Use Case Templates for SIMAT

Author

Ademar Aguiar
ademar.aguiar@fe.up.pt

FEUP - Faculdade de Engenharia da Universidade do Porto
www.fe.up.pt
INESC Porto - Inst. Engenharia de Sistemas e Computadores do Porto
www.inescporto.pt
Projecto SIMAT[1]
http://simat.inescn.pt

Use cases can be applied in many different fashions, they can be of various levels of detail, and can have varying format. Here, we will define the use case templates to follow in the SIMAT/INESC project for the description of use-cases in three formats: high-level, expanded, and detailed.

Introduction

Use-case modeling is an important tool for developing an outside view of a system, and is a powerful means of communication between users and developers.

To large systems, one flat level of use cases is normally insufficient to effectively capture and partition the large amount of functionality present. Usually it is needed to define multiple use case levels at different granularities as well as describing the relationships between the use cases at each level.

The various formats of describing use cases are used in different phases of the use case modeling process, and reflect the level of detail of the analysis done so far. Initially, use cases descriptions are vague and are increasingly being detailed till a level that fulfills their intent, i.e., proper understanding of the functionality of the system to be developed.

[1] SIMAT is an acronym of "Sistema de Informação Municipal - aplicações técnicas em tecnologia SIG"

© INESC 1998, all rights reserved.

High level Use-Cases

The high-level use cases initiated by each actor are identified in a first step. A high-level use case identifies the event that is a responsibility of the system to be performed. Events are actions that an actor initiates with the system in order to complete some activity. These events are valid within the context of the environment and should be described shortly so that they are understandable to the user.

Expanded Use Cases

Once the set of high-level use cases have been defined, the next step is the elaboration of the activities involved in each use case. The functionality defined in the high-level use-cases needs now to be progressively elaborated into more detail. The first step of this progress is the definition of *Expanded Use Cases*.

These use cases describe, in more detail, the activities involved in the interaction with the actor. There is a one to one mapping between each high-level use case and each expanded use case.

During this step, documenting exception or alternative conditions that may occur is avoided. Sometimes however, alternate or exceptions conditions are critical to the conceptual understanding of the behavior in the use case, so in these cases it is recommended to document this behavior.

Although use cases are very good at capturing system behavior or functionality, there are nonfunctional aspects or issues associated with an event that are not easily captured in the basic course.

Because such issues, as performance or security, associated with the interaction defined in the use case are difficult to model in the use case course they are documented in the assumptions sections.

Detailed and Abstract Use Cases

A detailed use case is a decomposition of an expanded use case, to be elaborated on steps in the expanded use cases.

The detailed use case provides more information on a specific step or steps in expanded use case. There may be a one to many relationships between an expanded use case and detailed use case.

Detailed use cases themselves can have detailed use cases, but it is important to remember that the goal of use case modeling is to understand the problem sufficiently to validate it with the user and to develop a design, and the goal is not to code the system.

Document Templates

Use cases associate a lot of documentation that needs to be managed. Well-organized documentation can improve the understandability of the system being developed and can support the traceability from system-level use cases to design models and test plans. Additionally to the functionality required to the system, there are other kinds of information that should be documented, including priority, risk and status of development.

As development goes further from the analysis to coding and testing phases, the level of description of the use-case becomes more detailed and new kinds of information must be included in the documentation. In the table below are presented the sections included in the use-case documents produced in SIMAT project, for each level of use-cases: high-level, expanded, and detailed. Each section is described later on the document.

Document Section	High-Level	Expanded	Detailed
Name	✓	✓	✓
Actors	✓	✓	✓
Goal			✓
Overview	✓		✓
Type		✓	✓
Priority		✓	✓
Status			✓
Preconditions		✓	✓
Description	✓	✓	✓
Postconditions		✓	✓
Exceptions			✓
Related Use Cases			✓
User interface			✓
Activity diagrams			✓
Subordinate Use Cases			✓
Participating Classes			✓
Special Requirements			✓
Assumptions		✓	
References	✓	✓	✓

Table 1: Document sections for each level of use-cases.

Document Sections

Name The name of the use case.

Actors The entire different user roles and/or systems that initiate the use case, the primary actors.

Goal The purpose of the actor on initiating and performing the use case.

Overview A very brief description of the most important interactions between the actor and the system. Usually two or three sentences are enough.

Type Type of the use case: primary/secondary/optional; essential/real/concrete; ... (FIXME).

Priority A statement of the importance of the use case for the project, typically described in a qualitative way, such as, high-medium-low. This priority can reflect a more quantitative indication: a calculated weighted score based on some defined qualities: architectural impact, design effort, function complexity, technology risk.

Status An indication of the development state of the use case.

Examples are:

- analysis;
- design;
- implemented in the demonstrator;
- implemented in the final application;
- to be implemented in the final application named <name>.

Preconditions The necessary conditions that have to be met before the use case can be performed. They could be other use cases as well.

Description A detailed description of the interactions between the actor and the system. This is the basic course or the typical sequence of events that should be taken by the system. The general form of prose that works well is something like:

<time or sequence factor> ... <actor> ... <action> ... <constraints>

If there is a natural order to the steps in the basic course, then number the steps. Otherwise, just use bullet points or repeat the same number for steps that can be performed at once.

Alternative courses can be shown using branching or by listing them under the section Exceptions.

Branching can be described using if-statement's. Repetitions can use for-statement's and while-statement's.

Postconditions The state of the system after the use case is performed, reflecting the main results achieved due to the interactions described.

Exceptions All the different alternative courses that can potentially be taken for various reasons: pre-conditions not met or other environment conditions not holded as expected.

Related Use Cases Enumeration of all the associations of this use case with others, grouped by type: by *uses* - Use cases that are associated by *uses* association arcs; by *extends* - Use cases that are associated by *extends* association arcs.

User interface The user interface pictures of this use case taken from the prototype or from the future system to be developed.

Activity diagrams Activity diagrams, or references to them, that represent the sequence of events of the use case, or some complex parts of it.

Subordinate Use Cases An use-case diagram, or a reference to it, that represents the subordinate use cases. A subordinate use case is an use case supplied by a subsystem that is used to realize this use case, named, superordinate use case.

Participating Classes Collaboration showing all the classes whose objects interact to implement this use case.

Special Requirements Any special non-functional requirements not referred yet that must be satisfied when performing this use case.

Assumptions Any assumptions that are made about the use case should be explicitly described here.

References All the sources that provided information regarding this use case, mainly document references and personal contacts. Here can also be included references to other artifacts related with this use-case: subsystem model, analysis model, design model, code, or test plans.

References

[Armour95] F. Armour, Lorrie Boyd, M. Sood; "Use Case Modeling Concepts for Large Business System Development"; ??

[Cockburn] Alistair Cockburn; "Structuring Use Cases with Goals"; ???

[Jacobson92] I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard; *Object-Oriented Software Engineering - A Use Case Driven Approach*; Addison-Wesley; 1992

[Jacobson94a] Ivar Jacobson; "Basic Use-Case Modeling"; ROAD, Vol. 1, No. 2, pp.15-19, July-August 1994

[Jacobson94b] Ivar Jacobson; "Basic Use-Case Modeling (continued)"; ROAD, Vol. 1, No. 3, pp.7-9, September-October 1994

[Jacobson94c] Ivar Jacobson; "Use Cases and Objects"; ROAD, Vol. 1, No. 4, pp.7-9, November-December 1994

[Jacobson95a] Ivar Jacobson; "Use Cases in Large-Scale Systems "; ROAD, Vol. 1, No. 6, pp.9-12, March-April 1995

[Jacobson95b] Ivar Jacobson; "Systems of Interconnected Systems"; ROAD, Vol. 1, No. 7, pp.???, May-June 1995

[Jacobson95c] I. Jacobson, S. Bylund, P. Jonsson, S. Ehnebom, "Using Contracts and Use Cases to Build Plugable Architectures"; ROAD, Vol. ?, No. ?, pp.???, ?

[Schneider98] Geri Schneider, Jason P. Winters; "Applying Use

Cases: a practical guide"; Addison Wesley; 1998.