Use Case modelling, Requirements Analysis and Requirements Specification for Web Applications

Week 2

ELEC3609

Frank Moisiadis

1

Tutorial on converting problem statements to use cases for Deliverable 1

- Look at problem statements and pull out actions, flow of events (high level functional view of system).
- From these events, actions, functions, group associated ones (high level use cases).
- Expand into steps (basic course of use cases)
- Expand use cases into their extensions

2

Tutorial in converting use cases to SRS statements for Deliverable 1

- Use:
  - Dependencies among the use cases
  - Pre and post conditions
  - Triggers
  - Assumptions made in use cases
  - Open issues resolved in use cases
- Turn use cases into "shall" statements
- Group statements under themes or functions or qualities
- Use titles/basic concept of use cases as high level SRS statements
- Use the parent-child relationship to write the requirements:
  - Specify all details under parent requirement statements.

3

Problem Statements

Project: Member services information system	Project manager: Sandra Shepherd
Created by: Sandra Shepherd	Last updated by: Robert Martinez
Date created: January 9, 2003	Date last updated: January 15, 2003

Brief Statements of Problem, Opportunity, or Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. Order response time as measured from time of order receipt to time of customer delivery has increased to an average of 15 days.	ASAP	High	$175,000	2	New development
2. The recent acquisitions of Private Screenings Video Club and GameScreen will further stress the throughput requirements for the current system.	6 months	Med	75,000	2	New development
3. Currently, three different order entry systems service the audio, video, and game divisions. Each system is designed to interface with a different warehousing system; therefore, the intent to merge inventory into a single warehouse has been delayed.	6 months	Med	515,000	2	New development
4. There is a general lack of access to management and decision-making information. This will become exasperated by the acquisition of two additional order processing systems (from Private Screenings and GameScreen)	12 months	Low	15,000	3	After new system is developed, provide users with easy-to-learn and -use reporting tools.
5. There currently exist data inconsistencies in the member and order files.	3 months	High	35,000	1	Quick fix; then new development
6. The Private Screenings and GameScreen file systems are incompatible with the SoundStage equivalents. Business data problems include data inconsistencies and lack of input edit controls.	6 months	Med	unknown	2	New development. Additional quantification of benefit might increase urgency.
7. There is an opportunity to open order systems to the Internet, but security and control are an issue.	12 months	Low	unknown	4	Future version of newly developed system
8. The current order entry system is incompatible with the forthcoming automatic identification (bar-coding) system being developed for the warehouse.	3 months	High	65,000	1	Quick fix; then new development

What do we use for Requirements Analysis in UML

- Use Case diagram and descriptions – Why?
- Sequence diagrams – Why?
- Activity diagrams – Why?
- Analysis Class diagrams with associations (has), generalisation (inherits), and aggregation (contains)
- State diagrams for important objects.

5

Use cases

- A use-case is a description of how one of the actors uses the system to accomplish a certain goal
- Use-cases describe in natural language the complete functionality of a system. Each use-case is a snapshot of a particular aspect of a system.
- A use case diagram shows the relationship among actors and use cases within a system. An actor is a role of object or objects outside of a system that interacts directly with it

6

Development of New Use case Template: One Approach:

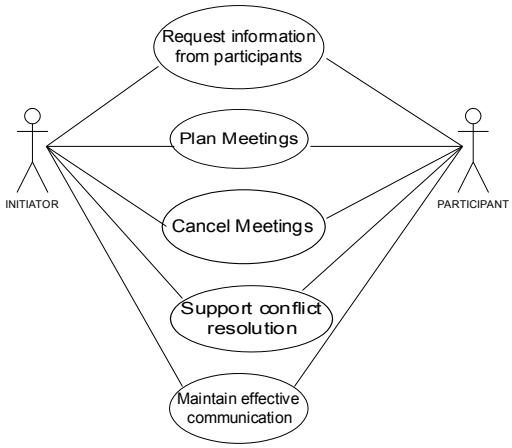
Identify all the actors who use the system

- Identify the Use cases of the system
- Identify the interaction (association) between actors and various Use cases
- Develop High level Use case Diagram
- Develop Detailed Use case Diagram showing all includes and excludes
- Provide descriptions for each Use case

7

Case study-A Meeting Scheduler System

High Level Use case Diagram – there are mistakes in the example – Can you find them?



8

Associations

- ⟶ actor triggers the use case
- ——— actor participates in use case.

- What about association between actors? Hierarchy of actors?
- Actor ---- actor

9

Template:Use case Description

Use case	Name of the use case
Goal	Main goal to be achieved by Use case
Scope&Level	Scope and level of Use case
Pre-Condition	A list of conditions that must be met before the Use case starts
Success end condition	Successful condition with the implementation of Use case
Failed end condition	Failed condition with the implementation of Use case
Actors	Actors that initiates Use case
Trigger	Anything that starts a train of actions

10

Template(contd)

Description	Steps Action
	…… ……...
Extends	Mention the Use case that adds to the existing functionality and characteristics of parent use case
Includes	Mention the Use case that is depicted as using the functionality of another Use case
Sub-variation	A list of branching actions
Stakeholders and interests	What is the interest of stakeholders to this Use case
Frequency of Use case	How often this Use case is used
Level of risk	What is the risk with the failure of this Use case
Priority	How important is this Use case

11

Use case Description

Use case1	Request Information from participants
Goal	To make requests to participants to provide information regarding their preferences on meeting date and location and receive the information.
Scope&Level	High-level system level
Preconditions	Initiator knows the participants who are to be informed and also their contact details for the meeting.
Success end condition	Initiator forwards request to participants for information and participants receive request and respond back to initiator with all details on time.
Failed end condition	Initiator doesn't receive preference details from participants.
Actors	Initiator, Participants
Trigger	Initiator needs information from Participants

12

Use case description (cont' d)

Description	Step Actions
	1. Initiator requests preference details from participants 2. Participants respond with all preference details to the initiator
Extends	2A Participant does not respond within 48 hours 2A1 Remind participant (separate use case 22)
Includes	(use case 11) Propose meeting details (with initiator, participants, preference dates, times and locations)
Sub variations	Initiator sends request for information by means of Email, letter or fax

13

Use case description (cont' d)

Stake holders and interests	• Initiator collect information from participants to arrange meeting • Participants need to provide their preferences in order to make meeting of their convenience
Frequency of use	Very frequently used (for every meeting)
Level of risk	High risk – could lead to cancelled meetings or conflicts
Priority	1

14

Class diagram

- Class diagrams with associations (has), generalisation (inherits), and aggregation (contains)
- Each class has a name, attributes, and operations.
- Aggregation (part of), composition (part to only one whole)

15

Sequence diagrams

- Sequence diagrams
- Sequence: Actors and Objects on top, line is object's lifeline, arrow between 2 objects is a message, from top to bottom, conditions of message use [ ], iteration marker, return as dashed line.
- Shows identification - interaction of objects found in use case.
- Sequence emphasises sequence (order) – time line.
- To understand objects for class and state diagrams

16

State Diagrams

- States an object and how state changes as events reach use case.
- Start → transition Event[Guard]/Action
- Eg. Item received[some items not in stock]/get next item
- →state with do/activity
- Eg. "Checking" do/check item

17

State and Activity Diagrams

- State diagrams with superstates
- Concurrent state diagrams
- Shows behaviour of an object over several use cases.
- ACTIVITY DIAGRAMS:
- Flow of activities, triggers from activities with guards (logical true or false result), synchronisation bar (activities can occur in parallel order irrelevant)
- Can handle parallel processes

18

Software Requirements Specification (SRS) template

TABLE OF CONTENTS

19

SRS template

20

SRS explained

- **1.0 INTRODUCTION**

This document specifies all the requirements for

- **1.1 Purpose**

The purpose of the …is to ….

The system should assist ….

The intended audience for this document is …

This specification describes …..

- **1.2 Scope**

This document applies only to ……

This specification is not concerned with …..

21

SRS explained

- **1.3 Definitions, Acronyms, and Abbreviations**

SRS - Software Requirements Specifications

IEEE - Institute of Electrical and Electronic Engineering

- **1.4 Reference**

[1] IEEE 830-1993: IEEE Recommended Practice for Software Requirements Specifications" IEEE Standards Collection, IEEE, 1997.

- **1.5 Overview**

In the following sections of this specification……will be presented.

In Section 2, the general product and its functions will be introduced.

22

SRS explained

In Section 3, all detailed requirements will be specified and grouped.

In Appendix …….

2.0 GENERAL DESCRIPTION

2.1 Product Perspective

This system allows stakeholders to…..

The system will display…..

The system will help ……

The system provides information about ….

2.2 Product Functions

The system provides the following functions:

23

SRS explained

- **2.3 User Characteristics**

The users of the system are:

- **Level of Users' Computer Knowledge**
- **Level of Users' Business Knowledge**
- **Frequency of Use**

- **2.4 General Constraints**

The system will support ….

The system will not allow ……

- **2.5 Assumption and Dependencies**

This system relies on ……

The system must have a satisfactory interface and ……

24

Section 3 of SRS

- **SPECIFIC REQUIREMENTS**
  3.1 Functional Requirements
  3.1.1 Unit Registration
- The unit registration requirements are concerned with functions regarding unit registration which includes students selecting, adding, dropping, and changing a unit.
- **SRS-001 (3.1.1.1):**
- The system shall allow the user to register a unit.
- **SRS-002 (3.1.1.2):**
- STS shall allow the user to delete a unit if the user has chosen to drop that unit.
- **SRS-003 (3.1.1.3):**
- STS shall check if a unit has been filled by enough registered students.

25

SRS functional egs

- **SRS-004 (3.1.1.4):**
- STS shall allow the user to add his/her name to the unit waiting list if the user wants to register in a unit which has been filled already with enough registered students.
- **SRS-005 (3.1.1.5):**
- STS shall automatically register the unit for the user who is the first one on the waiting list if a vacancy appears for that unit.
- **SRS-006 (3.1.1.6):**
- STS shall allow the user to change practical session(s) within a unit.
- **SRS-007 (3.1.1.7):**
- STS shall allow the user to change tutorial session(s) within a unit.

26

Functional parent reqs broken into many child-reqs.

- **3.1.2 Retrieving and Displaying Unit Information**
- The retrieving and displaying requirements are concerned with how information is retrieved and presented to the user.
- **SRS-014 (3.1.2.1):**
- The system shall allow users to enter the following selection criteria to retrieve unit information: by unit code, by unit number, by title of unit, by weight of unit (credit points).
- **OR** by unit code **(3.1.2.1.1)** , by unit number **(3.1.2.1.2)** , by title of unit **(3.1.2.1.3)** , by weight of unit (credit points) **(3.1.2.1.4).**

27

Design Constraints (3.2)

- **3.2 Design Constraints**
- **SRS-031 (3.2.1):**
- STS shall store and retrieve persistent data.
- **SRS-032 (3.2.2):**
- STS shall support PC and/or UNIX platforms.
- **SRS-033 (3.2.3):**
- STS shall be developed using the JAVA programming language

28

Non-functional requirements

- **3.3 Non-Functional Requirements**
- **SRS-034 (3.3.1):**
- STS shall respond to any retrieval in less than 5 seconds.
- **SRS-035 (3.3.2):**
- STS shall generate a report within 1 minute.
- **SRS-036 (3.3.3):**
- STS shall allow the user to remotely connect to the system.
- **SRS-041 (3.3.8):**
- The system will be accompanied by a comprehensive user manual.

29

Safety and security issues

- **3.5.3 Security**
- The security requirements are concerned with security and privacy issues.

SRS-029:

- VSS shall provide staff ID and password verification protection to protect from unauthorised use of the system.

SRS-030:

- VSS shall allow the store manager to add, remove and modify staff ID and passwords as required.

30

Other SRS template for section 3

- 3. Specific Requirements
- 3.1 External Interface Requirements
- 3.1.1 User Interfaces
- 3.1.2 Hardware Interfaces
- 3.1.3 Software Interfaces
- 3.1.4 Communication Interfaces
- 3.2 Functional Requirements
- 3.2.1 Requirement 1
- 3.2.1.1 Introduction
- 3.2.1.2 Inputs
- 3.2.1.3 Processing
- 3.2.1.4 Outputs
- 3.2.2 Requirement 2 .....

31

Other SRS template for section 3

- 3.3 Performance Requirements
- 3.4 Design Constraints
- 3.4.1 Standards Compliance
- 3.4.2 Hardware Limitations ……
- 3.5 Software System Attributes
- 3.5.1 Reliability
- 3.5.2 Availability
- 3.5.3 Security
- 3.5.4 Maintainability
- 3.5.5 Portability
- 3.5.6 Reusability
- 3.5.7 Usability 3.5.8 Other Factors .....
- 3.6 Other Requirements 3.6.1 Database 3.6.2 Operations .....

32