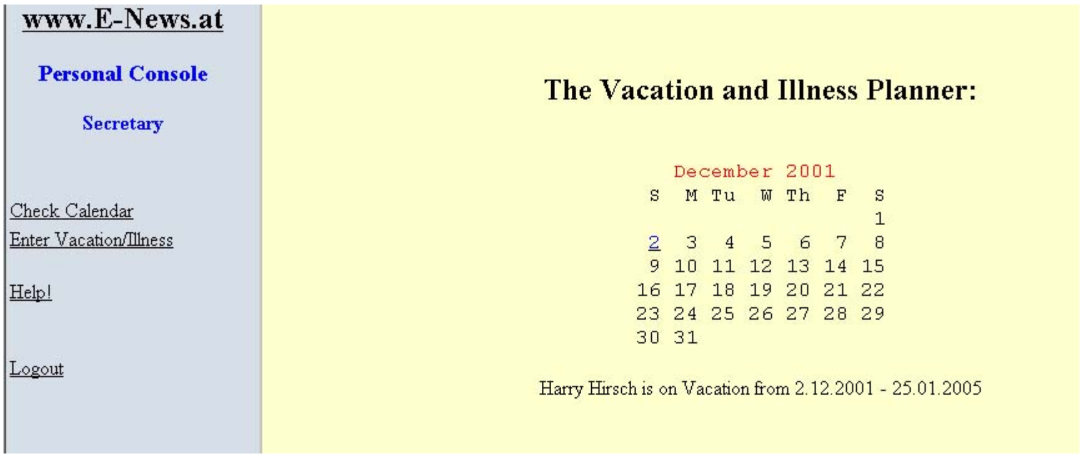

Figure 7.7: HTML user interface: Calendar

7.2.7 Competence Management

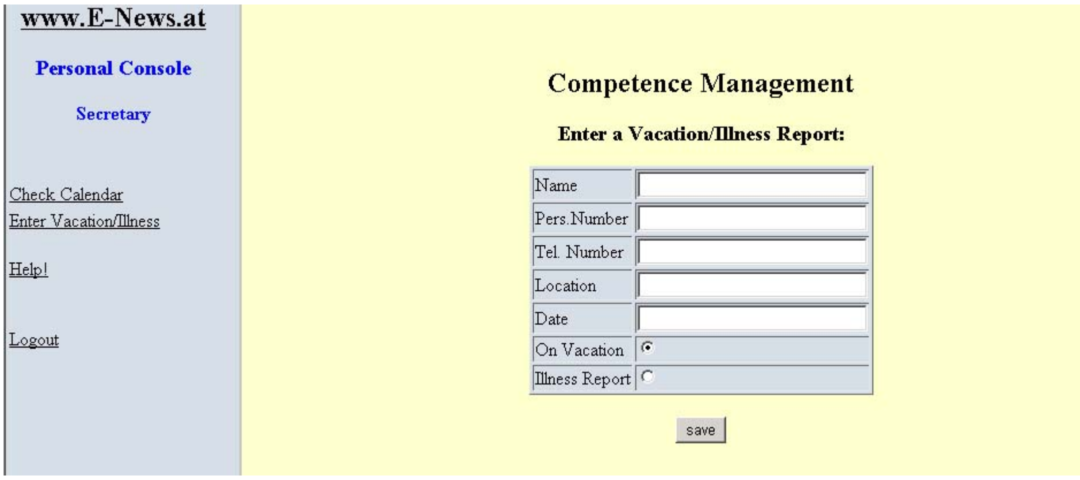

Figure 7.8: HTML user interface: Competence Management

7.2.8 Visualization (Ordered News)

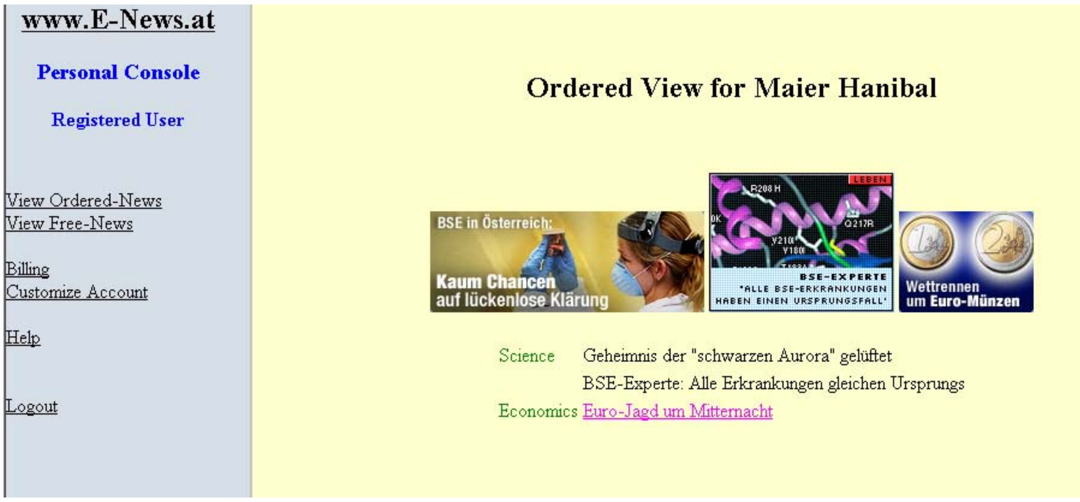

Figure 7.9: HTML user interface: Visualization

7.2.9 User Customization

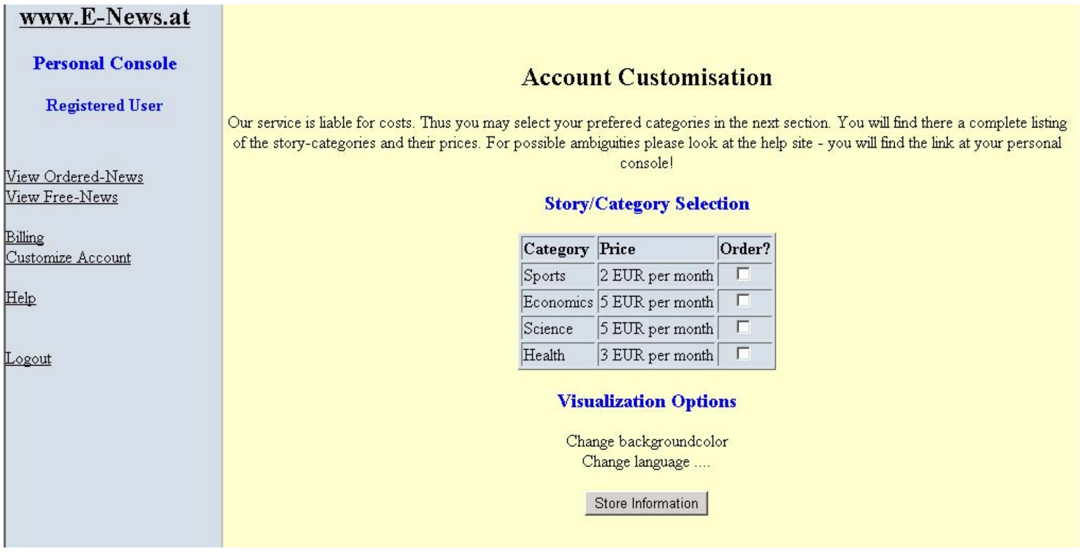

Figure 7.10: HTML user interface: User Customization

7.2.10 Billing Statistics

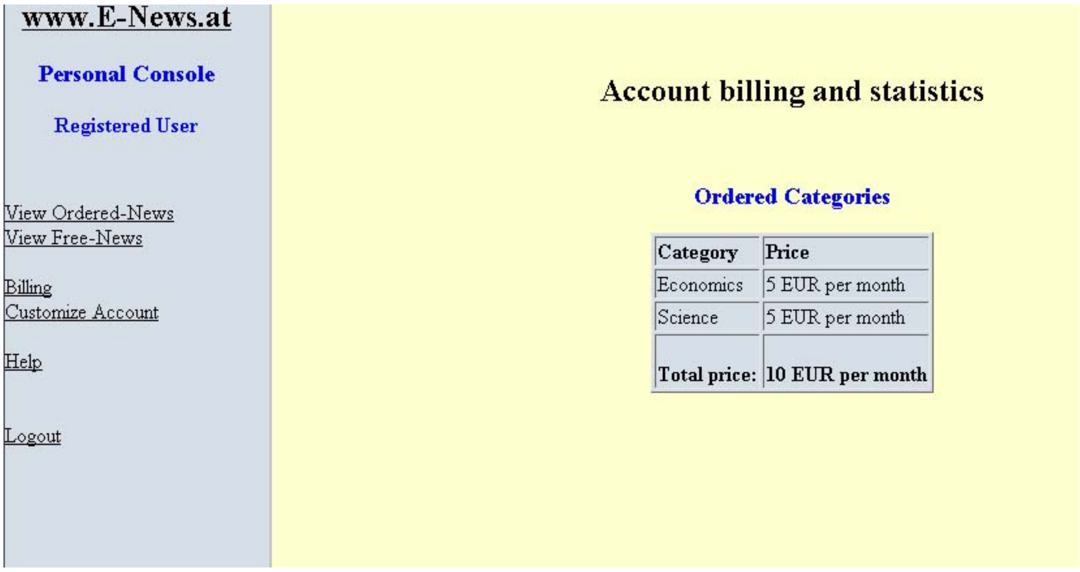

Figure 7.11: HTML user interface: Billing Statistics

7.3 Personal Protocol

In cooperation with Erich Eichberger I created the HTML User Interface. Because there are different potential users on the HTML site, it was not easy to design the pages. We decided to split them into two frames. The left one is the personal console. Every user (editor/reporter, secretary, normal user) gets his own console.

At the login page, users are able to look free news in the right frame. In the left frame users can login, register or look at the help site.

If someone is logging on, the database will look to which user group he belongs to. Then he gets his permission.
When the user had ordered some news, he can read them or search for other news. In the left Frame he has the possibility to get help, to change his personal settings, to order news or to look at the billing.
If the person logged on is a secretary then on the right frame a calendar is mapped. On the left frame he has the possibility to add new vacations or illness reports. Also he gets help on the left frame.
When the logged in person is a reporter or editor on the right frame he can see the workflow tasks in which he is involved. By clicking on the button he can edit the story. In the left frame he can add a story in the workflow system, but he can also view the calendar, and he is also able to get help.

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.