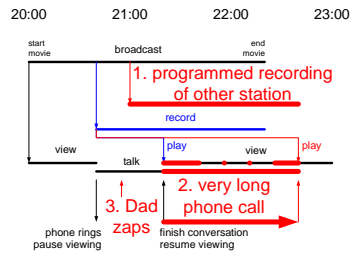# Use Case How To

-

Gerrit Muller

University of South-Eastern Norway-NISE

Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway

gaudisite@gmail.com

Abstract

Use cases are frequently used in Software Engineering. Use cases support specification and facilitate design, analysis, verification and testing. Many designers, unfortunately, apply use cases in a rather limited way. This presentation provides recommendations for effective use cases.

1 Introduction

The *use case* technique is frequently used in specification and design, for example RUP[2] is advocating it is as tool. The power of use cases is that they put specifications in *user* perspective. Use cases facilitate *analysis and design* and *verification and testing* by providing concrete inputs. In practice the following problems arise when use cases are used:

- designers apply the technique too local, for example software only

- the use cases are limited to functionality, ignoring quantified information

The purpose of this article is to explain the use case technique at system level, applied in a multi-disciplinary way. We will show how to obtain understanding from use cases of typical use and how to analyze the specification and design for worst cases and boundary conditions.

2 Example Personal Video Recorder

We use *time shift recording* as a use case of desired user functionality. Figure 1 shows the concurrent activities that occur when straightforward time shifting is used. In this example the user is watching a movie, which is broadcasted via conventional means. After some time he is interrupted by the telephone. In order to be able to resume the viewing of the movie he pauses the viewing, which starts invisible the recording of the remainder of the movie. Sometime later he resumes viewing where he left of, while in the background the recording of the not yet finished movie continues.
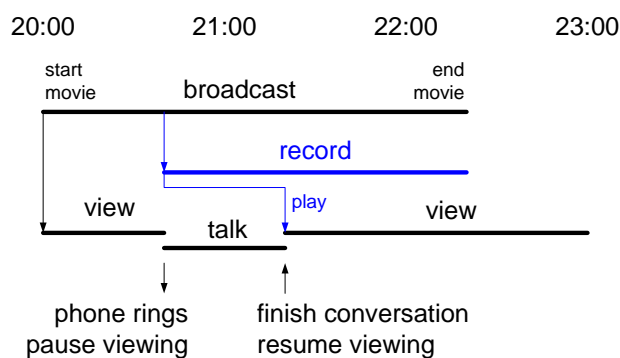

Figure 1: Example use case Time Shift recording

In this simple form (pause/resume) this function provides freedom of time to the user. This appears to be very attractive in this interaction modus. However

---

when such an appliance is designed limits out of the construction world pop up, which intrude in the user experience. The list below shows a number of construction limits, which are relevant for the external behavior of the appliance.

- number of tuners

- number of simultaneous streams (recording and playing)

- amount of available storage

- management strategy of storage space

Construction limits, but also more extensive use cases, see figure 2, show how the intrinsic simple model can deteriorate into a more complex interaction model. Interference of different user inputs and interference of appliance limitations compromise the simplicity of the interaction model.
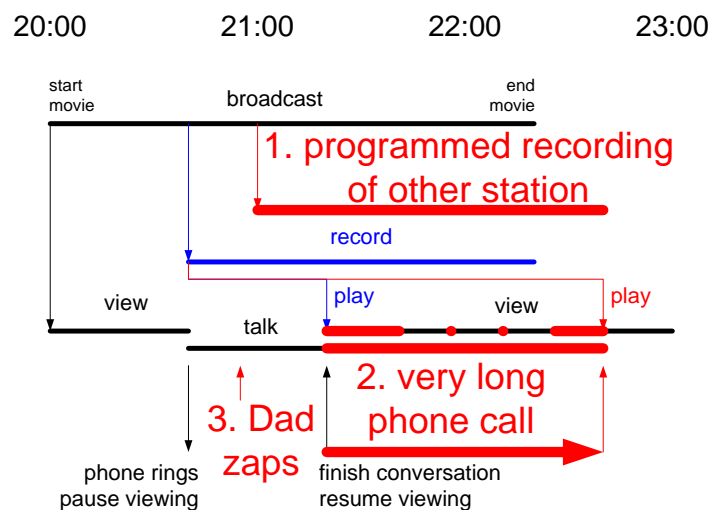

Figure 2: What if conflicting events happen during the pause interval?

3 The use case technique

Figure 3 shows what elements should be present in a use case. The purpose of the use case is to make the specification clear of functionality or behavior of the system and what the desired non-functional requirements (or qualities) are. The use case technique can also be applied for technical interfaces, where the use case illustrates the specification from the perspective of the using system.
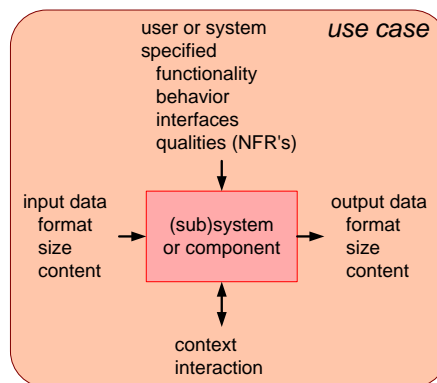
Figure 3: Content of a Use Case

The use case also described the input of the (sub)system in terms of format, size and content. The expected outputs are described with the same attributes. Then the interaction with the context of the system must be described, as far as relevant for this specific use case.
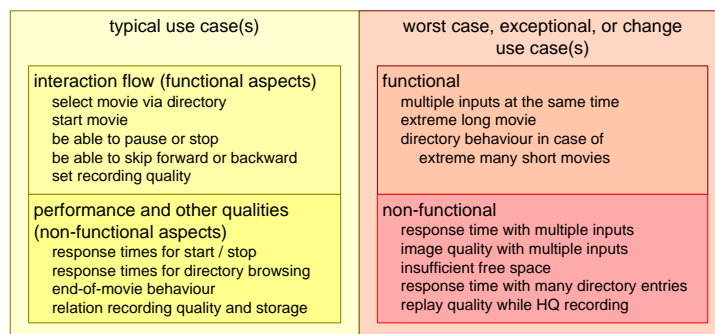

Figure 4: Example personal video recorder use case contents

Figure 4 shows the elements of two use cases also from the personal video recorder domain. At the left hand a typical use case is presented: watching a pre-recorded movie. The right side shows the elements of examples of worst cases or boundary cases. At the bottom of both use cases the possible quantification is shown. For example in the typical case user response times can be specified or image quality in relation to required storage capacity. For worst cases many more numbers are relevant for design. These worst case numbers arise from the confrontation of the extremes of the user needs with the quantification of the technology limitations.

4 Example URF examination

This use case example focuses on the quantification aspect. Figure 5 shows the typical case for URF (Universal Radiography Fluoroscopy) examinations when used image intestines. Three examination rooms are sharing one medical imaging workstation. Every examination room has an average throughput of 4 patients per hour (patient examinations are interleaved, as explained below for Figure 6).
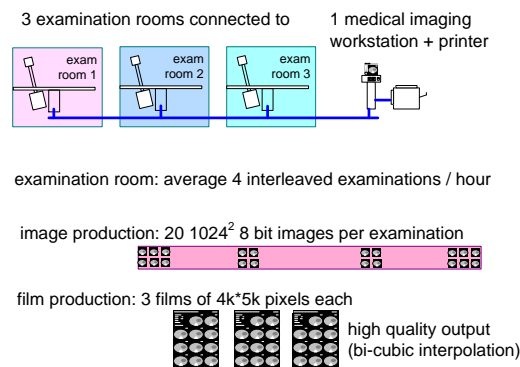

Figure 5: Typical case URF examination

The average image production per examination is 20 images, each of 1024^2 pixels of 8 bits. The images are printed on large film sheets with a size of approximately $24*30cm^2$. One film sheet consists of 4k by 5k pixels. The images must be sufficiently large to be easily viewed on the light-box. These images are typically printed on 3 film sheets. Image quality of the film sheets is crucial, which translates into the use of bi-cubic interpolation.

Figure 6 shows how patient examinations are interleaved. The patient is examined over a period of about one hour. This time is needed because the barium meal progresses through the intestines during this period. A few exposures are made during the passage of clinical relevant positions. The interleaving of patients in a single examination room optimizes the use of expensive resources. At the level of the medical imaging workstation the examinations of the different examination rooms are imported concurrently. The workstation must be capable of serving all three acquisition rooms with the specified typical load. The latency between the end of the examination and the availability of processed film sheets is not very critical.

The amount of worst case and boundary situations is very large, so selection of relevant ones is needed. The danger of working out too many use cases is that all this work is also transformed in realizations and verifications resulting in excessive implementation efforts. Reduction of the amount of use cases can be done in steps, by replacing several detailed use cases by one slightly more gener-
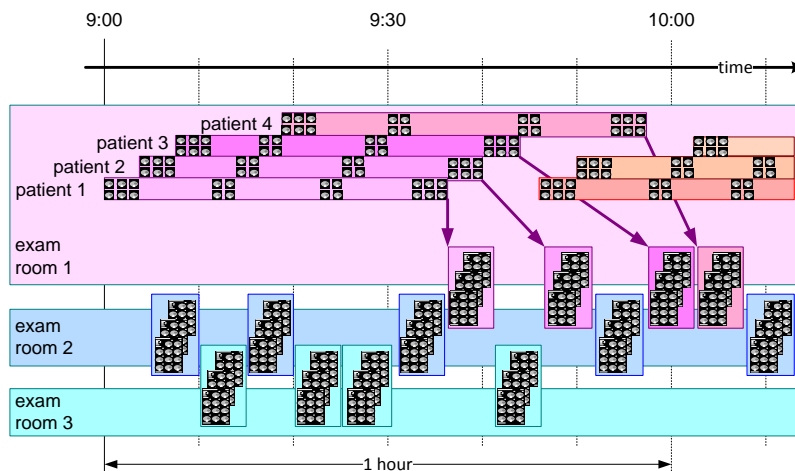
Figure 6: Timing of typical URF examination rooms

alized use case. The consequence of such a transformation is that also the design is simplified, where the focus should be on excellent performance of typical use cases and acceptable performance and behavior for worst cases and exceptional cases.

5 Summary

Figure 7 summarizes the recommendations for use cases. A common pitfall is that people describe use cases at single function level. The usage aspect disappears in this way and many small problems become invisible. Therefor a good use case combines several functions into one user activity. The use case should be quantified to make it useful for design, analysis and verification. The amount of use cases should be limited.

References

[1] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.

[2] Wikipedia. Rational unified process (rup). `http://en.wikipedia.org/wiki/Rational_Unified_Process`, 2006.

+ combine related functions in one use case

- do not make a separate use case for every function

+ include non-functional requirements in the use cases

+ minimise the amount of required *worst case* and
exceptional use cases
- excessive amounts of use cases propagate to
excessive implementation efforts
+ reduce the amount of these use cases in steps
- a few well chosen *worst case* use cases simplifies the design

Figure 7: Recommendations for working with use cases

History
Version: 0.1, date: 15 June 2007 changed by: Gerrit Muller
- Created text version
- set logo
Version: 0, date: 21 November 2006 changed by: Gerrit Muller
- Created, no changelog yet