Software Hazard Analysis for X-By-Wire Applications

Erendira Ibarra-Alvarado

Dept. of Engineering and Design, University of Sussex, Falmer Brighton, BN1 9QT, U.K.
eii20@sussex.ac.uk

Abstract. This paper presents a comprehensive safety approach to the development process of automotive software systems focusing on X-by-Wire applications. A modification to the traditional V-model Development Process is proposed as well as the use of the Rapid Object-Oriented Development Process for Embedded Systems. The system and controls are modelled in UML and then, as part of the safety assurance, can be analysed using hazard analysis techniques such as HAZOP. This paper describes the use of HAZOP as part of a systematic approach to develop complex software for embedded systems in safety-critical applications.

1 Introduction

There is an increasing trend towards having computer controlled functions in vehicles. Software is moving from an auxiliary to a primary role in providing critical services. Object-Oriented technologies have been present as a powerful solution to complex IT problems for some time; conversely this is quite a new approach for automotive embedded systems [1]. Current solutions using model based design employ graphical tools such as ASCET-SD or Matlab/Simulink. These tools are not standardised as they change with every version released, leaving the designers facing new problems. One factor in ensuring safety is the development process, it can promote a more systematic approach to tackle the complexities in the development of software for embedded systems, as well as produce a system of consistent quality.

In our research we are proposing a change to the traditional development process for automotive systems by integrating explicit hazard identification procedures, supported by the ROPES (Rapid Object-Oriented Process for Embedded Systems) which uses the standard UML (Unified Modeling Language) meta-model for its semantic framework and notation [2]. The UML models produced will be examined with adapted hazard analysis techniques, which in turn, will be reformulated for software use. The example applications we are proposing for this development framework are XbW (X-by-Wire) systems and in particular SbW (Steer-by-Wire).

Among the existing hazard analysis techniques we have HAZOP (Hazard and Operability) Study. The purpose of HAZOP is to identify what potentially hazardous variations from the design intent could occur in components and in the interactions between components of a system. A HAZOP study is performed by a team of people with different expertise that should be reflected in the safety analysis of the project, supported by the use of guidewords to conduct the analysis.

Hazard analysis is a process that produces a documentary record of faults, resulting hazards and the hazard control measures.

1.1 Aims

Based on the problems recently observed regarding software in vehicles [3], current software development practices need to be adapted to give a competitive answer to the demands in the automotive market.

This work explores the use of HAZOP on UML diagrams used in the analysis and design phases of the development lifecycle.

HAZOP was originally intended to find hazards arising from the interaction of components in chemical plants, mainly by analysing flows of substances. Our approach is to analyse the interaction of diagram elements, examining message flows, represented by relationships defined by the UML standard.

Our main aim is the explicit incorporation of hazard analysis in the development process itself. This allows us to have a more systematic approach to tackle the complexities in the development of software for embedded systems, as well as produce a system of consistent quality.

The objective of this research is to develop a set of guidelines to perform hazard analysis where models of both system and controls would be developed in UML following the ROPES process. By using UML we establish a common language to be used by the different designers involved in the project. This is due to the multidisciplinary nature of automotive systems, where engineers with very different backgrounds are going to be contributing to the development of these systems.

2 Safety and the Development Process

In the automotive domain, the traditional V-model Development Process for electronic and software systems, does not explicitly incorporate analysis for safety in its phases; in our proposed development process we use the traditional V-model [4] as a basis, but we add a branch to incorporate safety procedures as a fundamental part of the whole system development see Fig. 1.

The objective of this addition is to make sure that the safety assessment is consistent throughout the development lifecycle.

Since the V-model is of generic nature, we use of a Software Development Process in parallel to design the architecture of the XbW system and controls. This is the means to incorporate safety processes to the traditional V- model.

Together with the development process, a model based design using systematic refinement including early validation by simulation and hazard analysis, can mitigate the risks of error-prone software developed using traditional methods (code and fix for example).

Our technique applies the HAZOP guidewords with suitable interpretations to elements of the UML diagrams, at this initial stage UCD (Use Case Diagram) and CD (Class Diagram). The elements incorporated are those that could be subject to deviations leading to meaningful results. This systematic use of HAZOP applied to the UML diagrams has as main goals to explore, elicit and improve safety aspects.

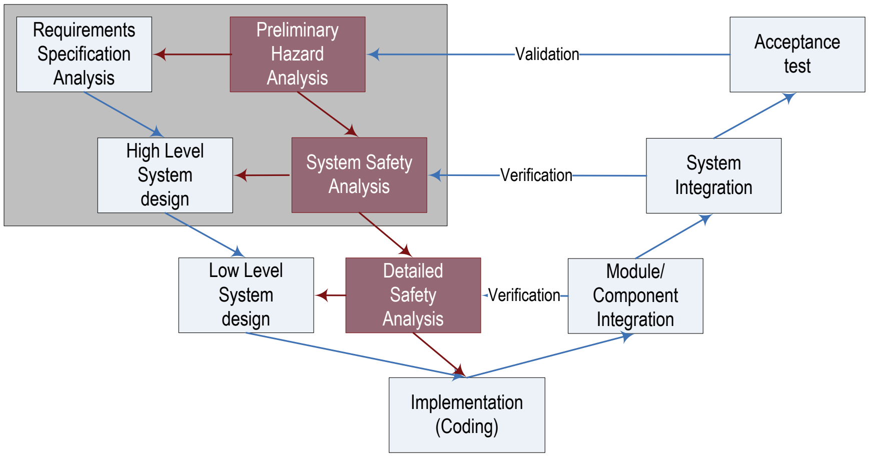

Fig. 1. Modified V-model Development Process, incorporating safety procedures

3 SbW system model and hazard analysis

From our point of view, the two most important diagrams to achieve a system model comparable to the system itself are the UCD and the CD. At the highest level of abstraction, the UCD captures and groups the requirements and the master CD maps these requirements to represent the interactions between subsystems without differentiating hardware and software. As we move from higher to lower levels of design, the CD becomes more detailed, and other diagram types (such as statecharts and sequence diagrams) may be introduced.

3.1 Safety Lifecycle

The first phase of the safety lifecycle starts with a preliminary hazard analysis, which is performed on the UCD and CD to confirm that the system requirements are met.

As the traditional context for a HAZOP analysis is the chemical industry, in the DEF STAN 00-58, the guidewords were modified to be suitable for the analysis of programmable electronics systems. See [5] for a comparison of guidewords. The general approach of our work is to use these guidewords with a very clear interpretation for each attribute under analysis, some examples in Table 1.

The initial phase of the hazard analysis starts by taking the UCD and the master CD to perform a HAZOP study on the requirements specifications. During the analysis phase of any project defects can be caused by poor understanding of the specifications. The master CD is analysed to verify that it can fulfil the requirements. By recognising that the design does not satisfy the specifications, we are trying to make sure that defects are being fixed. A "*defect is the evidence of the existence of a fault"* [6]. Two main things are evaluated with this strategy the first is how close the model is to the requirements and the second how well the requirements are expressed.

Table 1. Example of guideword interpretation for Requirements

Element: Requirement

Guideword	Interpretation
NO	complete failure to fulfil the requirement
MORE	a parameter related to the requirement is larger than normal.

The focus on the diagrams changes accordingly to the detail needed in the development lifecycle, e.g. in the *low level system design* phase we plan to have a much more detailed CD and statecharts. We also have to consider suitable hazard analysis techniques to match the type of information provided by each diagram.

4 The HAZOP Study

Having produced an initial design a HAZOP study was then conducted. The HAZOP team consisted of: The Chairman, experts in the areas of automotive systems and software, the system designer and a secretary. The highlighted area in Fig. 1. represents the lifecycle phase where this HAZOP study fits, since at least in this application the analysis and design phases are very much interlinked, so distinguishing between the two is not straightforward in practice. This is mainly because once the master CD has been drawn some degree of design is implicit in it, especially regarding to hardware. The preliminary hazard analysis was carried out implicitly in the requirements specification analysis. The HAZOP study delivered valuable insights, such as taking account of particular safety and operability issues, e.g. potential improvements to investigate and additional features to include in future designs, information which could ultimately be fed back to the requirements analysis phase. These feedback paths make difficult to perform the study in the analysis stage only, leading the HAZOP study to be performed on both, analysis and design stages simultaneously.

During the HAZOP analysis we focused on the master CD, using the requirements list as the basis to determine if the information there was mapped adequately to the master CD. Two things are important to notice here:

- A HAZOP study is not a design review. The master CD must be mature enough to be used in the HAZOP, making sure that it represents the designer's most refined representation of the system. Although informal design reviews were carried out between some members of the HAZOP team, a different perspective was brought up in the HAZOP meeting, such as a design change that needed to be considered for safety reasons, e.g. supplying additional information in the master CD such as redundancy for sensors and actuators.

- The master CD at this stage represents a combination of hardware and software, and the hardware at least needs to be represented as accurately as possible, since it constitutes a major design decision even at this early stage.

The changes to the master CD, that were suggested actions in the HAZOP study, were mainly adding either classes and their respective relationships, or just relationships between some classes. Other actions that were not reflected as changes in the master CD, were mainly about time management, such as delays introduced by the communications network, however this is out of the research scope; special road conditions that needed to be considered as part of the mechanisms to deal with disturbances; how to cope with faults in sensors and actuators making explicit the incorporation of redundancy; faults in the on board diagnosis system that could lead to hazardous conditions and finally measures to protect the system from user modifications.

We are now in search of some metrics to evaluate the effectiveness of the HAZOP analysis. However our expectations are to find most of the hazards when we have a more detailed version of the CD.

4.1 Relation to other Work

This paper is based on a combination of the fundamentals first explored in [7], [8] and [5] as a start point to work with UML and HAZOP together.

The idea to integrate hazard analysis into the development lifecycle has been looked at before by [7], where they propose an individual analysis first and at a later stage a team meeting, instead of a pure team approach for HAZOP analysis. They also propose to change the traditional HAZOP guidewords. Nevertheless we think that the guidewords main objective is to elicit discussion, so the relevant strategy is the interpretation of these guidewords when applied to attributes.

In [8] they provide a set of guidelines to perform HAZOP on UML models, focusing on software architecture. The application of this technique is considered for requirements analysis; however we suggest a number of changes in this matter, as they only consider a subset of UML elements. One of our aims is to produce a standard set of guideword-interpretations to perform HAZOP analysis for each type of UML diagram, based on the definitions given in the specification of UML 2.0 [9]. This step has been completed for the UCD and CD, but Remains to be done for the other diagrams.

5 Future Work

In the intermediate phase of the safety analysis, we intend to use the CD to verify its internal functionality, for this purpose a second HAZOP study will be performed.

HAZOP appears to be a useful technique for the initial phases of the safety lifecycle, due to its inherent level of abstraction. Nonetheless, in a later phase a more detailed safety analysis will be performed using a different technique, such as FMEA (Failure Modes and Effects Analysis) expecting that it will allow quantifying the probability of failures.

The more detailed safety analysis still has to be completed along with the design of the SbW system, the code produced has to be deployed to the target hardware and then the testing stage can be achieved. It is out of the project scope to build a complete SbW system in a vehicle; implementing a prototype of the system is feasible, where a comprehensive set of disturbances can be simulated and the software can be running in appropriate target hardware.

It is important to measure the impact of the hazard analysis in each phase of the development lifecycle, in order to put together a framework to work with safety-related embedded systems, and come out with the set of guidelines to design this type of systems. This aspect will be analysed in more detail as well as incorporate real data from manufacturers' studies.

6 Acknowledgments

I would like to thank my supervisor Prof. Richard K. Stobart[1], Dr. Rudi Lutz[2] and Dr. Michael English[3] for their valuable insights and support to develop this research.

[1]Department of Engineering and Design, University of Sussex, Falmer Brighton, BN1 9QT, U.K., R.K.Stobart@sussex.ac.uk

[2]Department of Informatics, University of Sussex, Falmer Brighton, BN1 9QH, U.K., rudil@sussex.ac.uk

[3]Independent Safety Engineering Consultant, Chairman of the HAZOP Meeting, menglish@menglish.force9.co.uk

References

1. Grimm, K. Software Technology in an Automotive Company - Major Challenges. in 25th ICSE. 2003. Portland Oregon.
2. Douglass, B., ROPES. 1999, I-Logix Whitepaper.
3. Furst, S. Autocoding in Automotive Software Development, Qualification Aspects Of ACGs. in Automotive Electronics Conference. 2005. London.
4. Benz S., D., E., Dieterle, W., Muller-Glasser, K. A Design Methodology for Safety Relevant Automotive Electronic Systems. in SAE SP1870. 2004. USA.
5. UK Ministry of Defence, Defence Standard 00-58: HAZOP Studies on systems containing programmable electronics. 2000. (2 Parts 1 and 2).
6. Conte, S.D., Dunsmore, H.E., Shen, V.Y., Software Engineering, Metrics and Models. 1986, USA: Benjamin/Cummings.
7. McDermind, J.A., Nicholson, M., Punfrey, D.J. and Fenelon, P. Experience with the application of HAZOP to computer-based systems. in COMPASS '95. 1995: IEEE.
8. Hansen, M., Wells, L. and Maier, T.: HAZOP Analysis of UML-Based Software Architecture Descriptions of Safety-Critical Systems. in 2nd NWUML. 2004. Finland.
9. Rumbaugh, J., Jacobson, I. and Booch, G., The Unified Modeling Language, Reference Manual. Second ed. 2004, USA: Addison-Wesley.