Requirement traceability in safety critical systems

Marie-Agnès Peraldi-Frati
ISS laboratory
CNRS, University of Nice Sophia-Antipolis, INRIA
Sophia-Antipolis, France
map@unice.fr

Arnaud Albinet
Advance Development Electronics of Powertrain
Continental Automotive SAS
Toulouse, France
arnaud.albinet@continental-corporation.com

ABSTRACT
Safety engineering analysis is a mandatory stage in the design of critical embedded automotive systems. The derivation of safety requirements and their verification require establishing traceability links between requirements and the different artifacts involved in the design flow. This paper presents the different steps of a method for expressing non functional requirements (safety, timing, hardware, performance) and ensuring their validation and their traceability over a design flow for automotive system design based on the conjoint use of EAST-ADL2 and MARTE languages and supported in an Eclipse platform. A specific meta-model for requirements modeling and traceability is used. The methodology is illustrated on an industrial knock-control system characterized by strict safety and temporal constraints.

Keywords
Safety, Non-functional requirement, Time, modeling, Traceability, Validation, MARTE.

1. INTRODUCTION
Nowadays, automotive systems are safety critical and their development process must comply with modern certification/safety standards such as the ISO-26262 [1]. During safety engineering analysis, performed at the early stages of the design, safety engineers establish the possible failures of the system and their origins and needs on design architecture. During the requirement development stages, a requirement phase elicits multiple requirements potentially of different types (functional, non functional –safety, timing, hardware, performance …), including those expressed by the safety engineers. A model based development phase follows, characterized by specification, design and implementation steps which take into account the overall requirements over different disciplines (system, software, hardware). The verification and validation phase checks whether or not models developed and the final product complies with the initial requirements.

In this context, requirement modeling, traceability and analysis is a key issue in a design flow for electronic embedded systems [2].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CARS’2010, April 27, Valencia, Spain
Copyright © 2010 ACM 978-1-60558-915-2/10/04... $10.00

Quoting Gotel and Finkelstein in [3] “the requirement traceability is the ability to describe and follow the life of a requirement through its development, specification, validation and verification”. Industrials from IT have proposed and developed standards and engineering tools which partially cover these needs as they focus on functional requirements. Moreover the relationship between the initial expression of requirements and their impact on solution models is not fully established. In particular, the traceability of non functional requirements raises some others automotive specific issues dedicated to reuse and maintenance, such as the way to express and distinguish a safety or a timing requirement in a set of requirements; the ability for models to express temporal and safety properties, the link with validation tools for the test or the analysis of models and finally, the feedback of the analysis results on the design flow.

This paper presents an industrial example of a knock control system characterized by various and stringent safety and temporal constraints. This example is the support for illustrating a tool methodology based on Eclipse that uses EAST-ADL2 [4] MARTE [5] and SysML [6] for modeling embedded systems from initial requirements in a specification document down to the model and the final product. Traceability links interconnect these elements and the associated tools for their verification and their validation.

The paper is structured as follows: in section 2, we introduce the main issues for traceability in real-time embedded systems and the benefits of introducing traceability concepts for managing safety. Section 3 presents an industrial case study for illustrating our methodology. The traceability main concepts are presented in Section 4. Section 5 deals with validation and backward annotation on traceability links. Section 6 gives a conclusion.

2. TRACEABILITY AND SAFETY CRITICAL SYSTEMS.

2.1 Requirements coming from safety analysis
Despite a lot of efforts, requirement management and traceability still remains a challenging problem in the automotive industry. Automotive applications design process should comply with safety standards (ISO 26262) and customer expectations which impose vertical and horizontal bi-directional traceability of requirements:

- Vertical traceability identifies the origin of items through the work breakdown structure
- Horizontal traceability identifies requirements relationships across workgroups or components
The requirement development process showed on Figure 1 illustrates the strong link between stakeholder’s requests, in particular safety analysis, and the requirement specification document. Currently, safety requirements are often not classified and generally not formally traced to safety analysis results.

In the automotive domain, safety analysis is decomposed in a failure mode and effect analysis (FMEA) and a fault tree analysis (FTA). A FMEA is a procedure for analyzing and classifying by severity degree the potential failure modes within a system, their effects and their causes. A failure mode can be any errors or defects especially those that affect the customer. Effects analysis refers to Fault Tree Analysis (FTA) in which an undesired state of a system is analyzed using boolean logic to combine a series of lower-level events. The FMEA and the FTA are under the responsibility of safety engineers which express in a safety document, needs and expectations on the system.

This documentation is collected as stakeholder requests during the requirement development process and the extracted requirements are identified and elicited by the requirement engineer and then formalized in the requirement specification document. The requirement engineer can trace the requirement from the request by using inter-document traceability tools or within an integrated framework.

2.2 Traceability of requirements
Requirements derived from a safety analysis process can impact the functional and/or the hardware architecture of the system to be designed. Multiple functional and non-functional requirements (timing, consumption, hardware, dependability) derive from such analysis. These requirements are tackled at various levels (analysis design implementation …) of a design flow that merges different tools dedicated to architecture modeling, code generation and validation and verification.

Provision must therefore be made to connect initial requirements to all system components in the design flow and this, in both backward and forward directions. Indeed, the design flow includes heterogeneous models and tools.

The validation and verification flow integrates dedicated tools for model and product testing, model verification and validation by temporal analysis and simulation.

In this context, performing a full and bi-directional traceability of requirements and system solutions requires developing a new model for traceability that takes into account all the characteristics of multi-levels modeling, requirements classification, and model heterogeneity. A survey on traceability techniques [2] has showed that above the 17 techniques evaluated, none of them fully satisfies the needs for real-time critical systems.

2.3 Methodology for traceability
The methodology illustrated in this paper –developed in the context of two research projects [7][12]–, handles the full traceability links from the initial customer requests (i.e. the requirement specification document) down to their modeling and their implementation.

This work ensures, at each abstraction level of the system design process, from functional analysis down to software implementation, a relation between requirements and the design model. This method targets on running backward and forward analysis during a change request, either on requirement or on solution (code or design specifications). The tool chain, based on Eclipse, ensures and identifies which elements on the solution is impacted by a requirement change and needed to be updated or verified again. ISO 26262 recommends the full requirement traceability, meaning from request to solution and test. The methodology answers to this request and propose a complementary usage of such tool for industrials using massively product line approach.

3. THE KNOCK CONTROL SYSTEM
3.1 Basic principles
On an ignition engine, the combustion process is produced by an electric spark generated at precise time by a spark plug in the combustion chamber. The start of this combustion is electronically-controlled. The knock is a physical phenomenon that leads to an auto ignition of the fuel/gas mixture and increases the compression in the chamber. Such misfiring may leads to combustion misses, lower energy performances, and higher fuel consumption or, in the worst case, the damage or destruction of the catalytic converter. Thus, a consistent and reliable ignition is essential.

In a stroke engine, valve timing is regulated by a camshaft, driven by the crankshaft rotation. The moment of the ignition always refers to a piston position measured by crankshaft degree i.e. the top dead center (TDC). Correcting the knock consists in applying a retard or an advance for the plug firing.

3.2 Safety analysis phase
This phase identifies and classifies safety hazards for the knock i.e. the system conditions or the states that can result in a mishap.

Table 1 is an example of information resulting from a FMEA in an Ignition control system, in which Effect and Cause of possible failures are identified. To prevent the different causes, a more detailed analysis identifies protective measures and derives system safety functions to be implemented. All this information is stored in safety system documentation.

1 MeMVaTEX project www.memvatex.org
2 EDONA project www.edona.fr
Table 1. Safety hazard example

<table>
<thead>
<tr>
<th>Function</th>
<th>Ignition Control System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Normal</td>
</tr>
<tr>
<td>Failure mode</td>
<td>Shockwaves in the cylinders</td>
</tr>
<tr>
<td>Effect</td>
<td>Cylinder destruction</td>
</tr>
<tr>
<td>Severity</td>
<td>10</td>
</tr>
<tr>
<td>Cause 1</td>
<td>No instantaneous spark delay correction</td>
</tr>
<tr>
<td>Cause 2</td>
<td>Wrong spark delay correction</td>
</tr>
</tbody>
</table>

3.3 Requirements development phase

During the requirement elicitation, the safety system document lists the stakeholder requests. For example, the requirement engineer identifies the safety requests associated with Cause 1 and Cause 2 of Table 1. The analysis requirements phase derives from these requests the requirement N°9 of Table 2 and its classification as a non-functional and safety requirement. Partial results of the requirement development phase for the knock are given in Table 2.

A requirement meta-model that extends SysML allows a classification of functional and non-functional requirements, the integration of the design architecture level concerned, a safety level and a precise link with validation and verification tools. By this way, in a set of requirements, a non functional requirement can be easily spotted and traced.

Table 2: Functional and non functional requirements for the Knock Control System

<table>
<thead>
<tr>
<th>N</th>
<th>Class.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>VLF</td>
<td>The system shall detect every knock.</td>
</tr>
<tr>
<td>7</td>
<td>VL-NF-V</td>
<td>The engine is composed of four strokes.</td>
</tr>
<tr>
<td>9</td>
<td>AL-NF-S</td>
<td>If a knock event is detected, an instantaneous Spark delay correction corresponding to this knock event shall be ensured.</td>
</tr>
<tr>
<td>10</td>
<td>AL-F</td>
<td>The acquisition of the knock signal shall be performed in 50 ms</td>
</tr>
<tr>
<td>11</td>
<td>AL-NF-S</td>
<td>Knock control on each cylinder on cycle n shall be performed before engine cycle n+2.</td>
</tr>
<tr>
<td>12</td>
<td>AL-F</td>
<td>A total spark retard shall be calculated by the correction sub function in case of knock detected</td>
</tr>
<tr>
<td>13</td>
<td>AL-F</td>
<td>The knock correction value should be within the interval [-15°,+15°] CRK .</td>
</tr>
<tr>
<td>15</td>
<td>DL-NF-P</td>
<td>The knock correction shall be computed each 720° measured on the crankshaft (CRK)</td>
</tr>
</tbody>
</table>

Requirements are identified by a number which is part of a unique identifier for requirement traceability. The second column provides clues for classification by indicating: the abstraction level of the design (VL, AL, DL, etc.), the type (Functional or Non Functional), and for non functional requirements a classification (Performance, Variability, Safety, etc.). A description of the requirement is given as a simple text. Other information contains by a requirement model element concerns the traceability management and the verification and validation analysis. These properties are represented on Figure 2.

The verificationType field indicates the type of procedure that could be used for the verification of a requirement (code inspection, test-case, OCL expressions checking or real-time analysis). The result of such verification applied on the design model or on the final product is then propagated and impacts the property’s value of the initial requirement (satisfyStatus and verifyStatus). This reporting becomes possible with the traceability features that interconnect the requirements at the different levels. These features are presented in section 5.

4. TRACEABILITY OF REQUIREMENTS

4.1 Traceability between requirements

4.2 Traceability in the design

All requirements must be satisfied during the design phase by either a structural or a behavior model. The satisfy relationship is used to point out in the solution model which model element satisfies a given requirement.
Figure 3 illustrates the satisfy relationship between an activity diagram of the knock controller model and the requirement AL_NF_P_15 coming from the requirement model of Figure 2. A timedDurationConstraint imposed to the activity diagram, models the temporal requirement with the MARTE syntax. This choice allows a formal specification of time and it is of particular interest in this example where temporal constraints are linked to a crankshaft time base. In this case the maximum duration of the knock must be equal to 720° measured on crkClk.

The verify relation links requirements to validation and verification elements, here a test case. This test case describes a procedure for the scheduling analysis of the temporal constraints imposed to the design. Both structural and temporal parameters can be extracted from the design model (period, deadline, WCET) and are used as input for the validation tools TIMESQUARE [9].

5. IMPACT OF VALIDATION ON TRACEABILITY

The final step of the approach aims at reporting on the traceability links status, results of the test cases procedures. Traceability links are endowed with status that changes after the completion of verification procedures. If all V&V procedures linked to a requirement are conclusive, then the status of verify and satisfy links are changed from pending to verified or satisfied. When all the links statuses are positioned, the properties statuses of the requirement inherit from the result (boolean expression calculus) of the link statuses. For each requirement it is then possible to checks if the design takes into account the requirement (satisfy relation) and if such design is correct (verify relation). Such information can be used in a second step for verifying safety aspects. If all requirements in relation to the safety analysis are fulfilled, the design can be considered has safe itself.

6. CONCLUSION

Traceability management has been identified as an important issue in safety critical systems. In automotive, the new standard for safety imposes a full traceability of requirements, their modeling and validation. As a consequence, requirements should be linked in a bi-directional way, with the design model and the validation activities. This paper illustrates the contribution of traceability in requirement modeling and validation and the impact on safety critical system design. Specifics meta-models for requirement are used that covers the needs for traceability and safety critical system modeling, and validation tools. We present a methodology illustrated on an industrial knock-control system characterized by strict safety and temporal constraints.

REFERENCES
