University of Salzburg
Institute of Computer Science

PS Software Engineering Part I

InfoWarrior

Online Information System for News Agencies

Personal Documentation

Thomas Aschauer

Date: November 7, 2001
Version: 1.0

Contents

1 Introduction (Task 1) **4**
1.1 Rough System Requirements 4

2 The First CRC Session (Task2) **5**
2.1 Fundamentals of the CRC Session 5
2.2 Planning the CRC Session 5
2.3 Scenarios 5
2.3.1 Information Grabbing Process 5
2.3.2 Information Presentation Process 7
2.4 Identified Classes 8
2.5 Role Allocation 8
2.6 Goals 8
2.7 Open Questions 9
2.8 Session Protocol for October 31, 2001 9
2.9 Session Results 9

3 The Second CRC Session (Task 3) **10**
3.1 Session Goals 10
3.2 Essential Use Case Modeling 10
3.2.1 Introduction 10
3.3 Essential User Interface Prototyping 11
3.3.1 Introduction 11

List of Figures

List of Tables

Chapter 1

Introduction (Task 1)

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

1.1 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.

Chapter 2

The First CRC Session (Task2)

This chapter describes the fundamentals till the finished CRC session.

2.1 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.

2.2 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.

2.3 Scenarios

2.3.1 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.

2.3.2 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.

2.4 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

2.5 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

2.6 Goals

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

2.7 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?

2.8 Session Protocol for October 31, 2001

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.

2.9 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.

Chapter 3

The Second CRC Session (Task 3)

3.1 Session Goals

The primary goal of the second CRC session will be the identification of use cases with its corresponding actors.

3.2 Essential Use Case Modeling

3.2.1 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 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.

3.3 Essential User Interface Prototyping

3.3.1 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.