USE CASES AS A SEMI-FORMAL WAY OF DESCRIPTION OF THE ETCS FUNCTIONAL BEHAVIOUR

Michal Kunhart[1], Jakub Marek[2], Jan Ouředníček[3]

The contribution presents methodology of functional requirements specification of the system ERTMS/ETCS. This methodology is based on Use Cases of Unified Modelling Language.

In the first part there is a brief introduction to Use Cases - their description, aim and structure. Next there is a description of particular parts of Use Cases, their aim, qualities and their mutual relationships with respecting solved problem.

The following part is dedicated to relationships between particular Use Cases and their hierarchical arrangement.

There are also mentioned the relationships to the following steps of the functional requirements specification and to verification and validation activities in the contribution. Also Use Cases management problem is presented.

Key words: Use Case, Unified Modelling Language, Methodology of Functional Requirements Specification, ERTMS/ETCS

1 Introduction

Making a verbal model normally precedes whatever activities heading towards making a formal model of system behaviour. Its target should be approximate clarification required behaviour of the modelled system. Also *Use Cases* have the same target.

As the primary methodology of the verbal functional requirements for the activities heading towards the designing of the functional behaviour of Radio Block Centre (RBC), generally of the system European Rail Traffic Management System / European Train Control System Level 2 (ERTMS/ETCS L2), was chosen Use Cases. It is a written (verbal) form of the description with obtaining of certain structured level of knowledge for more effective transition to (semi-)formal description technique.

This begs the question: Why the Use Cases? In the team which members are the authors of this contribution was for the modelling of the system functional behaviour decided to use primarily *Unified*

[1] Ing. Michal Kunhart, AŽD Praha s. r. o. (Ltd), Technika Plant, Research, Žirovnická 2/3146, 106 17 Prague 10, Czech Republic, tel.: +420 466 773 344, E-mail: kunhart.michal@azd.cz

[2] Ing. Jakub Marek, University of Pardubice, Jan Perner Transport Faculty, Department of Electrical and Electronical Engineering and Signalling in Transport, Studentská 95, 532 10 Pardubice, Czech Republic, tel.: +420 466 036 387, E-mail: jakub.marek@upce.cz, Mr Marek also works for AŽD Praha s. r. o. (Ltd), Technika Plant, Research, Žirovnická 2/3146, 106 17 Prague 10, Czech Republic, tel.: +420 466 773 344, E-mail: marek.jakub@azd.cz

[3] Ing. Jan Ouředníček, AŽD Praha s. r. o. (Ltd), Technika Plant, Research, Žirovnická 2/3146, 106 17 Prague 10, Czech Republic, tel.: +420 466 773 345, E-mail: ourednicek.jan@azd.cz

Modelling Language (hereinafter UML). Justification of this decision is supported by document [2], which advises to use UML for the specification, verification and validation of ERTMS/ETCS (hereinafter ETCS).

UML does not directly unify the notation of Use Cases, but it works with them. UML includes the *Use Case Diagram* (see Fig. 1).

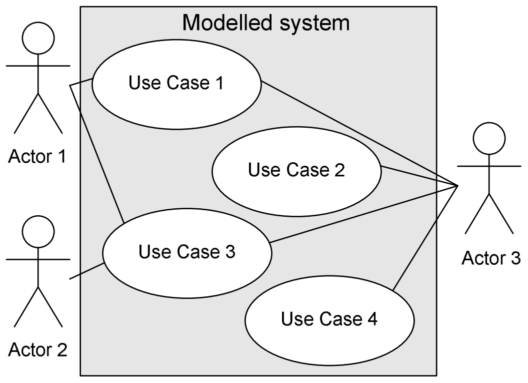

Fig. 1: Use Case Diagram according to UML standard

However, Use Cases is not only “bubbles” and “little figures”. For example, publication [3] shows why it is necessary to add a text description into the graphic statement of the UML standard. Also the experiences of the authors of this contribution show that making the meaningful and useful Use Case Diagram is not possible without more or less detailed behaviour specification, which is actually presented by each Use Case.

2 Use Cases

2.1 Purpose and definition

As adumbrated above Use Cases (hereinafter UCs) represent one of possible informal means of verbal description of system functional behaviour. It is a written structured text (verbal) notation of particular functionality of the system. Each UC usually represents a one overall functionality or if you like overall functional requirements of the modelled system. They make particular form of capturing (obtaining and documenting) modelled system requirements. Great advantage of this way of notation is in relatively easy and intuitive understanding of UCs for almost every reader.

As mentioned above the UML does not define the notation of UCs then the unified way of their notation does not exist. Each author is able to accept and use his own way of notation. This can bring advantages (adapting to the particular problem) and also disadvantages (difficult portability). The fact related to this features is that there are many different recommendations for making the UCs.

2.2 Specificity of Use Cases using

In this text is outlined a conception of the UCs that is used for initial requirements specification for functional behaviour of the RBC, generally for the ETCS as whole system. Final notation of the UCs follows from recommendations from the literature [1], [3], [4] and [6], which are focused on newly made IT systems modelling. However, applicability of these recommendations is often limited by differences between both modelled problems (i.e. IT systems versus signalling systems).

It is generally true that absolute majority of the modelled IT systems described in the literature is realised such way to provide some service to their environment. Users of such IT system as first define what they want system to do and supplier's task is to meet these requirements. They are met by creating internal structure of this IT system and by appropriate designing of its interfaces.

In the ETCS case there is a more complicated situation. It is caused not only because of certain safety integrity level requirement on functionality of the system (safety management, safety demonstration and others are related to this problem) but also because of:

- internal structure of the ETCS as whole has been just defined by SUBSETs, whereas it is always necessary to meet this specification of the ETCS because of interoperability of railway infrastructure achievement;
- environment of the ETCS, which is the whole railway system is considerably enough complicated, namely both its technical and organizational side;
- environment of the ETCS is managed by rules and habits that had been determined before ETCS was defined. So these rules and habits are realised without taking ETCS into account. However, a customer requirement is to be compatible with the rules and the habits in maximum possible rate.

Thanks to this specificity many attributes of UCs were omitted or modified in such a way that the aim of use of this methodology was met. It means that founding a fundamental set of functional requirements for the ETCS application was not obstructed by founding "how and what to fill to these attributes". Especially in such a cases when the filling is not clear. On the other hand this stance does not except the possibility of future change or completion of using methodology if it is useful.

2.3 Structure of Use Cases

Notation of UCs made for ETCS modelling has the structure, which is shown on the Figure 2.

Name should characterise the whole UC briefly, clearly and appositely. UC name should also be unique in the whole model. Furthermore, it is suitable to assign a unique identifier to the UC. It makes work with whole group of UCs more effective. The identifier used by authors of this contribution allows UC classification according to type of functionality and to degree of detail.

Actors represent roles which external entities (persons or objects) play in relation to modelled system. One person or one object may play more roles in relation to the system. One actor should be primary actor; the other actors should be so called supporting actors. Primary actor is such an actor, which has a main reason to reach the aim of the UC. Primary actor usually launches the UC. While UC is running, all actors are in direct interaction with this UC, i.e. with the modelled system.

Pre-conditions characterise a state of the system, which is necessary to launch the UC. They represent presumptions that have to be fulfilled before launching the UC.

Trigger is a specific event (moment) that causes launching the UC immediately. This is true only if the pre-conditions are fulfilled (see logical conjunction AND on Fig. 2).

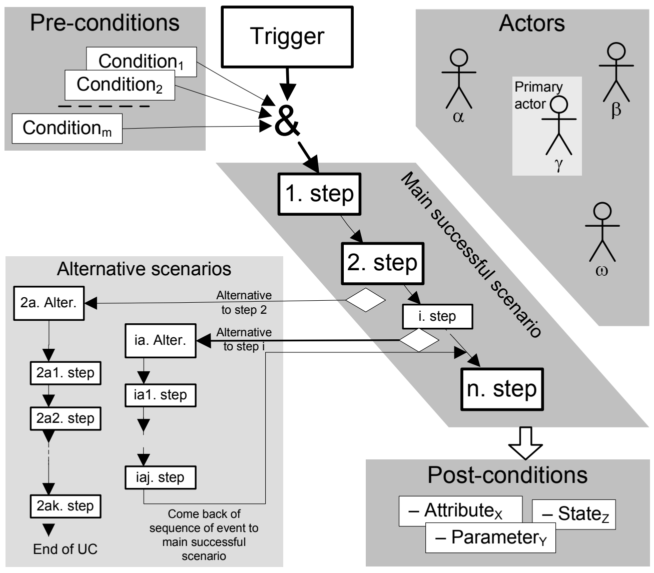

Fig. 2: Use Cases' sections

Main successful scenario describes interaction between actors and modelled system as time passes. It is done like a "theatre" scenario. The main successful scenario describes the situation when all goes right (so that's main *successful* scenario). It is not recommended to write single steps by prose, but it should have a certain logic structure (e.g. their sentences should start with subject).

Alternative scenarios allow catching alternative behaviour of the system. It is useful in cases when it is necessary to deflect from main successful scenario (so that's *alternative* scenarios). One main successful scenario may logically have more alternative scenarios.

Post-condotions characterise a state of the system after finishing this UC. Post-conditions may be related to UC as whole or to each its scenarios.

3 Relations among the Use Cases

3.1 General features

Wide number of actions and a repeated occurrence of identical sequences of events (see Fig. 3a) naturally lead to necessity to create relations among the UCs. Instead of UCs with extensive scenarios with the many same steps arises then (mostly) the smaller number of shorter, mutually connected UCs.

The practise shows, that it is one of the most significant problems. Due to combination of parts of several UCs can lose these UCs their purpose as a description of related actions – sequence of events, and obtain rather the character of state diagram of the part of the system, that participate on the operation of events (see Fig. 3b).

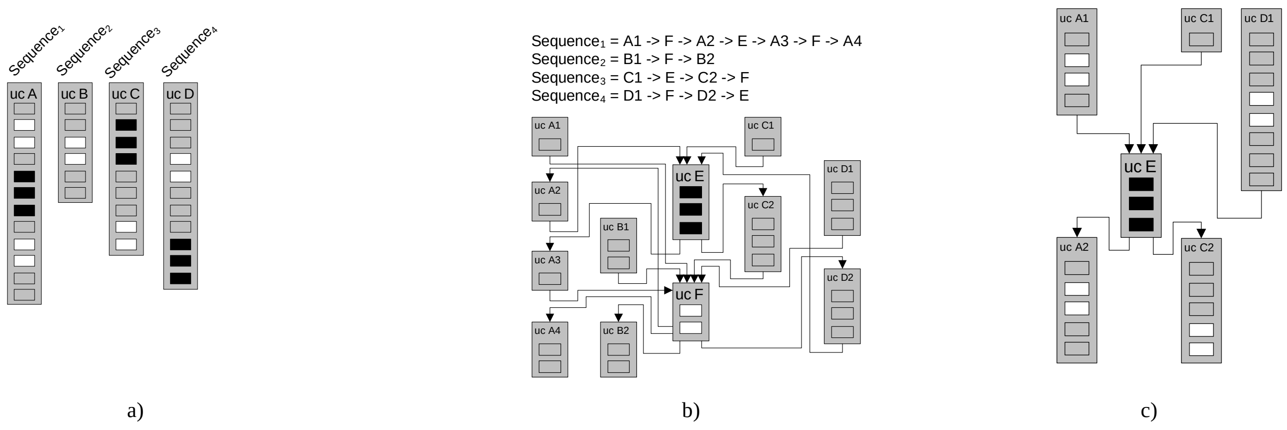
Sequence1
Sequence2
Sequence3
Sequence4
uc A
uc B
uc C
uc D
Sequence1 = A1 -> F -> A2 -> E -> A3 -> F -> A4
Sequence2 = B1 -> F -> B2
Sequence3 = C1 -> E -> C2 -> F
Sequence4 = D1 -> F -> D2 -> E
uc A1
uc A2
uc A3
uc A4
uc B1
uc B2
uc C1
uc C2
uc D1
uc D2
uc E
uc F
a)
b)
c)

Fig. 3: Relations among the

UC: a) Sequences of events with repeated steps, b) Joining of all duplicated passages, c) Partially joining of duplicated passages

On one hand this „degeneration“ of scenarios can be confused for a reader, but on the other hand it is the next step of problem analysis and the identification of repeated situations. In case the mutual relationships among the UCs form the state diagram instead of the series of events, it is good to create new UCs (with lower degree of detail) with the operation of events without loops (see Fig. 3c).

At figure 3a, there are depicted four sequences of events – four UCs. Five pair of white points and the three trios of black points are the repeated steps. Figure 3b describes the situation of complete joining of the repeated steps and figure 3c their partial joining.

The publication [3] deals with the relationships among the UCs only a little. It mentions the relationships, that interrupt the main successful scenario (after accomplishment of the referred UC the previous one continues with the next step), or that have a character of input condition of some UC.

3.2 Relations among the Use Cases in the ETCS

ETCS Use Cases contain the following relations:

- transition to another UC and its termination. In a step of main successful scenario or its alternatives there is a reference to scenario, in which the sequence of events continues, and the original UC is terminated. Return to the original UC is allowed only with its re-start. This way of relation is formulated as follows: “*This step is a trigger of the Use Case UC XX-YYYZ – title of the Use Case*”. In this UC is indicated in the note of the trigger, in which UCs this event occur.
- interruption of the UC (noted above, according to [3]). The step of main successful scenario is described with the self-contained UC. Having finished this UC, the main successful scenario continues with the following step. This way of relation is formulated as follows: “*see Use Case UC XX-YYYZ – title of the Use Case*”.
- UC as a pre-condition (noted above, according to [3]). This way of relation is formulated as follows: “*Use Case took place UC XX-YYYZ – title of the Use Case*”.

- Post-condition as a pre-condition. In the note, there are mentioned UCs, input conditions of which are contained in the output characteristic. This way of relation is formulated as follows: "*Fact described in the output characteristic is one of the input conditions of the Use Case UC XX-YYYZ – title of the Use Case*".

The explicitly described relations are used in the situations, when is appropriate in term of the whole context to express the particular relationships among the UC.

Due to better transparency exists the table of mutual relations among the ETCS UCs. It demonstrates the occurrence of the triggers in all steps of the UCs and the relations among their post-conditions and their pre-conditions. There are not described all relations, only the significant relations for the illustration of the UC in the context of behaviour of the whole system and its environment.

The relations among the UCs are described as follows:

- Step of the main successful scenario of the UC X, step of its alternative → Trigger of the main successful scenario of the UC Y
- Post-condition of the UC X → Pre-condition of the UC Y

Generally the UC X represents the *UC with a trigger* and UC Y the *initiated UC*. It is possible that the event or state in the main successful scenario or in the alternative scenarios starts (that is the necessary condition for starting of) the same UC, i.e. UC X = UC Y.

4 Consequential phase of functional requirements specification

UCs are the basic method of collection of functional requirements for a particular system then UCs present beginning of requirements chain processing and beginning development of whole system. It is not possible to assume that the requirements presented by UCs are complete. If it manages to cover all operational situations by means of UCs, appearance of a gap or an error related to even a small component (however with functional influence of the whole system) would remain a problem of detail of the analysis. For purpose identification of all (in possible range) functional behaviour features a more detailed structuring of requirements and modelling and their verifications are necessary.

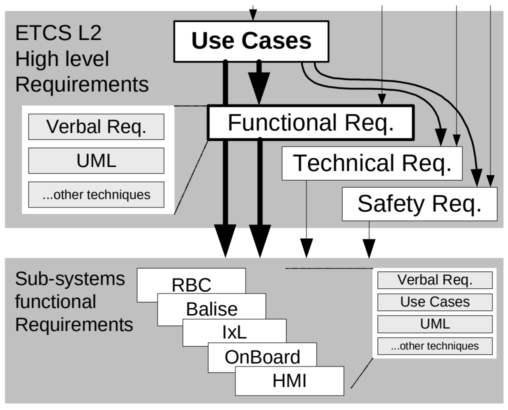

Fig. 4: Further activities coming after UC modelling

Verbal (text) form of requirements immediately comes after the analysis by means of UCs. Such each verbal requirement relates to a particular system feature but does not describe overall sequence of feature unlike UCs. Especially functional requirements are derived from UC, further technical and safety requirements have to be derived. A functional behaviour model based on UML is created at the same together with requirements specification. Such modelling enables more or less to apply formal methods to development process.

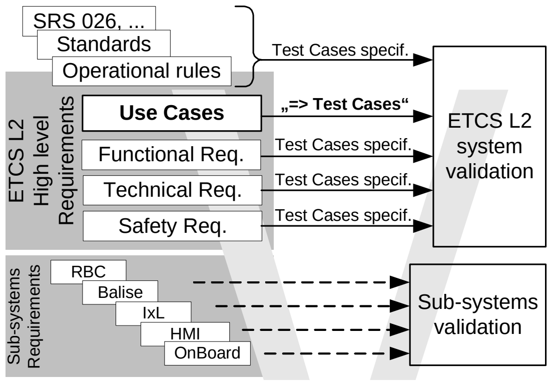

Fig. 5: UCs as a simple source of test cases

Separation of requirements otherwise derivation requirements for particular subsystems and components of ETCS presents further phase. During this phase the UCs are used for auxiliary purposes especially for specification of test cases for verification of particular requirements fulfilment (see also below).

5 Use Cases and Verification & Validation activities

If as basic life cycle model the “V diagram” is taken into account then UCs are in Verification & Validation activities useful in two ways:

a) Validation as a testing of a component or subsystem or system on corresponding level of upward branch of “V diagram” (in our case it is ETCS as whole system). UC scenario actually itself presents test case specification of components and subsystems and system, which are relevant for this UC (see Fig. 5). For the practical usage of UCs as test case specification is necessary to add into them parameters (monitored in framework of the test) and their expected values.
b) Verification next steps (in development process) coming after the making up the UCs (see Fig. 6). For example a clear trace have to exists between a particular result (requirement, definition, parameter value, algorithm, …) of a step and a particular UC eventually its part or set of UCs, which is realized by the particular result. So it is a traceability check. This activity presents verification the UC because inconsistency (between the UC and the requirement, which arise from it) does not have to mean always a problem in the requirement but the problem can be in the UC.

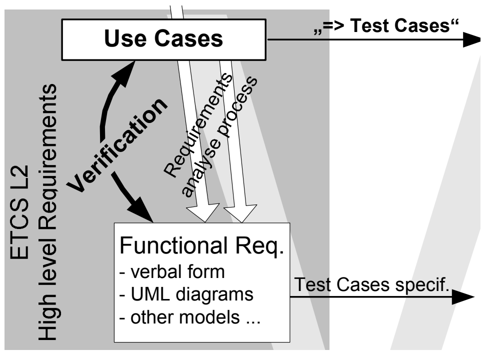

Fig. 6: Functional requirements validation according to UCs

UCs verification concerning to requirements on higher level (SUBSETs, standards, legislative) is also to a certain degree possible by means of the traceability check. However mostly it is realized by means of review method, which is logical because UCs in ETCS as whole system are High Level Requirements.

6 Conclusion

For basic specification of functional ETCS application requirements was chosen the methodology of structured notation of Use Cases form. Particularity of the ETCS application requires specific modification of this methodology against recommendation described in relevant literature.

Grouping of monitored occurrences into a partial Use Cases and their mutual linking is the biggest problem. More detailed structuring allows deeper understanding of the studying problem, but it can cause confusion about this way described behaviour. Optimal ratio between structuring and transparency is little different in different cases.

Reference literature

1. ARLOW, Jim – NEUSTADT, Ila. *UML2 a unifikovaný proces vývoje aplikací: Objektově orientovaná analýza a návrh prakticky*. 1. vyd. Brno: Computer Press, a. s., 2007. 567 s. ISBN 978-80-251-1503-9.
2. CIMATTI, A. – ROVERI, M. – SUSI, A. *ETCS requirements specification and validation: the methodology*. ERA. 2008. 48 s.
3. COCKBURN, Alistair. *Use Cases: Jak efektivně modelovat aplikace*. 1. vyd. Brno: CP Books, a. s., 2005. 262 s. ISBN 80-251-0721-3.
4. KANISOVÁ, Hana – MÜLLER, Miroslav. *UML srozumitelně*. 2. aktualiz. vyd. Brno: Computer Press, a. s., 2006. 176 s. ISBN 80-251-1083-4.
5. MENSE, Olaf. Challenges, successes and progress in ETCS products and pro-jects. In *UIC ERTMS World Conference*. Málaga 2009.
6. SCHMULLER, Joseph. *Myslíme v jazyku UML: knihovna programátora*. Pře-ložil Jiří Hynek. 1. vyd. Praha: Grada Publishing, spol. s r. o., 2001. 359 s. ISBN 80-247-0029-8.