BRAIL – SAFETY REQUIREMENT ANALYSIS

Jean-Louis Boulanger

Universite de technologie de compiegne
Laboratoire Heudiasyc centre de recherche de royallieu
compiegne 60205
France

Keywords: Formalization, Level Crossing, Risk analysis, System Requirements, Traceability, UML.

Abstract: In the European railways standards (CENELEC EN 50126, (1999); EN 50128, (2001); EN 50129, (2000)), it is required to obtain evidence of safety in system requirements specifications. In the railway domain, safety requirements are obviously severe. It is very important to keep requirements traceability during software development process even if the different used models are informal, semi formal or formal. This study is integrated into a larger one that aims at linking an informal approach (UML notation) to a formal (B method) one.

1 INTRODUCTION

Ambiguities and defects in system requirements specification may have consequences on the whole system development. We investigate how the Unified Modelling Language (UML), can be used to formally specify and verify critical railways systems. A benefit of using UML is it status as an international standard (OMG) and its widespread use in the software industries. The reader interested by more details in syntax and semantic aspects can refer to the reference guide of UML). Even if UML notation is a language in which models can be represented, it doesn't define the making process of these models. Nevertheless, several dedicated tools have strengthened the popularity of UML. These tools allow graphic notation and partial generation of the associated code and documentations. The UML notation is known by most computer scientists and is now used in several domains. Using UML class diagrams to define information structures has now become standard practice in industry. Recently, the critical application domains have used the notation and several questions exist around this use. Safety invariants can be derived from hazard analysis and can be supported by a system model in diagrams of UML.

2 CASE STUDY

To illustrate our approach, we will choose to design a level crossing. This example is inspired by Jansen, L. and Schneider, E. (2000). The term level crossing, in general a crossing at the same level, i.e. without bridge or tunnel, is especially used in the case where a road crosses a railway; it also applies when a light rail line with separate right-of-way crosses a road; the term "metro" usually means by definition that there are no level crossings. Firstly, a single-track line, which crosses a road in the same level, is modelled (figure 1). The crossing zone is named danger zone. The most important security rule is to avoid collision by prohibiting road and railway traffic simultaneously on level crossing. The railway crossing is equipped with barriers and road traffic lights to forbid the car passage. Two sensors appear on the railroad to detect the beginning (train entrance) and the end (train exit) of the level crossing protection procedure. The level crossing is not in an urban zone this implies a sound signalisation. Traffic lights consist of two lights: one red and one yellow. When they are switched off, road users (drivers, pedestrians,…) can cross. When the yellow light is shown road users (drivers, cyclists, pedestrians etc.) shall stop at the level crossing if possible. In the other case, the level crossing is closed and railway traffic has priority. The yellow and red light never must be shown together.

Boulanger J. (2005).
BRAIL – SAFETY REQUIREMENT ANALYSIS.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 108-111
DOI: 10.5220/0002507801080111

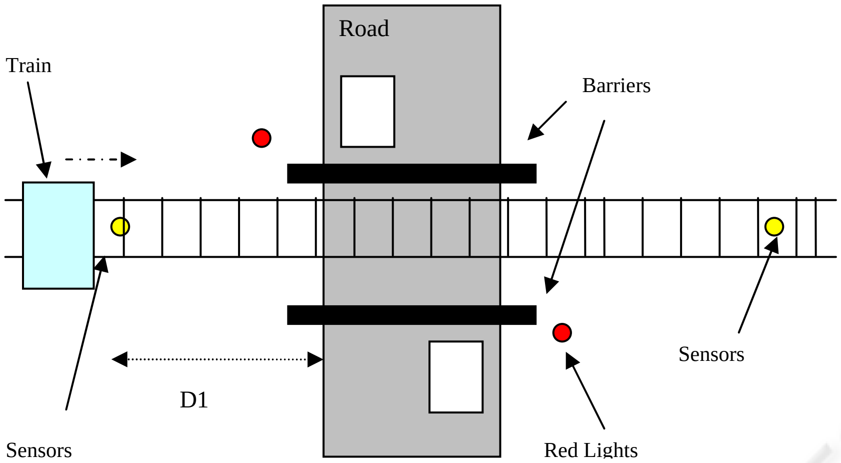

Figure 1: Single-track line level crossing

3 REQUIREMENTS ANALYSIS.

3.1 Environnement

It is often difficult to understand requirements if they are stated as a list. For that reason, functional requirements (and even some non-functional requirements) can be expressed by using some ”use cases”. A use case analysis involves the following steps:
Determine the actors, i.e. any outside entities (people, systems, etc.) that interact with the system.
Identification of Use Cases (name, purpose, goal, pre- and post-condition, ..).
A use case diagram describes and traces the functional requirements of the system and describe how the system can and will be used. The use case diagram gives an overview of the model.
UR3: The railway crossing is equipped with barriers and road traffic lights. Traffic lights at the level crossing consist of a red and a yellow light.

3.2 Failures

The user requirement gives information concerning the failures and their direct effects on the system.
UR12 : Possible failure conditions have to be taken into account for a safe control of the level crossing and the train.
In our model, failures of yellow or red traffic lights (to be separately), barriers, the vehicle sensor and the delay or loss of radio network are considered. Operational scenarios can be specified by means of sequence diagrams of UML.

3.3 Risk analysis

According to EN 50129, (2000) risk analysis essentially consists of four steps:
system definition,
identification of operational hazards,
consequence analysis,
risk assessement.

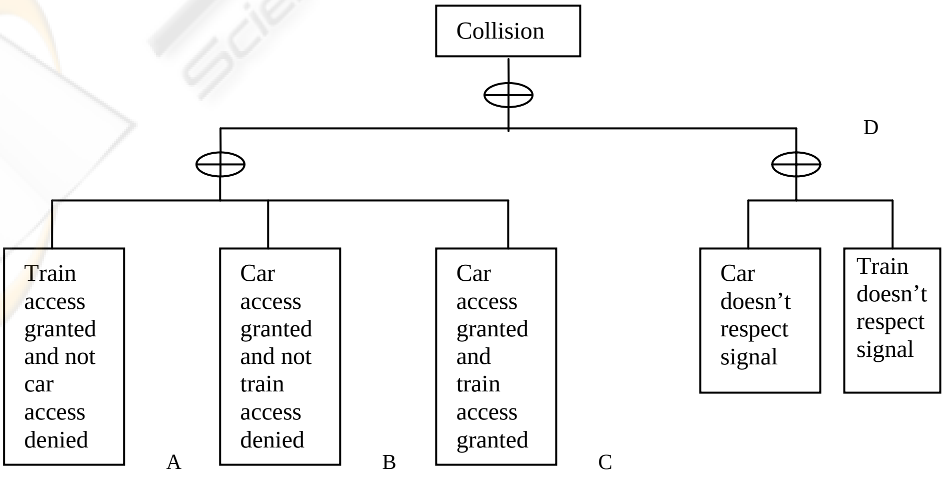
Collision
D
Train access granted and not car access denied
A
Car access granted and not train access denied
B
Car access granted and train access granted
C
Car doesn’t respect signal
Train doesn’t respect signal

Figure 6: Fault Tree Analysis

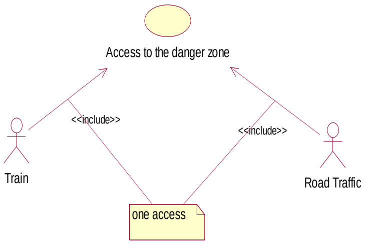

Figure. 2- Use case from UR2

The identification of operational hazards (step 2) can be done by the analysis the user requirement (UR) and/or by analysis of classical risk. In our case, the UR contains:

UR2: The intersection area of the road and the railway line is called danger zone, since trains and road traffic must not enter it at the same time to avoid collision.

In figure 2, this UR introduces a use case that introduces the basic risk. In first time, we derive safety requirement by using FTA (Fault Tree Analysis). A FTA is a graphical technique that provides a systematic description of the combinations of possible occurrences in a system, which can result in an undesirable outcome. This method can combine hardware failures and human failures. For safety-critical systems, the root node of the tree will often represent a system-wide, catastrophic event taken from an existing hazards list. From the collision risk we can derive the next FTA.

The first FTA is split in some part. The D part concerns some human errors. The C part introduces the principle property for the system: “The system does not granted access in same time to train and road traffic”. The A and B part deals with absence and failures of equipments (barrier, traffic light, communication, train sensor).

3.4 System Modelling

For modelling the system structure and interfaces between system objects class diagrams are suitable (see Figure 8). The class diagram describes the relationships between classes and shows the logical view of a system (static view). In respect with safety analysis, the control system provides the capability to authorise the danger zone access for road traffic or for train. This system immediately reports the occurrence and repair of failures to the Operation Center.

3.5 Sub-System Modelling

UR 3: Decentralized radio-based control system

This UR indicates that the system is split in 3 parts:
Communication sub system,
Train control system,
Level crossing system.

Final model purposes a complete class diagram which introduced some interactions between:
Level crossing control system and physical equipment (barrier, traffic light, train sensors)
Train control system and physical equipment (train sensor),
Level crossing control system and communication,
Train control system and communication,
Operation center and communication.

Statechart diagrams, also referred to as State diagrams, are used to document the various modes ("state") that a class can go through, and the events that cause a state transition. The state-transitions graph formalism is not a UML innovation. It has often been employed in other contexts and a large consensus, from David Harel’s works, exists around this notation. It introduces the description of possible sequences of states or actions which can occur to an element during its life. Such sequences arise from element reaction to discrete events.

We coded all properties in UML by using OCL constraints attached to classes or sets of associations to specify safety and operational invariants of reactive systems in a concise manner.

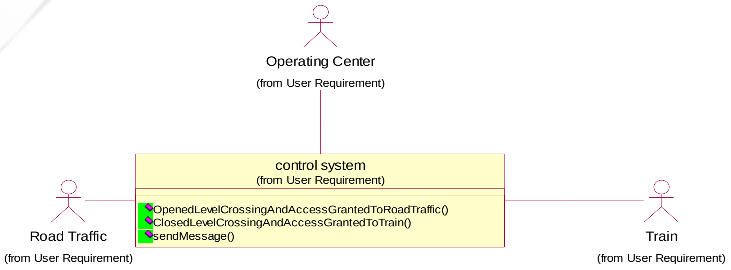

Figure. 8: First Class diagram

4 CONCLUSIONS

The main difficulty to specify railway case study is the less of harmonisation between the different European systems. The level crossing modelling presented here gives a first step to a computerised management of level crossing. In this paper, we purpose a method for modelling a safety railways application. But the precondition to use UML diagrams for system specification, which is usable for formal correctness proofs and refutation checks, is that the UML has to be used with a precise semantics. This is possible by definitions of translation rules for the conversion of UML notation in a formal language. Our global project (see Jean-Louis Boulanger, Philippe Bon et Georges Mariano (2004)) purposes to transform a semi formal modelling (UML model) to a formal specification (B method).

REFERENCES

Abrial, JR. (1996). ”The B Book - Assigning Programs to Meanings”. Cambridge University Press, August 1996.

Jean-Louis Boulanger, Philippe Bon et Georges Mariano (2004) ”From UML to B - a level crossing case study”, COMPRAIL 2004, 17-19 May 2004, Dresden Germany.

EN 50126, (1999).”Railways Application – The specification and demonstration of Reliabilty, Availibility, Maintenabiliy and Safety (RAMS)”, 1999.

EN 50128, (2001).”Railways Application – Communication, signaling and processing systems – Software for railway control and protection systems”, 2001.

prEN 50129, (2000).”Railways Application – Safety related electronic systems for signaling”, 2000.

Einer, S.; Schrom, H.; Slovák, R.; Schnieder, E. (2002) ”A railway demonstrator model for experimental investigation of integrated specification techniques”, In: Ehrig, H.; Grosse-Rhode, M., Hrsg.: ETAPS 2002 - Integration of Software Specification Techniques, S. 84-93, TU Berlin, DFG, Grenoble 2002.

Jansen, L. and Schneider, E. (2000) « Traffic Control Systems Case Study: Problem Description and a Note on Domain-Based Software Specification », Institute of Control and Automation Engineering, Technical UNIVERSITY of Braunschweig, 2000.