University of Salzburg
Institute of Computer Science

PS Software Engineering Part I (InfoWarrior)

Online Information System for News Agencies

Thorsten Engl

Date: November 7, 1999
Version: 1.0

Contents

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 CRC session

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 Goals

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

Chapter 3

Essential User Interface Prototyping

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, your goal is to focus on your users and their usage of the system, not system features. This is one of the reasons why you want to perform essential use case modeling and essential user interface prototyping in tandem: they each focus on usage. Second, your prototyping tools are very simple, including white boards, flip chart paper, and sticky notes. The minute that you introduce electronic technology to your prototyping efforts you have made a design decision about the implementation technology. If you use an HTML development tool to build a user interface prototype then you have immediately narrowed your design space to the functionality supported within browsers. If you choose a Java development environment then you have narrowed your design space to Java, and if you choose a Windows-based prototyping tool, you narrow your design space to whatever is supported on the Windows platform. Understand the problem first, then solve it.

Chapter 4

User Interface Prototyping

4.1 Introduction

User interface prototyping is an iterative analysis technique in which users are actively involved in the mocking-up of the UI for a system. UI prototyping has two purposes:

- It is an analysis technique because it enables you to explore the problem space your system addresses.
- UI prototyping enables you to explore the solution space of your system, at least from the point-of-view of its users, and provides a vehicle for you to communicate the possible UI design of your system.

4.2 Cycle

- Determining the Needs of Your Users
- Building the Prototype
- Evaluating the Prototype
- Determining if you're finished

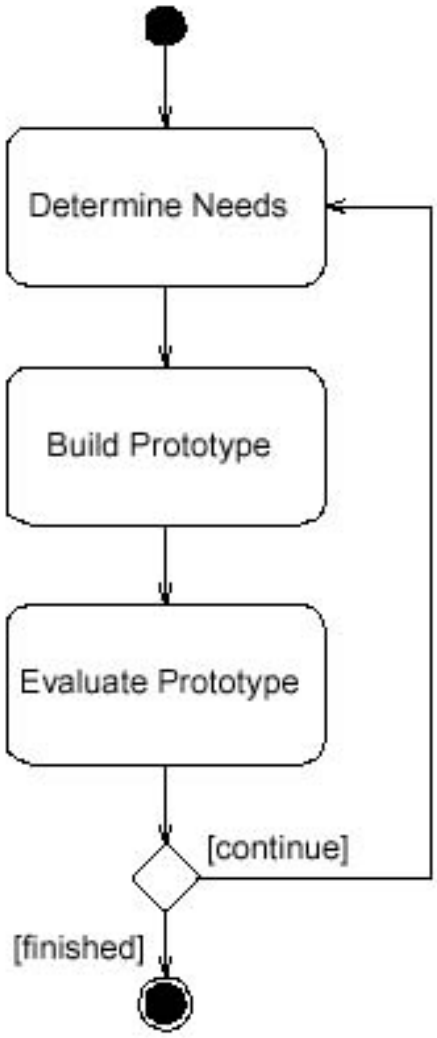

Figure 4.1: Iterative steps of prototyping

Chapter 5

The Use Cases

5.1 Introduction

An important goal in requirements modeling is to come to an understanding of the business problem that your system is to address, in order to understand its behavioral requirements. With respect to object-oriented development, the fundamental artifact that you should develop to model behavioral requirements is a use case model. There are two basic flavors of use case models:

- Essential use case models
- System use case models

5.1.1 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.

5.1.2 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.

5.2 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.