

Use cases

Paul Jackson

School of Informatics
University of Edinburgh

Use cases

- ▶ An important part of any requirements document for a system is a description of the system's behaviour from the viewpoint of its users.
- ▶ Behaviour can be broken down into units, each triggered by some stakeholder
- ▶ **Use cases** are one way of describing these units

What is a use case?

A **task** involving the system which **has value** for one or more stakeholders.

Actors are stakeholders who take an active part in the task

Each use case

- ▶ has a discrete goal the **primary actor** wishes to achieve
- ▶ includes a description of the a sequence of messages exchanged between the system and actors, and actions performed by the system, in order to achieve the goal.

The **primary actor** usually is the one triggering the use case.

Supporting actors may also be involved

Some stakeholders may be neither primary or supporting actors

Actors in use cases

An **actor** can be:

- ▶ a human user of the system *in a particular rôle*
- ▶ an external system, which *in some rôle* interacts with the system.

Not a user: a particular *kind* of user. E.g., Bank Customer.

The same human user or external system may interact with the system in more than one rôle: he/she/it will be (partly) represented by more than one actor. (e.g., a bank teller may happen also to be a customer of the bank).

Paths in a use case

Usually a use case describes a core of sequence of steps necessary to achieve its goal.

However might be alternatives, including some handling when all does not go to plan and the goal is not achieved.

Each path through use case called a *use-case instance* or *scenario*

- ▶ One talks about the **main success scenario** and alternate success or **failure scenarios**.

All scenarios in a use-case tied together by common user goal.

Warning: Sometimes *scenario* and *use-case* are synonyms

Example of use-case paths

Goal: Buy a Product

Main Success Scenario

1. Customer browses catalogue and selects items to buy
2. Customer goes to check out
3. Customer fills in shipping info
4. System presents full pricing info
5. Customer fills in credit card info
6. System authorises purchase
7. System confirms sale immediately
8. System sends confirmation email to customer

Example of use-case paths (cont)

Extensions - variations on main success scenario

3a : Customer is regular customer

.1 : System displays current shipping and billing info

.2 : Customer may accept or override these defaults, returns to MSS at step 4, but skips step 5.

6a . System fails to authorize credit card purchase

.1 : Customer may re-enter credit card information or may cancel

Example fields in use case template

- ▶ **Goal** What the primary actor wishes to achieve
- ▶ **Summary** A one or two sentence description of the use case.
- ▶ **Stakeholders** and each's **Interest** in the use case
- ▶ **Primary actor**
- ▶ **Supporting actors**
- ▶ **Trigger** The event that leads to this use case being performed.
- ▶ **Pre-conditions/Assumptions** What can be assumed to be true when the use case starts
- ▶ **Guarantees** What the use case ensures at its end
 - ▶ Success guarantees
 - ▶ Minimal guarantees
 - ▶ Failure guarantees
- ▶ **Main Success Scenario**
- ▶ **Alternative scenarios**

Use cases: connections and scope

A use case:

- ▶ can have different levels of detail, e.g. depending on where it is used in development process
- ▶ may refer to other use cases to provide further information on particular steps
- ▶ May describe different scopes: e.g. a system of systems, a single system or a single component of a system

Requirements capture organised by use cases

Use cases can help with requirements capture by providing a structured way to go about it:

1. identify the actors
2. for each actor, find out
 - ▶ what they need from the system
 - ▶ any other interactions they expect to have with the system
 - ▶ which use cases have what priority for them

Good for both requirements specification and iterated requirements elicitation.

Use cases primarily capture functional requirements, but sometimes non-functional requirements are attached to a use case.

Other times, non-functional requirements apply to subsets or all of use-cases.

Other uses of use cases in software processes

Driving design

Design validation

- ▶ You can walk through how a design realises a use case, checking that the set of classes provides the needed functionality and that the interactions are as expected.

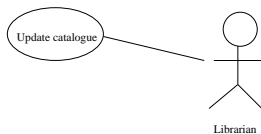
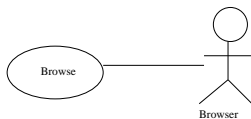
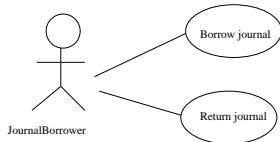
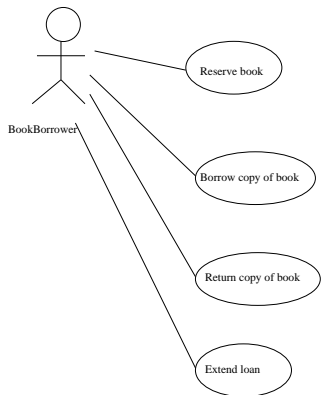
Testing

- ▶ Use cases can be a good source of system tests

Possible problems with use cases

- ▶ Interactions spelled out may be too detailed, may needlessly constrain design
- ▶ May specify supporting actors that are not essential for fulfilling goal of primary actor
 - ▶ Does borrowing a book have to involve a librarian?
- ▶ Focus on operational nature of system may result in less attention to software architecture and static object structure
- ▶ May miss requirements not naturally associated with actors

Gathering use cases together



This **use case diagram** notation is part of UML

The Unified Modeling Language

UML is a graphical language for recording aspects of the requirements and design of software systems.

It provides many diagram types; all the diagrams of a system together form a UML model, which must be consistent (in a weak sense...).

Mostly tailored to an OO world-view

Often used just for documentation, but in **model-driven development**, a UML model may be used e.g. to generate and update code and database schemas automatically.

Many tools available to support UML

Reading

Required: *Tokeneer ID Station System Requirements Specification*.

See Section 5 for use-case-like scenarios. More generally, browse the document to get a feel of what a real Requirements Specification looks like.

Suggested: *Sommerville*. Use Cases discussed are both in Requirements and System Modeling chapters. Look up Use Cases in index to find the relevant sections.

Suggested: *Stevens*, Chapter 7.

Suggested: *Fowler*, Chapter 9.