Meta Modelling in the Vehicle Industry

SARA SADEGHI

Stockholm
Abstract

The advance of electronics and information technology during the last years has made possible the proliferation of embedded systems in all fields. Accordingly, embedded systems have become increasingly available in all automotive products. These systems bring about improvements in functionality, increase of system complexity, and more interaction between hardware and software components. The development of these systems requires that engineers from multi-disciplinary fields cooperate closely in order to efficiently develop such complex products. However, there is often disagreements among these engineers about design concepts such as requirements, functions, and specifications. Despite that there have been various attempts of providing information models for automotive embedded system design, there is in practice a lack of a consistent structure that represents and describes the relationships between these concepts. Moreover, such structure has never been implemented in an industrial modeling language for complex physical systems such as Modelica. The objectives of this thesis are to provide a multi-level structure, which represents different design abstraction levels, and a meta-model for automotive embedded system design. This thesis was done at Scania, Södertälje, where these models were used for designing a fuel level display embedded system for a truck. The multi-level structure was designed and developed using a real case, fuel display system, from high-level abstraction (customer requirements) down to component/block level specifications. Afterwards, a meta-model was proposed. The proposed meta-model was evaluated based on nine interviews with experts in information modeling and development area from both industry and academia. The following criteria were considered for the evaluation of meta-model: correctness, comprehensibility, expressiveness, generality, and usefulness. Six experts confirmed that the proposed meta-model was correct while two experts commented that in general, models could not be said to be correct or incorrect. One of the experts considered that the model required more details. In addition, the model was comprehensible for the majority of the experts. Discussions regarding semantics and expressiveness resulted in some model refinements. Afterwards, the experts ac-
knowledged the expressiveness of this meta-model. The experts agreed that this meta-model was general for automotive system design, and six of them confirmed that it was useful. The rest recommended that the meta-model should be developed in order to understand its usefulness.
Acknowledgements

Firstly, I would like to thank Prof. Paul Johannesson for accepting to examine my thesis and providing me valuable feedback.

I am very thankful to Assoc. Prof. Mattias Nyberg for providing me the opportunity to work at Scania in his research group, his advices, and supervision during my thesis.

I would like to thank my supervisor at KTH Lic. Fredrik Asplund who guided me for thesis definition, academic writing, and scheduling. Fredrik is a very supportive, strict, and organized supervisor.

Furthermore, I would like to express my gratitude to the Mechatronics head Prof. Martin Törngren for his great support. I am very thankful to him for providing resources for interviews.

Many thanks to Jonas Westman for his valuable comments during meetings, and his guides and feedbacks.

Thanks to Dr. Olena Rogovchenko from Liköping university and her comments, encouragement, and support during the meetings that we had.

Thanks to my group-mate Feng Liang from Liköping university for being a very good colleague.

Moreover, I would like to acknowledge the experts that participated in the interviews, and meetings and collaborated with their guidance. I thank Thomas Näreskog, Jonas Biteus, and Lic. Gunnar Berg from Scania, Dr. Dag Bergsjö from Chalmers University, Jan Soderberg from Systemite, Lic. Mattihas Biehl from KTH Royal Institute of Technology, and Dr. Niklas Adamsson from Accenture. In addition, I want to thank Ola Ramström, and Jonas Edén from Scania for the meetings regarding functional architecture.

I want to thank Jon Andersson, Head of REPA at Scania, for his encouragement and support. He inspired me a lot to start to communicate in Swedish with my very
little Swedish knowledge.

Last, but not least, my special thanks to my husband, Dr. Saul Rodriguez, and my gratitude to my family in Iran for their eternal support.
Contents

Abstract \hspace{16em} i
Acknowledgements \hspace{13em} iii
List of Figures \hspace{12em} viii
List of Tables \hspace{12em} x

ABBREVIATIONS AND ACRONYMS \hspace{11em} xii

1 Introduction \hspace{1em} 1
1.1 Problem \hspace{9em} 2
1.2 Research question \hspace{8em} 2
1.2.1 Contributions and Target Groups \hspace{7em} 3
1.3 Scope and Limitation \hspace{8em} 3
1.4 Structure of Report \hspace{8em} 4

2 Method \hspace{1em} 5
2.1 Choice of Method \hspace{9em} 5
2.2 Application of Method \hspace{8em} 6

3 Background \hspace{1em} 9
3.1 Models \hspace{9em} 9
3.2 Model Based Development \hspace{8em} 10
3.2.1 Means \hspace{8em} 10
3.3 Requirement Classification \hspace{8em} 11
3.4 Technology \hspace{8em} 12
3.4.1 SysML \hspace{7em} 12
3.4.2 Modelica Language \hspace{7em} 13
3.5 Meta-Modeling \hspace{8em} 14
3.5.1 Properties ........................................... 16
3.6 Perspectives ......................................... 17
3.6.1 Abstractions and Perspectives Relations ............. 18

4 Scania ..................................................... 19
4.1 SESAM .................................................. 19
  4.1.1 Electronic Control Unit System .................... 20
  4.1.2 Functional Architecture (Fuel Level Display) ........ 20
4.2 Different Types of Document ............................ 22

5 A Meta-Model for Automotive Embedded System Design 23
5.1 User Perspective ..................................... 23
5.2 Logical Perspective .................................. 24
5.3 Technical Perspective .................................. 25
5.4 Requirement Perspective ............................... 25

6 Multi-level structure for Fuel Level Display 29
6.1 About Fuel Level Display System ...................... 29
6.2 Requirement Selection ................................ 30
  6.2.1 User Function Requirement 18 .................... 31
6.3 System Design ........................................ 32
  6.3.1 Requirement Perspective ........................ 32
  6.3.2 User and Logical Perspective ...................... 33
  6.3.3 Technical Perspective ............................ 35
  6.3.4 Fuel Level Display Structure ...................... 36
  6.3.5 Verification Design ............................... 38
  6.3.6 Multi-Level Structure in Modelica ................. 38
6.4 Simulation Result ..................................... 47

7 Evaluation .............................................. 51
7.1 General Information .................................. 51
  7.1.1 Target group ...................................... 51
  7.1.2 Usage and Importance of Meta-modeling ............ 52
7.2 Analysis ............................................... 54
  7.2.1 Correctness ....................................... 54
  7.2.2 Comprehensibility ................................ 54
  7.2.3 Expressiveness .................................... 55
  7.2.4 Generality ....................................... 55
  7.2.5 Usefulness ....................................... 56
7.3 General Comments ..................................... 56

8 Conclusions and Future Work 57
8.1 Conclusions .......................................... 57
8.2 Future Work .......................................... 58
List of Figures

3.1 Meta Object Facility .............................................. 15
4.1 ECU Configuration (Westman, 2003) ............................. 21
4.2 Dashboard of A Truck ............................................. 22
5.2 Meta-Model: User Perspective ..................................... 24
5.3 Meta-Model: Logical Perspective ................................. 24
5.4 Meta Model - Technical Perspective ............................. 25
5.1 Meta-Model for Fuel Level Display system Design .......... 27
6.1 Fuel level Display in Dashboard ................................. 30
6.2 Fuel level Display System ........................................ 30
6.3 Requirement Diagram ............................................. 33
6.4 Block Definition Diagram - Logical and User Perspectives . 34
6.5 Internal Block Diagram - High Level Functions of Logical Perspective . 34
6.6 Internal Block Diagram - Refined Logical Perspective .......... 35
6.7 Block Definition Diagram - Technical Perspective ............. 35
6.8 Internal Block Diagram - Technical Perspective ............... 36
6.9 Block Diagram - Fuel Level Display Structure ................. 37
6.10 verification and Requirement Top Level ......................... 38
6.12 Perspectives breakdown in Modelica .......................... 39
6.13 Perspectives breakdown in Modelica - SysML Structure ....... 40
6.14 Fuel Level Display - level 1 .................................... 41
6.15 Fuel Level Display - User Function 18 .......................... 42
6.16 Fuel Level Display - Verification ................................ 42
6.17 Fuel Level Display - Performance verification ................. 43
6.18 Fuel Level Display - Functional verification ..................... 43
6.19 Fuel Level Display - level 2 - SESAM .......................... 44
6.20 Fuel Level Display - level 3 - COO-ECU System ............... 45
6.21 Fuel Level Display - level 4 - COO-ECU ......................... 46
List of Figures

6.22 Fuel Level Display - AER201 ............................................. 46
6.23 Fuel Level Display - AER202 ............................................. 47
6.24 Simulation Result .......................................................... 48
6.11 Fuel Level Display Structure Multi Level Structure .................. 49

7.1 Area of Expertise ............................................................. 52
7.2 work experience(years) ...................................................... 52
7.3 Meta-Model Usage ........................................................... 53

C.1 Requirement Diagram ....................................................... 72
C.2 Block Definition Diagram - Logical and User Perspectives .......... 73
C.3 Internal Block Diagram - High Level Functions of Logical Perspective(Up), and Refined Logical Perspective(Down) ......................... 74
C.4 Block Definition Diagram - Technical Perspective .................. 75
C.5 Internal Block Diagram - Technical Perspective .................. 76
C.6 FLD Packages ................................................................. 77
C.7 General Model For Verification .......................................... 78
C.8 Verification Package ........................................................ 78
List of Tables

3.1 Requirement Relationships ........................................... 14
3.2 Design Perspectives (Baumgart et al., 2010) .......................... 18

6.1 User Function Requirement 18- Fuel Level Display (Erlandsson, 2011) . 31
6.2 AE201 Requirements- Estimate Fuel Level (Wallebäck, 2010a) .......... 31
6.3 AE201 Requirements- Estimate Fuel Level (Wallebäck, 2010b) .......... 32
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFU</td>
<td>Abstract function units</td>
</tr>
<tr>
<td>ADL</td>
<td>Architecture description Languages</td>
</tr>
<tr>
<td>AE</td>
<td>Allocation Element</td>
</tr>
<tr>
<td>AER</td>
<td>Allocation Element Requirement</td>
</tr>
<tr>
<td>BDD</td>
<td>Block Definition Diagram</td>
</tr>
<tr>
<td>C-App-SW</td>
<td>Application Software in C language</td>
</tr>
<tr>
<td>CAN Bus</td>
<td>Controller Area Network Bus</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>COO-ECU</td>
<td>Coordinator ECU</td>
</tr>
<tr>
<td>CF</td>
<td>Customer Function</td>
</tr>
<tr>
<td>EB-SW</td>
<td>ECU Basic Software</td>
</tr>
<tr>
<td>ECU HW</td>
<td>ECU Hardware</td>
</tr>
<tr>
<td>EMS-ECU</td>
<td>Engine Management System ECU</td>
</tr>
<tr>
<td>FAD</td>
<td>Function Allocation Description</td>
</tr>
<tr>
<td>FLD</td>
<td>Fule Level Display System</td>
</tr>
<tr>
<td>FV</td>
<td>Function Variable</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>HRC</td>
<td>Heterogynous Rich Component</td>
</tr>
<tr>
<td>IBD</td>
<td>Internal Block Diagram</td>
</tr>
<tr>
<td>ICL-ECU</td>
<td>Instrument Cluster ECU System</td>
</tr>
<tr>
<td>MBD</td>
<td>Model-Based Development</td>
</tr>
<tr>
<td>MDD</td>
<td>Model Driven Development</td>
</tr>
<tr>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>SESAM</td>
<td>Scania Electrical System created at the year 2000</td>
</tr>
<tr>
<td>SysML</td>
<td>System Modeling Language</td>
</tr>
<tr>
<td>UF</td>
<td>User Function</td>
</tr>
<tr>
<td>UF18</td>
<td>User Function 18</td>
</tr>
<tr>
<td>UFR</td>
<td>User Function Requirement</td>
</tr>
<tr>
<td>vVDR</td>
<td>Virtual Verification of Designs against Requirements method</td>
</tr>
</tbody>
</table>
Introduction

Nowadays, embedded systems are becoming present in all products. Embedded systems for automotive applications have evolved from a single computer system to distributed network computers in vehicles. They facilitate data communication and coordination among sensors, actuators, and human machine interfaces. Advanced technologies of embedded system have enabled new functionalities in vehicle systems. Already ten years ago, a vehicle provided more than 170 functionalities over 60 Electronic Control Units for controlling and monitoring various tasks (Petersen et al., 2001). These functionalities operate like assistants for drivers. For instance, when a failure happens, they inform drivers about it, and offer solutions. With the improvements in functionalities, the complexity of electronics and embedded software has increased (Graaf et al., 2003).

Coping with the growing complexity of current embedded system products is cumbersome and the cost of maintenance is high. Broy (2006) reported that the cost of electronics and embedded software made 40% of the total cost of an automobile in 2006. In addition, the probability of failures on vehicles due to electronic and information systems are increasing. Knippel and Schulz (2004) reported that 49.2% of car breakdowns in Germany were caused by electronics or electrical failures.

Increasing innovations in vehicles functionalities and more dependencies among software and hardware components require an efficient system development approach (Larses, 2005). The development of such systems, require a group of expert engineers from multi-disciplinary fields such as software, electronics, and mechanics. This group should support the agility, adaptability and flexibility of the work based on market requirements (Krikhaar et al., 2009).

One of the current issues in vehicle embedded system development is the lack of connections among requirements, functional description, and system specification. In addition, the lack of formal communication among different engineers involved in the development is another challenge. These problems have a direct negative influence on both system integration and costs. This project address these issues
and proposes a formal model to resolve them.

1.1 Problem

The increase in product complexity has been accompanied by an increase in the number of components and interconnections among them. The variety of components and usage of them in multi-disciplinary fields require agreements among engineers about product development concepts and their relationships. In practice, there is often no consensus among engineers from multi-disciplinary fields on product development concepts and their relationships. A formal description can enhance the clarity of the specifications, reduce conflicts, improve the information consistency, and assist engineers in the design and development process.

Moreover, there is a high demand in the automotive industry for product certification according to safety standard requirements. This issue is attended in early stages of the design phase, where the requirements need be formalized to the system components, and the components should be designed in order to comply with requirements. Currently, there is a lack of a consistent structure, which represents and describes the relation among requirements, functions, and specification of embedded systems design in Vehicles. In addition, such structure has not been implemented in an industrial modeling language for complex physical systems such as Modelica. Although, there are some investigations in this regard such as EAST-ADL language, and contract-based design, they are still being developed by researchers (Atesst, 2012; Baumgart et al., 2010; East-ADL, 2012; Larses, 2005). A structure which was originated from an industrial practice and inspired by these studies could be very useful.

1.2 Research question

This project has been done around the following questions:

Q1. What is the multi-level structure of a fuel level system design that represents the relation between design artifacts such as requirements, functions, and specifications from high-level of abstraction to the specific components in SysML and Modelica?

Literature study including state-of-the-art, and technical reports about fuel level display were reviewed in order to answer this question. Then, the relationships among requirements, functions, and specifications in different levels of abstraction were designed in SysML and developed in Modelica. Subsequently, validation blocks were designed and added in order to test the model. Finally, the modelica model was validated through simulation.

The Modelica model was a common work that was performed in collaboration with another master thesis student from the Department of Mechanical Engineering
1.3. SCOPE AND LIMITATION

at Linköping University.

Q2. How does a general meta-model for automotive embedded system design look like? The investigation on the previous question and literature study were used for the design of a meta-model. Afterwards, the designed meta-model was evaluated through unstructured interviews with experts. The interviews were based on five criteria: correctness, comprehensibility, expressiveness, generality, and usefulness.

1.2.1 Contributions and Target Groups

The contributions of this thesis are:

- To represent a multi-level structure for fuel level display system design that provides relationships among design artifacts such as requirements, functions, and specification in SysML and Modelica language.

- To provide a conceptual model for automotive embedded system design that represents relationships among design artifacts.

Contribution to industry: This project has been conducted for the research center of Scania, Advanced driver assistant group (REPA). The group aims to accomplish safety standards (ISO 26262) in Scania’s embedded systems. The audiences of this project are the same research group at Scania and other engineers who are active in design and development of embedded system the in automotive domain.

Contribution to Academy: besides this thesis report, the material of this thesis was in part the source of a conference paper accepted to the 9th International Modelica Conference 2012 (Liang et al., 2012) (peer-reviewed) which focuses on modeling of complex physical and cyber-physical systems. Accordingly, both engineers and researchers involved in the modeling field are the audience of this research.

1.3 Scope and Limitation

The scope of this research was to provide a multi-level structure and a meta-model for automotive system design. Due to the master thesis time limitation, this research was limited to study this meta-model in a single case study that consisted in a fuel level display system. The fuel level system is an example of the core technical architecture commonly used in most systems at Scania. Therefore, the results of this study can be applied to many other systems using the same architecture. In addition, the scope of this research, does not include configuration management due to time limitation.
1.4 Structure of Report

The next chapter, chapter 2, contains the research method. The background theories for this project is mentioned in chapter 3. In addition, an overview of Scania architecture and functions is noted in chapter 4. In chapter 5, a meta-model for automotive embedded system design is proposed. In chapter 6, the case study of fuel level display system is explained in SysML and Modelica. Chapter 7 includes the evaluation of meta-model through interviews. Finally, chapter 8 contains the conclusions and future work.
2.1 Choice of Method

Design-science research is a methodology with the purpose of “producing a viable artifact in the form of a construct, a model, a method, or an instantiation” (Hevner et al., 2004). Peffers et al. (2007) investigated design-science research process in prior researches. Consequently, he proposed a mature approach for design science in six steps: first step is problem identification and motivation. In this step, a research problem should be found by understanding the environment of the artifact. The second step is defining the objective of a solution. In this step, requirements for the artifact should be extracted. Subsequently, the problem will be transformed to some requirements. In the third step, the artifact will be designed and developed. The fourth step contains a demonstration of an instance that shows how the problem can be solved. The fifth step is evaluation of the solution. During evaluation phase, the result will be assessed against the requirements. The sixth and the last step includes communication of the solution with experts or researchers in the area.

In addition, research design provides a blueprint for data collection in an experimental research (Bhattacherjee, 2012). Its purpose is to find an answer for a research question or examine a theory. Research design involves several approaches for research and data collection such as interviews, case study, and so on.

Interviews capture a variety of unobservable data, like people’s beliefs, practices, and tacit knowledge about the problem or fact by the means of questionnaires. In comparison to other methods such as case and experimental researches, they are economical for researcher’s time, cost, and effort. Interviews are categorized to structured and unstructured interviews. Structured interviews contain fix questions and answers. Unstructured/open-ended interviews can have fixed questions but interviewee can answer questions in their own words. Unstructured interviews are suitable for deep discussion about the subject with specialists in a field. However, the outcome of interviews are highly dependent on the questions asked and
the interviewer attitude. The analysis of the qualitative data such as text from interviews are done through qualitative analysis techniques. Qualitative analysis includes some strategies such as grounded theory and context analysis. Grounded theory starts from data collection and analysis the data to codes, and concepts in order to obtain a theory. On the other hand, Context analysis (Bhattachetjee, 2012; Schilling, 2006) extracts samples from the text and classifies the texts to positive, negative, or neutral. This kind of analysis assists to understand that the candidate comment is positive, negative, or neutral in general.

Another approach is a case study. The Case study is a method for studying a problem in a limited amount of time in a site (Blessing and Amaresh, 2009). It includes data from a real practice or setting. Case study approach can be applied both for examining a theory (positivist method) or building a theory (interpretive method); therefore, it can be more powerful than its competing methods such as interview (survey research) that is only for theory testing. In addition, because of its ability to collect a rich group of contextual data, it can help to have a more genuine understanding of the problem than other research methods. Bhattacherjee (2012) claims that a single-case design is suitable for theory testing and a multiple-case design is appropriate for establishing a theory. Other investigations, Blessing and Amaresh (2009), expresses that a single case study cannot be used for examining 'casual relationships', but can result very important information. Case studies can be analysed through simulation. Computer simulation applications are powerful tools for studying behaviour of a system design in engineering domain. They utilize mathematical models in order to analyse behaviour of real physical systems. When real physical prototypes are not achievable and the experiments are costly, simulations tools are very useful.

Current research provides a multi-level structure for fuel level system and a meta-model for automotive system design which were requested by industry. The meta-model and fuel level structure are the artifacts of this research. These artifacts are in the category of design-science research. Considering the fact that this project aims to develop and design artifacts and evaluate them in result, this research is a development and evaluation focused design science (Johannesson and Perjons, 2012). Data collection and analysis methods for the meta-model are unstructured interviews and context analysis respectively. Unstructured interviews are considered as a data collection method due to the availability of nine experts for meta-model evaluation. Moreover, the the fuel level structure design and analysis are through case study and simulation approaches.

2.2 Application of Method

In this section, the design-science process is applied and explained:

(I) **Problem definition** The initial requirements for this research came from
the Department of Research and Development of Scania. The requirements were to have a multi-level structure for fuel level display system design and a meta-model for system design. In this phase the existing documents, reports, meeting notes of the company and also related literature on research databases such as KTH library, Google scholar, IEEE, Springer, etc. were studied. All these resources made the knowledge-base of this research.

(II) **Objective definition** After the definition of the problem, two objectives were specified. The first one was to develop a multi-level structure for fuel level display system design that provides relationships among requirements, functions, and specification in SysML and Modelica language. The other was to provide a conceptual model for automotive embedded system design that represents the relationships among requirements, functions, and specification. In this part, the knowledge gathered on the previous task were used as an input for this step and helped to formulate more precise requirements.

(III) **Design and development** The knowledge-base information of the research was used in order to design the multi-level structure for fuel level display system. Subsequently, the meta-model was designed based on the knowledge that was acquired from multi-level structure and literature review.

On the other hand, *embedded and embodied knowledge*\(^1\) from resources were transferred to *explicit knowledge* or to the artifacts(meta-model and multi-level structure) in this step.

(IV) **Demonstration** In this step, the artifacts were reviewed and discussed by experts in the field.

(V) **Evaluation** In order to evaluate the artifacts, the fuel level structure verification was done through Modelica simulation. On the other hand, evaluation of the meta-model was conducted by the means of unstructured interviews. The results were analyzed in a qualitative way. In order to conduct the interviews, the following activities (Bhattacherjee, 2012) were performed:

**Criteria definition** The criteria were defined as correctness, generality, comprehensibility, expressiveness, and usefulness. The expected result should show if the model is correct, general, comprehensive and understandable. Also, it should reflect if the meta-model is complex, and useful for the target group (Davis, 2008; Johannesson and Perjons, 2012; Klas et al., 2010; Winter et al., 2008).

**Respondents selection** The target group for interview were selected from experts and researchers in the area of embedded systems, real time systems, and automotive industry. They had considerable experiences in

\(^1\)Embedded knowledge were the functional concepts which did not have formal and consistent definitions on the organization. The embodied knowledge was the knowledge of engineers at company about these concepts and the relationship among the concepts.
modeling of information system, development, and modeling of fuel level display system.

Questionnaire/Interview instrument design and test A questionnaire was designed in order to fulfill the criteria. Then, the first draft of the questionnaire was sent to a selected group of engineers with the purpose of testing the questions. The questions were checked in order be clear, understandable, and free from ambiguity.

Survey-interview Before interviewing, the questionnaire were sent to interviewees and some arrangements such as duration of interview, confidentially of responses were done. Afterwards, a summary of each interview was sent to the interviewee for confirmation.

Data Analysis Finally data was collected and content analysis was applied on the results.

(VI) Communication In this step, the results were presented to the experts at Scania. In addition, the material of this research was used in a conference paper which was accepted in the 9th International Modelica Conference 2012 (Liang et al., 2012) (peer-reviewed). A copy of this report will be presented at KTH and sent to KTH library.
Background

This section provides an extended background for the current research on modeling and design of embedded systems. First, the models and model based development (MBD) approach are discussed. Subsequently, the requirement classification is investigated. Then the related technologies: SysML, Modelica, modeling perspectives, and meta-modeling concepts are introduced.

3.1 Models

Models are used in every mature engineering field (Törngren et al., 2009). Models can be found in different forms such as physical models for representing mechanical components and numerical models for showing heat transfer, and so on. They are the means to communicate with the stockholders in order to reach a common realization of products. In addition, they can be used for comparing of alternative design cases, representing the behavior and properties of the system against requirements, reusing purposes, showing the desired view and the level of abstractions to developers. Therefore, models are not only represent a real or an expected system, but also they are added value to the product development.

Models are categorize as formal, constructive, and conceptual models. Formal models are analytical or mathematical models. Constructive models emphasize on behavior of the system such as programming languages. Conceptual models are in the form of graphics and are used for communicating purposes and also knowledge transfer specially in complex systems in the form of documents. System Modeling Language(SysML) and Architecture description Languages(ADL) are the samples of conceptual models. These models cannot check the correctness of software logic but can check the consistency and the syntax. All these categories are tightly related and sometimes cover each other partially. The focus of this project are the conceptual and constructive type of models.
3.2 Model Based Development

Model-based development (MBD) approach supports communication, analysis, and documentation in system development through models. In this approach, models "form the basis for the interactions between the teams of the organization, information flow within and between development phases, and for the design decisions made" (Törngren et al., 2009). MBD encompasses several disciplines in engineering domain, for instance model driven design, model based testing, and so on. The overall goal of MBD addresses functionalities, quality, time, resources and innovations. The drivers of MBD are system complexity, and maturity of technology. Goals are met with the means of abstraction, structuring, visualization, traceability, and methodology through utilization of technology. The technology in use can be languages, models, and tools. In the following sections, means and technologies of MBD related to this research are introduced.

3.2.1 Means

MBD approach assists to develop standardization, communication, and system analysis. It employs the following means:

**Abstraction** Abstraction assists to have a better definition of a complex system or to simplify that. Some examples of abstractions are classes of functions for generalization, state machines for behavior abstraction, and so on. In abstraction, the focus is on useful and essential concepts. In addition, it supports re-use in order to reduce the the cost and time of the production.

**Structuring and formalization** A model should be meaningful, and responsive for decision making and analysis. Then, it should have a proper semantic and appropriate syntax. It is very important to have a readable model which aims at a particular purpose. Models should enable reuse and instantiations.

**Visualization** Visualization is a means to demonstrate models. An example of visualization is simulation that represents an structure and behaviour of a system.

**Traceability** Traceability facilitates requirement identification in the implemented system. It provides a way to trace requirements from the first order to technical deployment. Therefore, it would be possible to find a specific request or requirement for a particular hardware or software. For example, relationship matrix, and gap analysis matrix are the tools that provide traceability from requirement to code and reverse. In addition, traceability between platform specific models (PSM) and platform independent models (PIM) are possible in tools with transformation capabilities.

**Methodology** Embedded systems engineering does not have a confirmed and well-established methodology. Although, some methodologies and contributions exist. An example of methodology for system design and verification is Virtual Verification of Designs against Requirements (vVDR) method. vVDR (Schamai et al.,
3.3. REQUIREMENT CLASSIFICATION

2010) is a model based design methodology for the verification of embedded system design against requirements. It consists of four steps: selection and formalization of requirements, system design, verification scenario definition, generate and run the verification model.

3.3 Requirement Classification

Requirement classification facilitates modeling, analysis and verification of embedded systems. The requirement are classified to functional and non-functional in general. However, there is not proper consensus about non-functional requirements term in requirement engineering community. Glinz (2007) investigated the requirements classification in prior researches and proposed a taxonomy for classification of system requirements. In this taxonomy, the system requirements were classified to functional requirements, performance requirements, specific quality requirements, and constraints. Respectively, the last three types of requirements, which are not part of the functional requirements, were categorized as non-functional requirements.

**Functional requirements** concern system or system component reaction to an input stimuli. In addition, they are related to data and functions which are needed for generating responses to stimuli.

**Non-functional requirements** are related to the attributes or constraints of a system. They concern the system behaviour in connection with performance, specific quality requirements, and constraints.

**Performance requirements** are related to the requirements that address time, speed, and etc. in a limited range. In practice, measuring performance is not difficult in comparison to other non-functional requirements.

**Specific quality requirements** are the requirements that are related to the quality of maintainability, availability, security, portability, reliability, etc., except the quality of fulfilling the functional requirements.

**Constraints requirements** restrict what is required for fulfilling the specific quality requirements, performance, and functional requirements. Constraints are such as physical, environmental, interface, etc. requirements.

Eliciting of some qualities such as availability, maintainability, and so on require a consensus among the stakeholder. Other requirements such as performance and functional requirements, which have specific threshold or formal mathematical models, are better selections for simulation and requirement verification. Therefore, these kinds of requirements are addressed in the case study of this report.
3.4 Technology

MBD technologies include modeling languages, analysis techniques, and tools. For example, modeling languages in MBD are SysML, Modelica, ADLs, Autosar, and so on. Analysis techniques are simulations and decision making techniques. In addition, the examples for MBD tools are Dymola, Enterprise Architecture, and Simulink. The MBD tools should support modeling languages and analysis techniques.

As it mentioned, many modeling languages for development of embedded system exists. In this project, the focus is on two types of languages which are System Modeling Language(SysML) and behaviour modeling language(Modelica).

UML1 and UML2 standards could not respond to the requirements of the integrated systems. Therefore, OMG has done some efforts to address the problems in UML2. SysML (OMG Systems Modeling Language, 2012) standard is the result of such efforts. SysML is a visual modeling language for system engineering applications. It can facilitate the documentation, communication, coordination in embedded systems. In addition, it enables design, specification, analysis, validation and verification of integrated systems. SysML provides more powerful framework which still requires to be proven in the area of integrated systems.

Modelica language (Fritzson, 2004) is a behavioral modeling language. The behavioral modeling languages make it possibilities to model different types of behaviour, and computational models. Moreover, they facilitate development of refinements or transformations among different levels of abstraction. These languages perform simulation, and support analysis. Other behavioral languages are Simulink and SystemC. There are some efforts to connect behavioral languages such as Modelica with UML profiles. The result is the advent of ModelicaML which is still under development. The combination of SysML with Modelica (Friedenthal et al., 2012) is a new topic in the area.

3.4.1 SysML

System Modeling language (SysML) was adopted in 2006. It uses a subset of UML2.0 diagrams and added some diagrams to UML 2.0 diagrams such as Requirements and parametric diagrams. In addition, it reused and modified some UML diagrams such as Internal Block diagram, Block Definition Diagram, and the Activity Diagram. In this thesis, structural diagrams such as Block Definition Diagram, Internal Block Diagram, and the Requirement Diagrams were used.

Block Definition Diagram

A block is defined as a "modular unit of system description" (OMG Systems Modeling Language, 2012). Blocks describe structural elements of a system and their environment such as physical and logical elements: people, documents, hardware,
software, and so on. A block Definition Diagram (BDD) contains blocks, their features (properties, and operations), and relationships (e.g. composition, specialization). BDD can be used to describe the other types of entities such as interfaces, value types, and so on. An example of BDD is shown in Figure 6.7.

Internal Block Diagram

Internal block Diagrams (IBD) describe the internal structure of a block, decomposition of blocks to parts, and the connections and interaction among the parts of a block. The interaction points are defined by ports and interfaces. Ports are representation of access points of blocks. Interfaces are methods to show behavioural features of ports. In consequence, one or more interface can be associated to a port. Interfaces are kept in SysML in order to compatibility with UML. Figure 6.8 illustrates an example of IBD.

Requirement diagram

A requirement block is a new concept in SysML. A requirement block is a capability that should be satisfied, an operation that must be performed, or a condition that must be fulfilled. Requirement blocks are illustrated by an id and text properties. A requirement diagram contains the requirements blocks, and their relationship with the other design elements, or requirements. This diagram can depict the traceability of a single requirement as well as satisfaction, refinement, verification, derive, and copy relationships. In order to illustrate the requirement relationships with other design elements, they can be illustrated in block definition diagram, use case, and package diagram. Table 3.1 depicts the description of requirement relationships and their notation.

3.4.2 Modelica Language

Modelica is an object-oriented, equation-based, and open-source language. It is designed for modeling of complex physical systems in a multidisciplinary domain (Fritzson, 2004). Namely, it can be used for design and simulation of hydraulic, mechanical, and electrical systems in automotive, robotics, aerospace, and process oriented applications. Models in Modelica can be described by mathematical models. Modelica can handle a large amount of equations in a model. Moreover, Modelica solves equations automatically. Therefore, there is no need for mathematical variables to be defined and valued by developers. In addition, this language is used for hardware-in-the-loop simulations purposes in Mechatronic systems (Otter and Elmqvist, 2002).

In addition, Modelica is Meta-programming language. In meta models, models are input of other models. Likewise, some programming languages are the input of others, in meta-programming. In these languages, the data types for the meta-program are constructed in the language. Therefore, the correct syntax of object
Relationship Description Notation

Satisfy When a requirement is satisfied by a specific design element, satisfy relationship is used. <<satisfy>>

Verify A verification element or other elements are used to verify a requirement in order to satisfy the requirement. <<verify>>

Derive Requirement When a requirement is derived from a source requirement, it is used. <<deriveReqt>>

Refine When a model element clarifies a requirement, it refines that requirement. <<refine>>

Trace When a requirement is related to another model element in general, trace relationship is used. One example of trace relationship can be the relation between a requirement and its document. <<trace>>

Copy When a requirement is reused by another requirement, copy relationship is used. <<copy>>

Table 3.1: Requirement Relationships

programming is guaranteed (Fritzson et al., 2005). Modelica makes it possible to model both high level or composite models as well as detailed libraries or equations.

A Modelica model consists of blocks, their relationships, and the connectors. Each block and connector is a class in the language. Modelica has an enrich standard library. In addition, classes of each system can be abstracted, added to its library, and reused on the other systems.

Various commercial tools such as MathModelica, Dymola, and SimulationX as well as a free open-source environment named OpenModelica are using Modelica language. Among these tools, Dymola is widely used in industry. It has a high capacity of symbolic transformation which are necessary for modeling. It is equipped with both text and graphical views. The interface of Dymola (Dymola, 2012) facilitates system integration with other simulation tools such as Simulink and MATLAB. In addition, OpenModelica (OpenModelica, 2012) is an open source environment with the purpose of promoting research and industrial development. In this research Dymola was used for modeling of fuel level system.

3.5 Meta-Modeling

MBD approach uses meta-models to define and describe models and programming languages. Meta-models specify abstract and concrete syntax, and semantics of
models. The abstract syntax describes a model syntactically; it defines entities, relationships among them, constrains and principles for making a model. The concrete syntax describes visualization styles which can be textual and/or graphical (Szti-panovits and Karsai, 2001). Semantics are meaning and interpretations of entities in models or programming languages.

Object Management Group (MOF, 2012) defined meta-model definition in a general framework (MOF) in order to move from object based to model based architecture (Kleppe et al., 2003). Earlier, OMG proposed Unified Modeling Language for modeling of object oriented information system architectures. By introducing Meta Object Facility (MOF), other similar languages could get benefit of these concepts (Bezvini and Gerbe, 2001). MOF has a four layer structure which are: object layer, model layer, meta model layer, and meta-meta model layer. The object layer is a real or imaginary system; for example, an instance of code execution. This layer might not be a model but can be illustrated by models. Accordingly, each layer is the abstraction of its lower level layer in MOF. Lower layer entities are instances of their higher level layer. The abstract concepts, their attributes, and relationship specify their behavior and structure in a language. (Figure 3.1)

![Figure 3.1: Meta Object Facility](image)

There are some standards for exchanging information format and transformation
between different abstraction layers in tools. Such tools should support interfaces and platform infrastructures, which can exchange APIs\(^1\) with the other tools. For example, tools with CORBA or COM platforms provide these facilities. Also, there are some standards for exchange of information contents and formats. Examples for the information content exchange are extensible markup language (XML), document type definition (DTD), and STEP standard (automation systems and integration, 2005). XML and XMI are used for the exchanging information formats. XMI is XML meta data interchanges that is used by OMG (XMI, 2011). XMI can exchange the meta data whose meta-model is made according to MOF framework. In addition, SysML uses XMI standard for model transformation. The standard for the exchange of SysML diagrams takes care of the graphical information exchange.

### 3.5.1 Properties

Models in general can have generic, internal, management, and usage properties (Johannesson and Perjons, 2012). *Generic* properties show how much a model can interact with the environment. The examples of generic properties are expressiveness, correctness, and generality. *Internal* properties explain about the internal structures of the model such as coherence, and consistency. *Management* properties explain how the model evolves during the time. The samples of such properties are maintainability and flexibility. *Usage* properties show how the model can be perceived by users. Comprehensibility, and usability are in the usage property group. In this project, the focus is on generic and usage properties, and in particular correctness, generality, expressiveness, comprehensibility, and usefulness.

**Correctness** Correctness means the degree that a model match exactly with its specific domain.

**Generality** Generality means that a model should be free of any application domain and languages. Ahrens et al. (2011) defines generality in reusability category. Generality is one of the the success factors of programming languages according to Chen et al. (2005). In meta-models, generality indicates that meta-modeling can describe entities, and their relationships on the lower level models in a specific domain.

**Expressiveness** Expressiveness means that a model should represent entities of its domain. It also implies the ability to indicate a complex structure in an intuitive way. Furthermore, expressiveness indicates that a model is semantically consistent and concepts can only be interpreted to the same meaning in the domain.

**Comprehensibility** Comprehensibility is the degree at which a user understands a model. There are other related properties to comprehensibility like simplicity. Simplicity of a model indicates that a model holds the minimal required entities and concepts, while it keeps a consistence structure (Chen et al., 2005).

---

\(^1\) Application Program Interfaces
3.6. PERSPECTIVES

Perceived usefulness Perceived usefulness is a degree which someone believes that a specific model is valuable to use in practice and would improve his/her performance. Perceived usefulness has a direct relation to understandability, simplicity and clarity. In usage category, usefulness is more powerful than ease of use. If users believe that a particular function is useful, they will learn it in order to achieve desired functionality (Davis, 2008).

3.6 Perspectives

An abstraction level includes a combination of different aspects of a whole product architecture. Each aspect, which represents a distinct model, is called a perspective. The architecture of an embedded system can be viewed in different perspectives at various abstraction levels.

The perspectives and abstraction levels are widely used in EAS-ADL language. In EAST-ADL, each architecture level contains different perspective models. For instance, the design level includes a hardware and functional perspective. The elements of one perspective can have relationship with the elements of another perspective (relationships are explained in the subsection 3.6.1). Other Meta-Models such as HRC for contract-based design followed EAST-ADL language. Although predefined levels of abstraction in some modeling languages and approaches such as contract-based design and EAST-ADL exist, they are not acceptable to manage variety of application domains (Atesst, 2012; Baumgart et al., 2010; East-ADL, 2012). For instance, applications in various companies in the same domain might require tailored abstraction levels based on their needs.

Perspective, view, and break-down are referred to the same concept in different areas such as embedded systems (Lönn et al., 2004), software engineering (Philippe, 1995), and product life cycle support (Björn, 2009; Product breakdown, 2012) respectively. Contract-based design classification for perspectives was appropriate for current research. Contract-based design defines user, logical, and technical perspectives. Moreover, requirement perspective was added by following EAST-ADL design.

---

2EAST-ADL is an Architecture Description Language for embedded systems in automotive industry. It uses AUTOSAR and complements that by adding higher levels of abstraction (Biehl, 2010). EAST-ADL has different aspects such as features, functions, software component, hardware component, and requirements. Traceability relations exist among different level of abstractions.
3HRC means Heterogynous Rich Component meta-model which is used by contract-based design approach. Baumgart et al. (2010) claims that HRC is an extension of EAST-ADL language.
CHAPTER 3. BACKGROUND

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>User Perspective contains function-centric models. The function and features that are used in this perspective are understandable by the customers.</td>
</tr>
<tr>
<td>Logical</td>
<td>Logical perspective includes logical structure of a system. Embedded software is a part of logical perspective of a system.</td>
</tr>
<tr>
<td>Technical</td>
<td>Technical perspective concentrates on physical Hardware systems with communication links.</td>
</tr>
<tr>
<td>Requirement</td>
<td>Requirement perspective contains requirements of a system.</td>
</tr>
</tbody>
</table>

Table 3.2: Design Perspectives (Baumgart et al., 2010)

3.6.1 Abstractions and Perspectives Relations

In this section the relationship among the abstraction levels and perspectives are described. These relationship are supported by some modeling languages, and approach such as SysML, EAST-ADL, and Contract-based design.

**Realization** Realization relationship is referred where a detailed component realizes the specification of the supplier. The realization relationship can exist between components on distinct abstraction levels. For instance, a software implementation should realize a software design.

**Allocation** When a set of model elements are allocated to another, allocation relationships are used. This relationship exists among different perspectives. The allocation includes the decision to map a logical component to one or more technical/hardware component. Such as allocating software or logical abstraction to physical/technical abstractions.

**Refinement** Refinement is the relationship among various abstraction levels in the same perspective. It is related to the design when the implementation details are successfully inserted. For instance, a function refines another function by inserting more implementation details. Refinement is sometime used in different ways: the refinement relationship within one language or different language. It means a refined element can be in another language such as refinement between design and implementation.

Each relationship in SysML is shown by an arrow between two elements with a constraint which indicates the type of relationship.
In 1900, Scania company was founded for bicycle production in Malmö. In 1911, Scania joined another company, Vabis, for manufacturing of trucks and buses. After one century of production, Scania is one of the world’s leading manufacturers for heavy vehicles, trucks and buses, and marine engines. The company operates in around 100 countries with more than 35,500 employees. Nowadays, research and development center is in Sweden, while the production activities are in South America and Europe. Scania have initiated modular concept in their products since 1930s. Modularization means each customer is proposed a tailored and customised vehicle configuration during a suitable time. Therefore, a customer could choose various functionalities and the selected functionalities would define the final product and cost. Having a unique modular product range is the Scania’s most important success factor.

This research was performed at Scania AB in Södertälje. This section provides a background in formation to concepts which are used in this report.

4.1 SESAM

During the years 1991 and 1992, the initial steps toward having a new electrical system at Scania were taken. The project named SESAM that means 'Scania Electrical System created at the year 2000'. The pre-developments phases of SESAM project was initiated at 1994. The objectives of SESAM were increasing the reliability, availability, safety of the Scania electrical system, supporting old functionalities, adding new technologies, supporting modularity of functionalities, maintainability, decreasing costs of system functionalities, and so on.

Nowadays, SESAM is an electrical system inside all vehicles at Scania. By introducing SESAM concept, both buses and trucks could employ the same electrical system (Persson, 2009). SESAM consists of a collection of Electronic Control Unit Systems (ECU system).
4.1.1 Electronic Control Unit System

Electronic Control Unit (ECU) system is a complex system which is comprised of some electronic components such as sensors, actuators cabling, an ECU which is connected to a CAN-BUS, and embedded software. ECUs manage other electronic system(s) or subsystem(s).

Each ECU system has 'well-defined' and 'closely related' functions known as user functions (SystemUtredning, n.d.). ECU should be flexible on choosing these functions in a modular way; one user function can be included or excluded from an ECU according to customer's request.

CAN BUS

A Controller Area Network (CAN) is one of the techniques for constructing a network (Westman, 2003). Buses are network parts which are used for communication between network nodes (ECUs). A CAN BUS is a communication segments based on CAN techniques. The CAN-BUS is a standard for network communication.

Different Types of ECU

An electrical system of automotive consists of various kinds of ECUs. ECUs, which are involved in fuel level display, are named COO, EMS, and ICL ECUs.

COO (Coordinator system), EMS (Engine Management System), and ICL (Instrument Cluster System) ECU systems are the core ECU systems on a vehicle. They are the minimum ECUs that a vehicle requires (Figure 4.1). Other ECUs can be added based on the vehicle specification which customer defines (Westman, 2003). COO-ECU system controls the input and output of the other ECUs, collects information and works as an integrator in the system. EMS-ECU system controls the engine. ICL-ECU system controls related human machine interfaces (HMI). Human-Machine Interface (HMI) is designed to show information to the driver. It is a part of the ICL-ECU system. Information should be directed from the COO-ECU and pass through CANBus until it will be available on the ICL-ECU and HMI. A HMI consists of gauges, telltales, text display, warning lamp, etc. One of the functionality of HMIs is to show warnings related to malfunction of safety or legislation functions (SystemUtredning, n.d.). For example in fuel level display system, COO-ECU receives fuel consumption from engine (EMS-ECU), reads the fuel level from the tank and estimates the fuel volume and fuel level warning. As a result, COO-ECU system sends the result to ICL-ECU which contains the gauge in order to be readable for driver.

4.1.2 Functional Architecture (Fuel Level Display)

ISO-IEC-IEEE24765 (2010) defines functional architecture as an arrangement of functions, their decomposition to sub-functions, and interfaces. It enables reusabil-
ity and change resistance (Eden and Selmarker, 2007). Functional architecture consists of user functions, allocation elements, and function variables at Scania.

**User Function**

A user function consist of a group of functions that address a functionality of the desired system. This functionality should be perceivable by users. On the other hand, they can be seen as an arrangement of use-cases and scenarios of the system (Westman, 2003). In other words, user functions express the customers’ desires of the system (Eden and Selmarker, 2007; Scania Wiki, 2012). Therefore, the implementation of an user function is directly related to the vehicle configuration which is chosen by users. With different configuration, the scenarios of the system functionality will change. Therefore, The entire of the SESAM system can be considered as a provider of user functions for the customers. Users are able to interact with the electrical system in the scope of user functions. It means that drivers can have some observations via HMI or by the symptoms of the system such as ABS pedal pulsing(figure 4.2). There are different groups of users who interact with the vehicle system such as passengers, mechanics, salesmen, and drivers. The "User" term refers to a customer or a driver on "user functions".

**Allocation Element**

User functions are realized by logical functions of the system which are called allocation elements. Allocation elements are realized by the software components during implementation (Oberg, 2007; Scania Wiki, 2012).
Function Variable

Function variables are the logical variables of the system. They are realized by (internal and external) interfaces of allocation elements. The relation between allocation elements or allocation elements with system components interfaces are possible through function variables.

4.2 Different Types of Document

Common documents at Scania are UF, AE, FAD documents and system descriptions.

**UF** User Function (UF) documents contain the description of user functions, user function requirements, and safety related information. User function requirements are the high level requirement which are desired by customer.

**AE** Allocation Element (AE) documents contain allocation elements description, allocation element requirements and the related information to the AE such as specifications of AE interfaces. Allocation element requirements are the system level requirements which can be perceived by engineers.

**FAD** Function Allocation Description (FAD) documents contain the information about the relation between user functions and allocation elements, allocation elements and ECU, allocation elements and function variables, and function variable and interfaces on the system.

**System Description** System Description describes an ECU system.
A Meta-Model for Automotive Embedded System Design

Meta-modeling is the use of models in building other models. In this concept, some models are the input of other models. Meta-models can facilitate modeling and simulation by providing a higher level of abstraction. In this section, a conceptual model (Figure 5.1) for automotive system design was proposed. Since the model should be suitable for the industrial sector, the inputs from Scania functional architecture, multi-level structure for fuel level display, and other case studies in the literature (Larses, 2005) were reviewed. This meta-model provides the relationship among design artifacts such as requirements, functions, and specifications. This model was designed in four different perspectives: User, Logical, Technical, and Requirement perspectives.

5.1 User Perspective

User perspective (Figure 5.2) holds features and functionalities that are perceivable by customers. These features and functionalities are high level functionalities of a vehicle. Each vehicle consists of various functionalities which are chosen by customers. They are named 'Customer Functions (CF)’. An example of a customer function is a fuel level display. Customers can read and understand the fuel level indicator in the gauge. Therefore, they can get the idea about the amount of fuel level in the tank. In addition, they can be aware when tank should be refilled in order not to get out of fuel during driving.
5.2 Logical Perspective

Logical perspective (Figure 5.3) is a result of analysing features and functionalities of vehicles. It consists of logical functions, abstract functional unit, logical interface, function variable, and data type. The features on user perspective are transformed to "logical functions" by system engineers. Each logical function is the refinement of other logical function in a system. They are containers for some "abstract function units" which realize the higher level functionalities (CFs). Abstract function units (AFU) are responsible for a specific behavior of a system. They enable modularity in a system. Different AFUs can realize a specific CF and also specific CF can be realized by different AFUs. Abstract functional units communicate through "logical interfaces". Logical interfaces are gateways for communications among different AFUs. Logical interfaces contain "function variables". Function variables are a set of variables that are exchanged by AFUs.
5.3 Technical Perspective

Technical perspective (Figure 5.4) represents a physical system. The main component and also the container of the technical perspective is 'hardware component (HW Component)'. HW Component can be both a physical part of a system such as a sensor or the whole ECU system that contains other HW Components. In addition, it includes 'hardware interfaces'. Hardware interfaces are such as pins that enable the connection among different components. The 'communication links' in technical perspective are the actual cables which are considered as a specific HW component. The communication links have connection to at least two interfaces. The links carries (a) signal(s), and energy, which flow(s) in the system and is named 'item flow'. Item flow contains one or a set of signals in the system. They are realized by hardware interfaces.

![Figure 5.4: Meta Model - Technical Perspective](image)

5.4 Requirement Perspective

Requirement perspective (Figure 5.1) contains requirements for customer function, logical function and physical elements which are called: CF requirements, LF requirements, and physical requirements respectively. 'CF Requirements' are the high level requirements that are demanded by customers. The lower level requirements are 'LF Requirements'. These requirements are defined by the system engineers and derived from CF requirements. LF Requirements are the requirements for log-
ical functions and abstract function units. There are also 'Physical Requirements' that define specification of physical parts of the system or HW components.

In this meta-model, different perspective are interrelated through realization, and allocation dependencies to another. Logical and User perspectives are related through realization. 'AFU' in logical perspective realizes 'CF' in user perspectives. In addition, logical and technical perspectives are connected through the allocations of 'AFU' to 'HW component', and 'function variable' to 'item flow'.
Figure 5.1: Meta-Model for Fuel Level Display system Design
Multi-level structure for Fuel Level Display

The multi-level structure implies a system decomposition from higher level of design into detail components in different design perspectives. It illustrates the requirement satisfaction and system design verification. The purpose of this chapter is to represent a multi-level structure of a fuel level display system design in order to provide relationship between requirements, functions, and system components.

In order to design and develop the multi-level structure for Fuel Level Display system (FLD), the technical report of FLD was studied. Then, the requirements were extracted, and the system was designed in SysML. Subsequently, the verification model was designed in order to test the multi-level structure. Finally, the structure was developed in Modelica and the result of the simulation was illustrated.

6.1 About Fuel Level Display System

The Fuel Level Display system (FLD) estimates total fuel level in a tank. In addition, it activates fuel level warning when the fuel level is lower than a determined value. The total fuel volume and low fuel level warning are displayed in the dashboard (figure 6.1) to the driver.

Vehicle movements cause the fuel to move and sway in any direction. When the amount of fuel in the tank decreases, sloshing of fuel increases. Therefore, it is hard to estimate the actual fuel level in the tank(s). It is very important to understand the precise fuel level when the tank is getting empty. The Fuel Level Display (FLD) system should estimate the fuel level in different road situations, and due to any fuel movement. In result, output information should assist the driver to refill the tank when it is necessary. In order to have an accurate fuel level, Kalman filter is used in FLD system. Kalman filter (Kalman, 1960) is a recursive algorithm for predicting the values of linear dynamic system from noisy data.
"Estimation of fuel level" and the "activation of low fuel level warning" are performed by COO-ECU system. In order to estimate the fuel level, COO receives "fuel consumption rate" from EMS-ECU system and also the "fuel level" from tank-sensor(Figure 6.2). Tank-sensor is a part of COO-ECU system. Afterwards, COO-ECU system calculates the total fuel level and compares the result against the lowest value of fuel level. When the fuel level is lower than the predefined value the warning is activated. COO-ECU system sends the total fuel level and low fuel level warning to the gauge and warning lamp in ICL-ECU system.

6.2 Requirement Selection

FLD is a User Function(UF) for a vehicle. In addition, "Estimate Fuel Level" and "Fuel Level Warning" are the Allocation Elements(AEs) for FLD. The requirements for these functionalities can be found in three technical reports correspondingly at Scania: UFR 18, ARE 201, and AER202. Each technical report contains a background description, requirements, and the related safety description.
6.2. REQUIREMENT SELECTION

6.2.1 User Function Requirement 18

User Function 18 (UF18) is a high level function for fuel level display system. It indicates that fuel gauge shall be displayed and low fuel level warning shall be shown to the customer. Therefore, UF18 has two functions for fuel gauge and low fuel level warning. All vehicles contain fuel gauge but only some of them have fuel level warning indicator. Warning can be presented as a light in a warning lamp or as an alarm. Table (6.1) illustrates the nominated requirements from UFR18 for this study. UF18 is realized by two Allocation Elements 201 and 202 which are explained in the following sections.

<table>
<thead>
<tr>
<th>UFR18-1.2</th>
<th>The UF shall provide information about the fuel level in the instrument cluster (ICL).</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFR18-15</td>
<td>The displayed fuel level shall not deviate more than -5% from the total actual remaining fuel volume that is collectable from the tank(s).</td>
</tr>
<tr>
<td>UFR18-16</td>
<td>The displayed fuel level shall not deviate more than +5% from the total actual remaining fuel volume that is collectable from the tank(s).</td>
</tr>
<tr>
<td>UFR18-24</td>
<td>The limit for activating low fuel level warning shall be 10% for a total tank size below 900 litres.</td>
</tr>
</tbody>
</table>

**Table 6.1:** User Function Requirement 18- Fuel Level Display (Erlandsson, 2011)

Allocation Element Requirement 201

Allocation Element 201 estimates the percentage of fuel level in tank. Table 6.2 shows the selected requirements from AER201.

<table>
<thead>
<tr>
<th>AER201-1.1</th>
<th>This AE should calculate and display the current percentage level in the fuel tank.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AER201-9(a)</td>
<td>If the truck has one fuel level sensor connected, its voltage value should be transformed into a corresponding volume value in litres.</td>
</tr>
<tr>
<td>AER201-9(b)</td>
<td>The volume value should be low pass filtered before set to fuel Level signal ($y_a$). The filter is given by equation (refer to equation B.1 in Appendix B) with $C_{pre}=0.99$.</td>
</tr>
<tr>
<td>AER201-11</td>
<td>Total Fuel Level should be the output of a filter that includes information from both fuel Level and fuel Rate to achieve a stable signal. The filter should be implemented with a Kalman algorithm given by equation (refer to equations B.2-B.3 in Appendix B) with the feedback gain $K = 1.0786 \times 10^{-5}$.</td>
</tr>
</tbody>
</table>

**Table 6.2:** AER201 Requirements- Estimate Fuel Level (Wallebäck, 2010a)
Allocation Element Requirement 202

Allocation Element 202 enables the warning lamp when the amount of fuel is fairly low in the tank. Table (6.3) shows chosen requirements from AER202.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AER202-1.1</td>
<td>A low fuel level warning is introduced which is activated if the fuel level drops below a predefined level. The warning will be displayed for the driver with the ICL.</td>
</tr>
<tr>
<td>AER202-2</td>
<td>The low Fuel Level Warning should be set to 1 (true) when input total Fuel Level is below a pre-defined level. The level should be 10% for tank sizes equal or below 900 liters and 7% for tanks sizes larger than 900 liters. (The total Fuel Level represents the usable volume in the tank, i.e. the total volume minus the volume left in the bottom when the fuel level sensor indicates 0%.)</td>
</tr>
</tbody>
</table>

Table 6.3: AE201 Requirements- Estimate Fuel Level (Wallebäck, 2010b)

6.3 System Design

This part includes the design of FLD system. FLD system design has been done in four perspectives: requirements, user, logical, and technical perspectives. In addition, SysML standard and Modelica language were used for modeling of the system.

6.3.1 Requirement Perspective

After analyzing the requirements, requirement diagram of the system based on SysML standard was sketched. Figure (6.3) displays the requirement diagram that extracted from requirements in UFR18, AER201, and AER202 documents. The UFR18-1.2 is the higher level requirement which has the general description of FLD. Other UFR18 requirements; UFR18-15, UFR18-16, and UFR18-24 are derived from UFR18-1.2. UFR18-15, and UFR18-16 describe that the total fuel volume shall be ±5. They are performance requirements. In addition, UFR18-24, which describes the limitation for warning, is a functional requirement. Other requirements such as AER201-1.1 and AER202-1.1 are derived from UFR18-2.1 requirement. These are high level requirements for AE201 and AE202. Accordingly, the AER201-9(a), AER201-9(b), and AER201-11 are derived from AER201-1.1. In addition, AER202-2 is derived from AER202-1.1.
6.3.2 User and Logical Perspective

After designing the requirement perspective, user and logical elements of FLD could be defined. Subsequently, the User and Logical Perspective based on SysML standard were designed. Figure 6.4 illustrates the block definition diagram of User and Logical Perspective. User and logical elements were identified as UF18, Fuel_Level_Sensor_Function, AE201, AE202, Gauge_Function, Lamp_Function, Simulink, and c-code sub-functions.

In FLD, UF18 carries the main features and customer functions in the system. Therefore, only UF18 block stays in user perspective boundary. UF18 satisfies UFR18-1.2, UFR18-16, UFR18-17, and UFR18-24 requirements. In addition, it is realized by AE201 and high level functions on logical perspective. On the other hand, the logical perspective consists of the logical functions for reading the fuel level in tank, and calculating and displaying the total fuel level. Therefore, high level functions are Fuel_Level_Sensor_Function, AE201, AE202, Gauge_Function, and Lamp_Function. Since this project concentrates on the COO-ECU application functions such as AE201 and AE202, the graph explores the related functionalities to the functions in COO-ECU application domain. AE201 and AE202 are realized by two Simulink sub-functions; fuel Level estimation Algorithm and Lower Fuel Level Warning. In fact, the Simulink model is "refinement" of higher layer functions which are application related. The Simulink model will be translated into C language which is named fuel.c. The C-file is one of the components of FLD application in
the application layer of COO-ECU. C compiler compiles the whole SW package to Byte-code or an executable code. The compiled code is allocated to the COO-ECU. The internal structure of the SESAM for FLD is depicted by Internal Block

![User Perspective](image1)

**Figure 6.4:** Block Definition Diagram - Logical and User Perspectives

Diagrams. Figure 6.5 shows the internal block diagram for the high level functions in the logical perspective. It also specifies the connections and relationships among high level functional elements.

![Logical Perspective](image2)

**Figure 6.5:** Internal Block Diagram - High Level Functions of Logical Perspective
6.3. SYSTEM DESIGN

In addition, figure 6.6 shows the Internal Block Diagram for the Simulink model which is called refined logical perspective.

![Refined Logical Perspective](image)

**Figure 6.6:** Internal Block Diagram - Refined Logical Perspective

6.3.3 Technical Perspective

Technical Perspective of FLD system consists of tank and SESAM system (figure 6.7). SESAM is made up of ECU systems namely COO, ICL, EMS which are interconnected through CAN cable. Power Supply provide the electronic power to the ECU systems. According to the project definition, COO-ECU system is explored. COO-ECU System contains fuel level sensor and COO-ECU which are linked through sensor cable. COO-ECU consists of ECU Hardware and the Byte-code which is embedded in the Hardware. Figures (6.7, and 6.8) illustrates the BDD and IBD of FLD system in technical perspective.

![Technical Perspective](image)

**Figure 6.7:** Block Definition Diagram - Technical Perspective
6.3.4 Fuel Level Display Structure

In this part, dependencies among among different perspectives are discussed and the whole structure of the system is depicted. The dependencies includes realization, allocation, and refinement relations (section 3.6.1). The satisfy relationship from requirement diagram is used for relations between the requirement perspective and any other design perspectives. In FLD system, UF18 satisfies UF18-1.1 and also UF18-15, UF18-16, and UF18-24 (figure 6.9). In addition, Fuel_Level_SensorFunction satisfies AER201-9(a), and AER201-9(b) requirements and AE201 satisfies AER201-11 requirement. Similarly, AE202 satisfies AER202-2, and AER202-1.1 in higher level. Moreover, the relationship among the elements in user perspective and logical perspective and also among different abstraction levels in logical perspective are realization. For instance, AE201, and AE202 realize UF18; therefore, Simulink model realizes AE201, and AE202 (Figure 6.9). Dependencies among the user and logical perspective with the technical perspective is allocation. It means that UF18 is allocated to SESAM and C-code is allocated to Bytecode.
Figure 6.9: Block Diagram - Fuel Level Display Structure
6.3.5 Verification Design

When a structure for traceability among different perspectives of the system was designed, a scenario for verification were defined. Finally, verification model was designed.

Scenario

For verification, tank generated two fuel level signals for SESAM and requirement input. The signal for SESAM input contained noisy data to simulate movement in trucks. In comparison, the signal for requirement input contained actual value of the fuel level in tank. SESAM estimated the fuel level and warning and sent the result to verification box. In addition, the requirement checked the warning and sent the actual value of fuel level and warning to verification box. Verification box compared these pairs of results based on constraints on performance and functional requirements. The result of verification showed the validation or violation of the design by 1 and 0 values respectively.

Verification Model

In this part the verification model was designed and combined with the designed model in SysML. Figure 6.10 illustrates the verification design for top level requirements.

![Figure 6.10: verification and Requirement Top Level](image)

In addition, it was possible to validate different levels of abstraction against each other in the logical perspective as well as the technical and the logical perspectives against the requirements.

6.3.6 Multi-Level Structure in Modelica

The resulted structure (refer to section 6.3.4 figure 6.9) was used to model the system in Modelica. The FLD structure shows the system in different levels of abstraction. In this section, the corresponding elements in each level of abstraction were identified.
in order to design in Modelica (Figure 6.11). As figure 6.11 shows, UFR18 contains the requirements of UF18. UFR18 and UF18 are the requirements and logical elements of tank and SESAM (Level 1). There is no corresponding requirement and logical elements for the second and third level of SESAM decomposition (Level 2, and 3). However, the AER201, and AER202 are the requirements for AE201, and AE202. Subsequently, AE201, AE202, Simulink model, and C-code are the logical elements for the lowest level of SESAM decomposition (Level 4).

**Perspectives in Modelica Model**

In order to model the multi-level structure (figure 6.11) in Modelica, three perspectives in each level should be modeled. User and logical perspectives provide the functionalities which are expressed in requirements. On the other hand, the requirements and their functionalities are integrated tightly together. Therefore, the logical/user and requirement perspectives were considered in one boundary in order to simplify the Modelica model. Figure 6.12 illustrates the arrangement of different perspectives in Modelica. Two perspectives were considered in each level of Modelica model. The top level of Modelica was consisted of user-and-requirement perspective and technical perspective. The other levels contained logical-and-requirement perspective and technical perspective.

![Perspectives breakdown in Modelica](image)

**Figure 6.12:** Perspectives breakdown in Modelica

Figure 6.13 depicts the perspectives structure in Modelica based on SysML standard.
The technical perspective was breakdown in different levels and their corresponding logical and requirement perspectives were modeled in the same level. The technical perspective and other two perspectives were connected with links and ports in Modelica.

Level 1

Based on the discussed structure (figure 6.11), Modelica model was designed in Dymola. The first level contained SESAM system, and tank in Technical perspective. In addition, it contained UFR18-1.2 and verification blocks in user and requirement perspectives. In technical perspective, the tank generated the ramp signal as an input to the requirement perspective and it also sent the noisy signal to the SESAM system. The technical and logical perspective were linked to internal ports. Since the ports of SESAM and the verification block were internal in their own block, they could not be connected together in the outer class in Dymola. Therefore, an extra internal port in their outer class were defined (visual port, and tank port) to connect the internal ports of the blocks. The set of visual-port and tank-port were the interface for connecting two logical and technical perspectives. Dashed lines were considered for the links to the logical perspective. Because the tank had an output to the logical perspective in top level (UFR 18), the link was defined as a dashed line. In addition, the tank sent the fuel volume to AERs inside SSAM. Therefore, tank had a connection to SESAM with a dashed line. (See figure 6.14)
SESAM received the fuel level input from tank and proceeded with the estimation of fuel volume and enabling warning lamp. UFR18 (Figure 6.15) received the actual fuel volume from tank and displayed the same result to the user, also it check if the actual fuel volume was less than ten percentage of the tank, the output set to zero/one and de/activated the lamp correspondingly.

Figure 6.14: Fuel Level Display - level 1
The result of the SESAM and UFR 18 were sent to the verification block in Figure 6.16. The verification contains two blocks; performance and functional verification.

The performance verification block (figure 6.17) checked whether SESAM estimation for fuel volume was less or more than five percentage of the actual volume in
tank. Therefore, fuel level result from SESAM should be between (1.05*fuel volume) and (0.95*fuel volume) otherwise the result of "and" operator will be "false".

On the other hand, the functional verification (figure 6.18) checked the warning lamp result from SESAM and compared that to UFR18 output. The output of this block was true when both of the inputs were the same.

Level 2
The SESAM for FLD consists of three ECU systems; ICL, EMS, and COO which are connected by yellow and red CAN Buses. Power supply provides the electricity of the SESAM system and is connected to the ECUs through electrical wires. EMS
system sends fuel rate through red CAN Bus to COO system. After calculating fuel volume in COO system, it delivers the results through yellow CAN Bus to ICL system. The results are shown to the customer by HMI which is part of ICL ECU system. In this level wires and CAN Buses are designed as a block. They contain switches in order to be able to check the possibility of disconnections during simulation for hazard analysis. In this step, the SESAM system was broken down to a lower level and the mentioned scenario was modeled (figure 6.19).

Figure 6.19: Fuel Level Display - level 2 - SESAM

Level 3

The next level (figure 6.20) shows breakdown of COO-system to Fuel level sensor and COO-ECU. The fuel level sensor is connected to the COO-ECU through sensor wire. It senses the level of fuel in tank and sends the fuel level to the COO-ECU. The inputs from higher level logical layer are connected to the COO-ECU in order to link them to the lower level layers.
6.3. SYSTEM DESIGN

In figure 6.21, COO-ECU is broken down to the corresponding technical, logical and requirement perspectives. In technical perspective, COO-ECU includes hardware component, and bytecode. The *hardware component* is COO-ECU hardware. The *Bytecode* is the executable code which is mounted on COO-ECU. The corresponding components in the logical perspective are AER201, AER202, and the Simulink model. The verification block is reused (figures 6.17, and 6.18) to verify the requirements in this level. Figure 6.21 shows the verification of requirements on level 1 against Simulink model. The *AER201* and *AER202* blocks contain background and derived requirements (figure 6.22). The *boolean constant* in level 4 helps to switch between background or derived requirements in AERs decomposition. AER201 derived requirements include "converting voltage to fuel level" (AER201-9(a)) and "low pass filter" (AER201-9(b)). It gets the signal from sensor and filters that before sending to simulink block. The Simulink block (figure 6.21) receives two inputs: the "fuel rate" from EMS system and "low pass filter" from AER201. It calculates Kalman filter and sends the "estimated fuel level" to evaluation block. Simulink model is the actual model is running currently at Scania.
CHAPTER 6. MULTI-LEVEL STRUCTURE FOR FUEL LEVEL DISPLAY

Figure 6.21: Fuel Level Display - level 4 - COO-ECU

Figure 6.22: Fuel Level Display - AER201
6.4 Simulation Result

In this section, requirements were validated in Dymola simulation environment. The validation scenario was performed between the Simulink block in logical perspective and user function requirements. The simulation of the model was in the scope of another master thesis from Linkoping University in the same team. However, the result is brought here in order to prove the behavior and usefulness of the designed model. In order to execute the model, the co-simulation between Simulink and the Modelica model was performed. This means that the Simulink block at level 4 (figure 6.21) of the Modelica model sent the fuel rate and the fuel level to the Simulink model in Simulink environment. Simulink model calculated the results for FLD. It sent back the fuel level display and fuel level warning to Modelica. Therefore, the outputs of Simulink block in Modelica were the fuel level and fuel level warning. These results were sent to the output of SESAM system and verified.

The simulation result of validation is presented in the Figure (Figure 6.24). The red line shows the actual fuel level in the tank in nominal case. The actual fuel level in tank is the input for user function on logical perspectives. In this case, the fuel level in the tank is changing from 20% to 0% during 21600 seconds. In real scenario, the tank is shaking. Therefore, the actual level in the tank, based on the movements on the truck, changes. Therefore, a random number generator algorithm is added to the actual fuel level in tank in order to simulate the shakes on the tank. The blue graph shows the fuel level with noise. This is the fuel level that is provided by tank and is sent to SESAM input. The yellow line presents the total fuel level which is the output of allocation element in level 4 of the Modelica model. The green line shows the total fuel level which is the output of Simulink model at level 4. The black line expresses the top level requirement verification which is the
verification of Simulink model against the high level requirements (UFR18). The result shows that in the first 17000 Second the requirements are violated. It means that the estimated total fuel level is more than \( \pm 5\% \) of the actual fuel level in tank. In this case, the requirements are violated from 17000 second of simulation.

![Simulation Result](image)

**Figure 6.24:** Simulation Result
Figure 6.11: Fuel Level Display Structure Multi Level Structure
In this chapter, the designed meta-model (chapter 5) was evaluated through unstructured interviews. In order to conduct the interviews (Bhattacherjee, 2012), the criteria for evaluation was defined, the respondents were selected, the questionnaire was defined and tested. Subsequently, the interviews were conducted. After interviews, data analysis was performed.

The results of the interviews were collected and analysed with two purposes: (1) to obtain a holistic view of modeling and meta-modeling usage among the target group, and (2) to evaluate the meta-model with the focus on generic and usage properties such as correctness, comprehensibility, expressiveness, generality, and usefulness.

### 7.1 General Information

#### 7.1.1 Target group

The target group for interview were chosen based on their expertise and their familiarity with information modeling, and fuel level display. They were selected from different organizations, competencies, and positions. Nine experts participated in interviews for evaluation of the meta-model.

Four interviewees were researchers from different universities in Sweden such as KTH Royal Institute of Technology, Chalmers University of Technology, and Linköping University. Others were active in development, software architecture, and management sectors from different industrial organizations and companies such as Scania, Accenture, and Systemite.

Besides being experts in embedded system area, the target group was expected to be familiar, and worked with concepts such as development, and information modeling. Among the target group, two were experienced in development, two had worked both in the area of development and information modeling, one was
experienced in modeling and especially was familiar with fuel level information modeling, and the rest (four) were experienced in information modeling.

Figure 7.1: Area of Expertise

Among the interviewees, two of them had about one to four years work experience, two had five to nine years work experience, and five had about ten to twenty years work experiences in the area of their expertise. Therefore, the majority of these people had more than five years work experience. The interviewee with lower years of experience (one to four years) were Ph.D. researchers. One especially worked with model-based development of embedded systems and EAST-ADL. The other was familiar with modeling of fuel level system. Therefore, their input were very valuable for this research.

Figure 7.2: work experience(years)

**Result:** According to the analysis, the target group competence was appropriate for this research. They were either knowledgeable in information modeling, or development of embedded systems. In addition, they had a noticeable work experience (more than 5 years experiences) in the field.

### 7.1.2 Usage and Importance of Meta-modeling

In order to capture the opinion of the interviewees about the importance of meta-modeling and usage of that in practice, two questions were raised. The target group were asked whether their organization was following any meta-model for modeling and/or development. In addition, they were questioned about the importance of having an updated meta-model of systems in their research or organization. These questions emerged during some discussions. Therefore, the question regarding the
usage of meta-modeling was asked from eight interviewees. The question about the importance of meta-modeling was questioned from six. However, discussions about conceptual-modeling existed beyond the interviews.

In response to the question regarding the usage of meta-models, four of the participants claimed that they were using and updating meta-models for their own development and research. On the other hand, three participants mainly from Scania told that they were aware of the need for having a meta-model and they aim to develop an appropriate meta-model for system design and development. One of their motivations for development of a meta-model was the incompatibility of concepts definitions with their usage, which created lots of discussions and disagreements among system engineers and researchers. In comparison, one interviewee expressed that his group was not using meta-modeling for in-house development, because, he had a small group. However, they provided meta-models for their customers.

According to one of the experts, "meta-models are the key concern of the organizations". Here is the list of the reasons which the target group mentioned for the importance of meta-models:

- The market demand for certifying automotive products based on safety standards (such as ISO-26262).
- To cope with complexity of products.
- To manage changes and variety of configurations.
- To facilitate communication among engineers with a common understanding of development concepts.
- To utilize meta-models for quality checking purposes.

**Result:** Around half of the participants were using meta-modeling in their working area, while the rest were initiated to develop this concept or not using meta-models. In addition, having an appropriate meta-models could help the organizations in communications, change and configuration management, and safety certifications purposes.
CHAPTER 7. EVALUATION

7.2 Analysis

In this section, the designed meta-model (chapter 5) was explained to the experts. Later, the questions were asked and responses were used to evaluate the meta-model. The evaluation has been done with the focus on meta-model properties (refer to section 3.5):

7.2.1 Correctness

Correctness of the described meta-model was enquired. Most of the experts directly expressed that they think the model is correct (6 of 9). Two of the experts indicated that the models represent one part of reality; therefore, models cannot be correct and not correct. However, they did not see anything wrong with current model. On the other hand, one of the experts mentioned that the model caught the central information but it needed more details to be added to the model in order to judge about that. For instance, he mentioned that many-to-many relationships should to be broken to one-to-many relationships by creating intersection classes.

*Result:* In result the model was correct based on six interviewees. Two interviewees mentioned that models generally could not be said to be correct or incorrect. In addition, more details should have been applied to the model based on one expert.

7.2.2 Comprehensibility

As it mentioned in section (3.5), comprehensibility of a model is related to understandability and simplicity. Therefore, complexity, understandability, and perceivability in a logical manner were investigated. The answers of these questions could show comprehensibility of the meta-model for the selected experts.

Regarding the complexity of the meta-model, six experts mentioned that the model is not complex. One of them commented that she thinks it is not complex because of her background knowledge about fuel level system. However, one of the interviewees mentioned that no meta model is easy and simple at the first glance, but this meta-model was not more complicated that another meta-models for him. Another expressed that the analysis of the model was complex for him. In addition, one other expert mentioned that the model was complex in his opinion. Therefore, it can be concluded that the model was complex for 3 and was not complicated for 6 experts that interviewed.

Regarding the understandability, all the experts told that the explained meta-model was understandable. In addition, the experts mentioned that the model could be realized logically.

*Result:* The meta-model was understandable and could be perceived logically for the target group. Moreover, the majority of experts mentioned that the meta-model was not complex.
7.2.3 Expressiveness

Expressiveness of the meta-model was investigated in two aspects: if semantics could be interpreted as they were described, and if the meta-model could express the same concepts and relationships in embedded systems.

For semantics, there were some discussions about some terms such as "allocation element" and "user function" definition and the need to find a more specific term for these concept. In addition, there was a comment for specifying different perspectives for more clarity between different aspects of the meta-model. Moreover, there were some questions about the term 'cable'. These comments were applied in the very early interviews. Subsequently, the rest of interviewees(7) confirmed that the semantics are properly used in the way that they were explained. Besides, one expert expressed that there must be a definition for each concept or a background of fuel level display in order to have the same interpretation of semantics.

In order to have a general meta-model in embedded systems area, the meta model should express the same concepts and relationships that exits in general embedded systems. The result of asking this issue, one of the experts at Scania indicated that he was not sure if the same concepts are used out of Scania. Another expert at Scania mentioned that the term 'function variable' might be a concept specific for Scania. However, the other experts out of the company understood function variable as a general term, and confirmed that the meta-model is expressive for embedded system design.

Result: After discussions with the experts, some specific terms such as 'allocation element', 'user function', and 'cable' were more specifies and changed to 'abstract functional unit', 'customer function', and 'communication link' respectively. After applying the modifications to the meta-model, and continuing with other interviews, the experts confirmed the compatibility of semantics with their description as well as the expressiveness of the meta-model in embedded system domain.

7.2.4 Generality

The next investigation was about generality of the meta-model. The experts were asked if the meta-model was general enough for fuel level system or embedded system design. One of the experts commented that the meta-model was general for Scania because of the usage of specific terms from Scania. Based on this comment, the meta-model was revised (refer to the changes on 7.2.3) and the interviews were done. Among the rest, six mentioned that the meta-model was general for embedded system design. One of the interviewees mentioned that it was general for the fuel level display system. This interviewee was expert in modeling of fuel level system. Another expert indicated that it was general for automotive industry where communication link (cable) was an important component, while communication link (cabling) would not be important for another high-tech embedded systems.
**Result:** Based on the discussions, all mentioned that the meta-model was general. However, the domain of generality was varied from fuel level system (one expert), Scania (one expert), embedded system design (six experts), to automotive embedded system (one expert). In conclusion, the responses indicated that the meta-model is general for automotive embedded system design.

### 7.2.5 Usefulness

At the end of the interview, usefulness of the meta-model for system design was questioned. This question was asked in eight interviews. Five of the experts thought that this model could be useful for system design. Three of the interviewees mentioned that general model should be developed or implemented as a software in order to find out the usefulness.

**Result:** More than half of the interviewees believed that the meta-model was useful. The rest thought that the meta-model required to enter in the development or implementation phase in order to be judge for usefulness.

### 7.3 General Comments

According to the experts, this kind of modeling is very important for the industry. During the interviews, there were comments for making more details models and also examining the meta-model. Therefore, the fuel level model was redesigned based on proposed meta-model (Appendix C). Other subjects such as differentiating among different perspectives of the system, generalizing of the concepts, and defining instance interfaces such as pin and ports, and so on were discussed with the experts, applied, and got their confirmation. Finally, this meta-model can be used for further design purposes such as safety considerations.
8.1 Conclusions

One of the important pre-requisites for the design and development of a complex embedded system is to have a meta-model for information consistency and communication purposes. In addition, the design of a multi-level structure can assist to cope with complexity, improve requirement traceability, and facilitate verification. According to investigations in this research, multi-level structuring and meta-modeling of embedded systems are the challenges that system engineers face in automotive industry. This research aimed to provide a multi-level structure for fuel level display system and a general meta-model that can be used in automotive embedded system design.

Through studying a real system in industry such as fuel level display (FLD) system, a multi-level structure of fuel level system was outlined. This study included modeling of FLD system structure for design artifacts such as requirements, functions, and specifications in SysML and Modelica. When the multi-level structure for FLD system design was prepared, it was possible to add verification concepts and verify the resulting model against the high-level requirements. Afterwards, the Modelica model was simulated and the validation result was depicted. The validation results showed the situations in which the FLD requirements were violated by the FLD system. A case study and simulation approach was considered in this case.

Subsequently, a meta-model for automotive embedded system for design artifacts was provided. It was designed in five different perspectives: user, logical, technical, and requirement perspectives. Subsequently, it was evaluated by the means of interviews with experts in the field. Moreover, this research considered five criteria for the interviews: correctness, comprehensibility, expressiveness, generality, and usefulness.

Interviewees should have had a good knowledge of information modeling and development. Otherwise, it might cause disregarding the existing problems in the
model, addressing wrong comments, and leading to incorrect modeling. Therefore, the target group were chosen from experts who had experienced or researched in information modeling, and development. A majority of them had more than five years of work experience. Therefore, they were suitable for this interview.

As for correctness of the meta-model, most of interviewees mentioned that the meta-model was correct and some claimed that models could not be correct or incorrect. For comprehensibility, they mentioned that the meta-model was understandable and can be perceived logically, while it was complex for some of them. Therefore, the model was comprehensible for the majority of the experts. The discussions regarding semantics and expressiveness caused some changes on the terminology. Subsequently, the experts confirmed expressiveness of the meta-model in the embedded system domain. For generality, the meta-model is general for automotive system. In addition, most of interviewees confirmed that the meta-model was useful, while others recommended that the meta-model should be developed or implemented in a tool in order to judge for usefulness.

8.2 Future Work

The result of this research proposed a framework for future researchers and designers in automotive embedded systems. Here are the possible future works that could be performed in continuation of this research.

1. Validation of the meta-model by a larger sample size and statistical method. The statistical method and quantitative analysis can validate the result of this study in a wider level.

2. The development and implementation of a software tool based on the proposed meta-model in order to assist embedded system design.

3. Addition of safety requirements features to the meta-model and to the multi-level structure in Modelica. Nowadays, automotive industry are moving towards certification of their products through safety standards. The structure in simulation environment makes it possible to develop and test safety related requirements and risks. As a result, the meta-model can be extended by adding appropriate risk concepts. Such a research may enrich the results of this project by demonstrating the importance of conceptual modeling and multi-level structure in risk analysis.

4. Simulation of this structure in Modelica with real data and Hardware-in-the-loop (HIL) simulation technique. When the project is tested in a real-time simulation, the structure could be applied in real FLD system.
5. Application of other aspects to meta-model. This research looked at the structure of requirements, functions, and specification, however, other researchers could target other aspects such as configuration management, and variants.
References


automation systems, I. and integration (2005), Product data representation and exchange, ISO.


Bhattacherjee, A. (2012), SOCIAL SCIENCE RESEARCH: PRINCIPLES, METHODS, AND PRACTICES, 2 edn, Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License; Florida.


Björn, A. (2009), Beskrivningsguide/produktnedbrytning, Technical product data, Scania.


URL: http://www.3ds.com/products/catia/portfolio/dymola

URL: http://www.east-adl.info/


URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4384163


Johannesson, P. and Perjons, E. (2012), ‘Science Primer (draft)’.


URL: https://www.modelica.org/events/modelica2012/authors-guide/authors-guide


URL: http://www.omg.org/mof/


URL: http://www.omgsysml.org/

OpenModelica (2012). Available, Online.

URL: https://www.openmodelica.org/


Perssson, S. (2009), Begreppsamling for resa, Technical product data, Scania.


**URL:** [http://www.plcs-resources.org](http://www.plcs-resources.org)


**URL:** [http://www.omg.org/spec/XMI/2.4.1/PDF/](http://www.omg.org/spec/XMI/2.4.1/PDF/)
Expert Interview

Welcome
Thanks a lot for your time and participation. My name is Sara Sadeghi. I am working for my thesis project at Scania. The purpose of this interview is to reflect experts comments on the empirical part of my thesis at KTH. I would like to confirm with you following issues before starting:

- I am going to record the interview beside taking notes not to miss any of your comments in the interview.
- I will delete completely the record and notes after I finish my thesis report.
- Your responses will not be identified with you and your affiliated company in the report.
- I will send a summary of our interview and will check with you before further use.
- I expect the interview takes around 45 min.

Now, I will explain briefly about my project, if you do not have any question.

Introduction
Nowadays, the use of models in product development are getting more and more popular. Model based development defines the process and rules for system modeling with the aim to reduce the overall cost and time of projects, to improve communication among engineers from different domains and to facilitate integration between systems in the domain of interest. Therefore, Object Management Group recommends using of meta-models to satisfy mentioned goals. This project aim to have a meta-model for the structure of Requirements, Specification, Architecture, and Functions for system design.
In order to get a better understanding of the embedded systems and practice system engineering process, the case of Fuel level display system has been studied and designed at Scania. Then, a general model based on the studied system was designed. Later, the model was discussed with engineers at Scania. In this stage, there is an interest on my thesis to show how this meta model can be generalized and also validated/improved based on experts’ comments in other companies/research areas.

A.1 Background

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

(involved in design models, design meta models, use models, use meta models, development, other)

A.2 General Information

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?
2.2. Does your organization follow any meta models for modeling and development?
2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)

A.3 Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

A.4 Factors

Correctness-Syntax

4.1. Do think that the model is correct?(ask for explanation in the case of being incorrect)

Generality
5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for E/E embedded system/Fuel Level Display system design?

Comprehensibility

6.1. Can the explained meta model be perceived logically? (ask for elaboration in the case of negative response)
6.2. Do you consider the explained Meta Model complex?
6.3. Is the explained Meta Model understandable?

Semantics

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?
7.2. Is there any other interpretation for the concepts used in this model?

Expressiveness

8.1. Does this meta model describe the same concept in embedded system products?

Usefulness

9.1. Do you think this Meta Model can be useful for embedded system design?

Final Thoughts

Those were all of the questions that I wanted to ask. Do you have any final thoughts about the other aspects that effect the generality of meta model and should be considered?

Review

Thanks again for your attention and comments. I will send you the summary of this talk in order to have your confirmation. I wonder if I can call or email you back in the case that I have any question in this regard.

A.5 List of Interviewees

Dr. Olena Rogovchenko, Liköping university
Thomas Näreskog, Scania
Jonas Biteus, Scania
Dr. Dag Bergsjö, Chalmers University
Jan Soderberg, Systemite
Lic. Mattihas Biehl, KTH Royal Institute of Technology
Dr. Niklas Adamsson, Accenture
Jonas Westman, KTH Royal Institute of Technology and Scania
Lic. Gunnar Berg, Scania

**Formal Meetings for Functional Architecture**
Jonas Edén, Scania
Ola Ramström, Scania
Thomas Näreskog, Scania
Equations

Here are the equations related to AER 201 in section 6.2.1:

\[
y_s(t + T_s) = \begin{cases} 
x_in(t), & \text{during start – up} \\
(1 - C_{pre})x_in(t) + C_{pre}y_s(t), & \text{otherwise}
\end{cases} \quad (B.1)
\]

where:
\(y_s = \text{filtered fuel level} \ [m^3]\)
\(T_s = \text{sampletime} \ [s]\)
\(x_in = \text{measured fuel level} \ [m^3]\)

Kalman algorithm is used for signal processing. Following equations represent Kalman algorithm.

\[
x_{\text{start}} = \begin{cases} 
y_s(t), & |y_s - \hat{x}_{\text{old}}| > 0.1X_{\text{tot}} \text{ or } y_s > 0.9X_{\text{tot}} \\
\hat{x}_{\text{old}}, & \text{otherwise}
\end{cases} \quad (B.2)
\]

\[
\dot{x}(t + T_s) = \begin{cases} 
x_{\text{start}}, & \text{during start – up} \\
\dot{x}(t) + T_s u(t) + K(y_s(t) - \dot{x}(t)), & \text{other}
\end{cases}
\quad (B.3)
\]

\[
y(t) = F(\hat{x}(t)) \quad (B.4)
\]

where:
\(y_s = \text{measured fuel level} \ [m^3]\)
\(\dot{x}_{\text{old}} = \text{fuel level at last shutdown} \ [m^3]\)
\(X_{\text{tot}} = \text{total fuel volume} \ [m^3]\)
\(T_s = \text{sampletime} \ [s]\)
\(u = \text{fuel consumption} \ [m^3/s]\)
\(\dot{x} = \text{estimated fuel level} \ [m^3]\)
\(K = \text{feedback gain} \ [-]\)
\(F(x) = \text{function converting } m^3 \text{ to corresponding } \%\)
\(y = \text{total fuel level} \ [%]\)
\(x_{\text{start}} = \text{start state} \ [m^3]\)
Different perspectives of FLD in Enterprise Architecture Sparx Tool based on the designed meta-model are depicted here.
Figure C.1: Requirement Diagram
Figure C.2: Block Definition Diagram - Logical and User Perspectives
Figure C.3: Internal Block Diagram - High Level Functions of Logical Perspective(Up), and Refined Logical Perspective(Down)
Figure C.4: Block Definition Diagram - Technical Perspective
Figure C.5: Internal Block Diagram - Technical Perspective
Figure C.6: FLD Packages
C.1 Verification

Figure C.7: General Model For Verification

Figure C.8: Verification Package
Transcripts of Interviews

This chapter contains the transcripts of the interviews. The questionnaire is in Appendix A.

D.1 Interview 1

Background
I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee The main working area is language design, and real-time systems (development), and more than five years experience.

General Information
2.1. Does your organization follow any meta models for modeling and development?

- Interviewee No.

2.2. Do you follow system engineering process steps at your organization?

- Interviewee No particular process. We are working on small projects.

Meta Model
Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.
- Interviewee For the structure of the component, SW, HW and allocation element, I am clear and it is good. To be sure the details are right; I want to see how FV are supporting connectors.

Factors
Correctness-Syntax

4.1. Do think that the model is correct?
- Interviewee Yes, I think is good.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?
- Interviewee To me it is general.

Comprehensibility

6.1. Can the explained meta model be perceived logically?
- Interviewee Yes.

6.2. Do you consider the explained Meta Model complex?
- Interviewee It is good for me because I know fuel level display system.

6.3. Is the explained Meta Model understandable?
- Interviewee You crossed different levels of abstraction. I understand. I think you have to have an example ready when you present the meta-model.

Semantics

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described? Is there any other interpretation for the concepts used in this model?
- Interviewee AE is very specific to Scania and other names are pretty generic. So, one thing that I would rename is the Allocation Element and anytime I look at, I forget what it means. A more generic name can be "abstract functional unit".

Expressiveness

8.1. Does this meta model describe the same concept in embedded system products?
- Interviewee AE needs to be changed (Refer to 7.1.).
D.2 Interview 2

Background
I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee I am working as a researcher in information modeling and Product Life Cycle Management, for seven years.

General Information
2.1. Does your organization follow any meta models for modeling and development?

- Interviewee Yes.

Meta Model
Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee Did not you consider the attributes?

- Interviewer: No, this is a conceptual model.

Factors
Correctness-Syntax
4.1. Do think that the model is correct?

- Interviewee It makes sense in the high level that I am familiar.

Generality
5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?

- Interviewee I would say that this is a general model at Scania because you use Scania’s specific names in it.

- Interviewer: If I consider general terms for user function and allocation elements, will the model be general?
- **Interviewee** Yes, the idea is general but I think you should name entities slightly differently. However, outside Scania, you could call them customer function instead of user function and sub function or allocation function instead of allocation element.

**Comprehensibility**

6.1. Can the explained meta model be perceived logically?

- **Interviewee** Yes, I think so.

6.2. Do you consider the explained Meta Model complex?

- **Interviewee** No. You have tried to make it simple. Even it is possible to make it more simple, but I think it is simple.

6.3. Is the explained Meta Model understandable?

- **Interviewee** I think it is possible to read it without some examples. I think it is understandable.

**Semantics**

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?

- **Interviewee** Yes, I think so.

7.2. Is there any other interpretation for the concepts used in this model?

- **Interviewee** No.

**Expressiveness**

8.1. Does this meta model describe the same concept in embedded system products?

- **Interviewee** I am not sure about the answer of this question. However, you can look at some ISO work that has been done, to get the sense what kind of names do they use.

**Usefulness**

9.1. Do you think this Meta Model can be useful for embedded System design?

- **Interviewee** Yes, I think so.

**Final Thoughts**

- **Interviewer**: Those were all of the questions that I wanted to ask. Do you have any final thoughts about the other aspects that effect the generality of meta model and should be considered?
- Interviewee It is very good to test the model. That’s the way you know that is going to work or not. For example to test it in a product management system then you will realize the things that are not required, or not working: for example using DOORs or other tools at Scania. Maybe if you put everything to work it become more complex.

D.3 Interview 3

Background
I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee I have more than 20 years as a system architect, both in development and modeling.

General Information
2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- Interviewee It is important for communication among the developers in a team.

2.2. Does your organization follow any meta models for modeling and development?

- Interviewee Yes, modeling is used along with development.

2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)

- Interviewee No particular process. We were a small group working on this project.

Meta Model
Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee I would say that AE is a sort of Function. It gives an answer. It does not need to be SW. It can be a networks of sensors and switches and relays could be some
sort of AE. But you will not see that too often. You would surely not seen AER for those.

AE s can change from one vehicle to the other vehicle.

AE talks with FVs through different sort of ports could be internal ports which is structural variable, or could be over cable.

Port can be the level of fuel in the tank. Like a physical boundary between air and diesel fuel. Then tank sensor is part of SESAM and that is interacting with fuel pressure in the environment.

Not all the components have system description before but I am not sure any more.

In one vehicle the relation between the UF and AE is many to one relationship. This is true for all vehicles. In one vehicle AE can be allocated to COO and in the other vehicle it is connected to gearbox. But one AE cannot be allocated to both gearbox and COO at the same Vehicle.

A function variable consists of several signals. Signal can be anything: smoke signal, voltage, sound, etc. one FV can carry signals on different media.(one FV can carry CAN signal and Smoke signal at the same time.) FV is not a single variable. FV is a set of the Signals.

To me port is something specific. Either might be like a plot or might be IP number.

Interface can be a HW interface IP cable, cane be a pressure gage, ...

HW interface can be an analog digital sensor, or a light with 12 voltages. Interface is much easy than port. Port is more abstract and relates to many things that exist.

Factors

Correctness-Syntax

4.1. Do think that the model is correct?

- Interviewee Yes, it is. It is an abstract model.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?
- **Interviewee**: This is quite general. Components can be anything from lamp to ECU. We would like to see the network of components that we can see at your model.

**Comprehensibility**

6.1. Can the explained meta model be perceived logically?

- **Interviewee**: I understand the concepts.

6.2. Do you consider the explained Meta Model complex?

- **Interviewee**: It is not at all complex. It is simple.

6.3. Is the explained Meta Model understandable?

- **Interviewee**: I understand like this: Interface is realized through a FV. FV is something very abstract. AE communicates with FVs. A complement set of FVs is an interface to AEs. In this sense they are exchangeable.

**Semantics**

7.1. Can semantics (meaning of the concepts/entities) be interpreted in the same way as they were described?

- **Interviewee**: Yes.

**Expressiveness**

8.1. Does this meta model describe the same concept in embedded system products?

- **Interviewee**: Yes

**Usefulness**

9.1. Do you think this Meta Model can be useful for embedded system design?

- **Interviewee**: It could be useful. It should be tested.

## D.4 Interview 4

**Background**

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- **Interviewee**: Around 11 years working experience in simulation and modeling.
General Information

2.2. Does your organization follow any meta models for modeling and development?
   - Interviewee There is no meta-model. We are developing a meta-model.

2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)
   - Interviewee The Scania System Engineering process model.

Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.
   - Interviewee It is better to add directions for relations among the objects. There is not formalized function at Scania.

Factors

Correctness-Syntax

4.1. Do think that the model is correct?
   - Interviewee We cannot say it is correct or not correct. It depends on your objectives. It is quite general. I do not see anything wrong with it.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?
   - Interviewee Yes, I think it is general; you should be able to use it.

Comprehensibility

6.1. Can the explained meta model be perceived logically?
   - Interviewee Yes.

6.2. Do you consider the explained Meta Model complex?
   - Interviewee No.

6.3. Is the explained Meta Model understandable?
   - Interviewee I understand.
Semantics

7.1. Can semantics (meaning of the concepts/entities) be interpreted in the same way as they were described?

- Interviewee 'System component' is better to be 'component' you know that is system and you do not need to call it system component.

7.2. Is there any other interpretation for the concepts used in this model?

- Interviewee 'Port' can be interface in Scania.

Expressiveness

8.1. Does this meta model describe the same concept in embedded system products?

- Interviewee Yes.

Usefulness

9.1. Do you think this Meta Model can be useful for embedded system design?

- Interviewee You have to try it. This is very general but you cannot say if it works or not unless you apply it to a system.

D.5 Interview 5

Background

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee We have two platforms and it is a generic platform and it has support for collaborative engineering, support for version-ing configuration management and apart from that the meta model is programmable, working with the EAST-ADL language. EAST ADL is for automotive development, and modeling of the system is divided to different abstraction levels. One abstraction level is feature model, which is typically for the market department. It is an abstraction level to define the architecture, what each feature should be and what feature variant should be and so on.

The next level is to define an analysis model. It is a block like model. We have different blocks which communicate. These blocks are logical analysis functions and they are similar to Simulink.

The next level is the design level. We model all application SW components. So, this in ADL that focuses on the structure of your system.
In addition to this there are some modules, for example requirement modules that implement different levels. Requirements can be in different formats, for example requirement for the use case models or the functional models and so on. And each module is a support module of ISO 26262. It will be connected to your model. This model is quite complex. Check error propagation in the model and how failure propagate to each component.

The failure model should be possible to reuse in an efficient way for version or variant of that system.

I have working experience in embedded systems development for more than 15 years.

General Information

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- Interviewee: We use Meta Models for the customers and for quality check.

2.2. Does your organization follow any meta models for modeling and development?

- Interviewee: We do only this for the customers.

2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)

- Interviewee: Agile method is used for our software development, which is more feature-driven.

Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee: You have to define if this is a conceptual model or a Meta Model that is managing the other models in the system.

- Interviewer: This is a conceptual model that describes components and their interconnections.

- Interviewee: Many to many relationships: If you want to implement the model, you have to remove many to many relationships. Otherwise, changes are costly.

- Interviewer: Implementation steps can be considered in the next phase of the project. When there are agreements on the concepts and the relations among them.
- **Interviewee** Multiple instances: If you are allowed to have multiple instances of the SW or HW component, you would have distinguished between your interfaces and the ports. Meaning that, the Interfaces are the types of the ports of the SW and HW. Therefore, when you do this little connecting, you need to distinguish between instances of the ports of a SW.

- **Interviewer**: Different instances for HW and SW will be considered in the modified model.

**Factors**

**Correctness-Syntax**

4.1. Do think that the model is correct?

- **Interviewee** You catch the most central information. All the details can be justified and there are some cases that you would need them. However, here is the main minimal thing.

   The relation between the SW component and HW component: You could also have containment structure. Any SW component can connect to any other SW component. However, since the components can be composite. You could connect between different levels of components, so you will not have pure containment of your component structure.

- **Interviewer**: This is a self-containment structure. A HW component should be the container of the SW component/functions that will be modified.

**Generality**

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?

- **Interviewee** I think it is generic.

**Comprehensibility**

6.1. Can the explained meta model be perceived logically?

- **Interviewee** I understand it perfectly.

6.2. Do you consider the explained Meta Model complex?

- **Interviewee** No. Meta-Model has some limitations. When you add more details, the meta-model will become more complex and there is a tradeoff between how complex should it be and how accurate should it be.

6.3. Is the explained Meta Model understandable?
- **Interviewee** Yes.

**Semantics**

7.1. Can semantics (meaning of the concepts/entities) be interpreted in the same way as they were described?

- **Interviewee** Yes, they are very generic.

7.2. Is there any other interpretation for the concepts used in this model?

- **Interviewee** No.

**Expressiveness**

8.1. Does this meta model describe the same concept in embedded system products?

- **Interviewee** Yes.

**Usefulness**

9.1. Do you think this Meta Model can be useful for embedded system design?

- **Interviewee** My opinion is that the meta-model would need some more details in order to become useful, by some good tool implementation that supports needed modeling constraints.

**D.6 Interview 6**

**Background**

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- **Interviewee** I am a master of computer science and PHD candidate in Mechatronics. I am working on modeling of embedded system and modeling parts of embedded systems that support the development of embedded systems for 4 years. (Specially work in EAST-ADL), In East-ADL there are similar meta classes.

**General Information**

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- **Interviewee** When we want to build the language, we use Meta Models. Otherwise, models can be used for model checking, verification activities constructively.
2.2. Does your organization follow any meta models for modeling and development?

- **Interviewee** Yes.

2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)

- **Interviewee** In development of embedded system, usually people use V-model.

**Meta Model**

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- **Interviewee**

  I think this is a good model. For modeling sometimes, there is no right and wrong, there are just different views and based on those views the model will be different.

  About the modeling style: Interface does not have direct connection with component. You can have overwriting.

  Do 'SW Interface' and 'HW Interface' have different attributes? if they have different attributes it make sense.

- **Interviewer**: Yes, they have different attributes.

- **Interviewee** It is better to consider arrows in the model.

  For multiplicities between 'sub function' and 'Customer Function': Then you modeled cross containment functionalities and that’s why you did not model containment relationship. The same reasoning can be between the 'sub function' and the 'component'.

- **Interviewer**: The model has self-containment structure.

- **Interviewee** Is it the 'component' or the 'component type'? The component can be in one place. Types can be re-used.

- **Interviewer**: Yes, it is component type.

**Factors**

**Correctness-Syntax**

4.1. Do think that the model is correct?

- **Interviewee** Yes.
Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?

- Interviewee Ya, I think it can be considered as a model for an embedded system. In addition, the border and view of generality should be defined properly. Therefore, It is hard to answer this question about generality and it should be specified where it used and in which scenarios it used.

- Interviewer: Do you think that the model can be general in the area that I am working especially in busses and trucks?

- Interviewee I see that there is nothing specific for trucks. At a certain level, it can be used for buses, trucks, and cars.

Comprehensibility

6.1. Can the explained meta model be perceived logically? Is the explained Meta Model understandable?

- Interviewee Yes, with the introduction you gave the model is understandable.

6.2. Do you consider the explained Meta Model complex?

- Interviewee No.

Semantics

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?

- Interviewee Yes.

7.2. Is there any other interpretation for the concepts used in this model?

- Interviewee Of course, there are always other interpretation but you are the one that defines the semantics. As you are the creator, you can define it. You do not need to worry about it in my point of view.

Expressiveness

8.1. Does this meta model describe the same concept in embedded system products?

- Interviewee They are also used the same.

Usefulness

9.1. Do you think this Meta Model can be useful for embedded system design?
- **Interviewee** Yes, there are certain tasks where it is helpful. Which task of system design do you want this model works for?

- **Interviewer:** Simulation.

- **Interviewee** Yes, I think it can give the overall impression of where things are located and how thing are allocated and more often architectural view.

**Final Thoughts**

Those were all of the questions that I wanted to ask. Do you have any final thoughts about the other aspects that effect the generality of meta model and should be considered?

- **Interviewee** Making an instance model would be valuable.

### D.7 Interview 7

**Background**

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- **Interviewee** Master of Science on Vehicle engineering product development (2002), PhD in integrated product development and Mechatronics (2007). My research was in the aspects related to Mechatronic embedded software in traditional manufacturing industries. My focus was on organization aspects, process aspects and technology aspects such as product life cycle management but also how the product and architecture could influence the way you were working (General view of mechatronic development). After graduation worked with modeling and product modeling in both Hardware and embedded software in industry and telecom and high-tech. Mainly with data modeling and product life cycle.

**General Information**

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- **Interviewee** In Mechatronic sector, the modeling or the product modeling aspect is one of the key concerns of the organizations. It secures the consistency of the products in the way that they describe them,

In order to implement supporting technology such as software configuration management system, product life cycle management system, application life cycle management systems, the functions and the features are the key concerns.
If you do not consider that in place, you will end up rather with inefficient operating model or the organization. It really does not matter which industry it is (automotive, electronics, and so on) but the definition of the core product model is the key.

2.2. Does your organization follow any meta models for modeling and development?

- Interviewee Yes.

2.3. Do you follow system engineering process steps at your organization? (if not, can you clarify which approach do you use?)

- Interviewee Of course everyone uses a process and it depends on the industry how formalized and strict it is. An example might be regulation for safety etc, which requires a strict and formalized working process.

Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee I assume that requirements are out of scope of your model. I would say, distinct better between the functional and logical perspective. Customer function is functional and sub function is the logical dimension. Concentrate on customer variants. Whether it is parametric based or is logical or what do they use as criteria for selecting different customer variants. What parameters are configuring the products based components.

Factors

Correctness-Syntax

4.1. Do think that the model is correct?

- Interviewee A model can never be correct or incorrect. A model is always a model. The definition is that a model represents one view of reality. This definition can never be correct. There is no such a thing as correctness in this way. Depending the scope you can see how consistence or usable it is.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?
- Interviewee: Yes and no. This model is on automotive industry and then cable is important component. In another high-tech embedded system, cabling would not be such important. Most of them are in the same level of detail except cable that is in more details. However, I know that cable is important for automotive and vehicles. It makes the complete model not as general as it would be.

**Comprehensibility**

6.1. Can the explained meta model be perceived logically?

- Interviewee: Yes. About the naming of the component, I would prefer 'item' instead but the component is at the same abstraction level.

6.2. Do you consider the explained Meta Model complex? 6.3. Is the explained Meta Model understandable?

- Interviewee: Of course, there is complexity in the model. This is like correctness and depends on who look at this model. For example, for a model engineer is not complex but for hardware engineer is complex. For me is understandable. I have been working with Mechatronic, products, and data modeling. It depends on the experience and the background of the engineers for answering such questions.

**Semantics**

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?

- Interviewee: Yes, I think so. It comes back to the customer function and subfunction that we discussed.

7.2. Is there any other interpretation for the concepts used in this model?

- Interviewee: No, I think it is clear.

**Expressiveness**

8.1. Does this meta model describe the same concept in embedded system products?

- Interviewee: Yes.

**Usefulness**

9.1. Do you think this Meta Model can be useful for embedded system design?

- Interviewee: Yes, to some extent. I think what is missed here is the architectural perspective of the products. The architecture from functional, the software and, hardware point of view is missed here. This point is not clear for me. Yes, as long as you do not consider the architecture point of use. If the model is for conceptual design level, you do not consider architecture, and it is useful. If you consider architecture constrain you will have a limitation.
Final Thoughts

Those were all of the questions that I wanted to ask. Do you have any final thoughts about the other aspects that effect the generality of meta model and should be considered?

- Interviewee This work is very important especially for industry. It is very important to use it in a practice or to find an IT solution that work with models. It is important to differentiate between the logical view and the functional view. For example, customer does not understand what the logical function is and just want to see the features. And also how you do variant configurations.

D.8 Interview 8

Background

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee I took my master in engineering physics. My background to Scania and modeling started from my master thesis from beginning of summer 2011. Basically we need to understand the basic concepts in the standard 26262 in order to map different terms from the standard to the Scania system. Because of this we have to understand the basic architecture structure of systems at Scania.

General Information

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- Interviewee At Scania, until now it has not been a need for a specific and a formal meta model which relates all the elements to each other. But as the functionality grows in the system, increasing complexity, we need to define the system boundaries and interfaces in order to keep track of all dependencies. So, I think Scania become aware of this that’s why they have started doing this kind of analysis.

2.2. Does your organization follow any meta models for modeling and development?

- Interviewee This question was answered in question 2.1.

2.3. Do you follow system engineering process steps at your organization?

- Interviewee Sure. At Scania They have their own process that they use. When they develop the system, it is called system engineering process usually by
definition. It is their own version of system Internal Engineering process. I would not say hay they work against some sort of standard. It look like a version of V-model in theory not always in practice.

Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee: How about ports? Interface usually consists od several ports.

- Interviewer: Yes, the interface are the interface types which means port or pin.

- Interviewee: In other meta-models interface will have several ports, and then they will be linked to interaction point.

- Interviewer: I think you refer to contract based design meta model. Here is the meta-model of the Modelica model. They refer to interface for contraction points.

- Interviewee: I remember that we spoke of where to actually place the SW component. If it is in the logical perspective or technical perspective. I think It is up to you basically how you define the perspectives. How does your reasoning come to this?

- Interviewer: Abstraction Functional Unit or Allocation Element has some refinement that will be byte code and SW component is that code. Byte code can be allocated to HW/technical perspective.

- Interviewee: One of the basic definition of a component is that to have a specific interface. How would you be able to identify a SW interface in a HW structure? Where would it be?

- Interviewer: The interface will be a logical interface.

- Interviewee: Where will the SW be in the Technical perspective?

- Interviewer: SW (refinement of abstract functional unit) will be just bit and bytes that are allocated to HW. I think it is important to know what is the boundary of HW and SW.

- Interviewee: The SW will be in logical perspective. This is an abstract model. It usually depends how do you define the lowest atomic level. If every one see this meta model they would understand it, I think.
Factors

Correctness-Syntax

4.1. Do you think that the model is correct?

- Interviewee In general I think it is correct. If you ask different people they would tell you about small differences from their own perspectives. You will never have a model that everyone will be satisfied with that. Depending on which engineering you ask at Scania the actual implementation is different. Sometimes it will fit and some time more or less fits depending on where it is applied.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for E-E/embedded system/Fuel Level Display system design?

- Interviewee I can consider this as a general model for most parts of E-E design at Scania. However, for some specific parts of it especially in the drive line areas they have different kind of system architecture or I do not really think that this really applies. For fuel level Display system I think it makes sense.

Comprehensibility

6.1. Can the explained meta model be perceived logically?

- Interviewee Sure, for some one who works with this kind of modeling. I mean you need background about design elements.

6.2. Do you consider the explained Meta Model complex?

- Interviewee No meta-model is very easy to understand at the first glance. It is not more complex than the other meta-models

6.3. Is the explained Meta Model understandable?

- Interviewee Yes, I understand it. This is subjective answer.

Semantics

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?

- Interviewee Yes, sure.

7.2. Is there any other interpretation for the concepts used in this model?

- Interviewee No. I had some comments, but they were my own idea.
Expressiveness

8.1. Does this meta model describe the same concept in embedded system products?

- Interviewee Generally yes. But for example in the case of Function Variables, I am not sure that it is commonly used. I think that is unique for Scania.

Usefulness

9.1. Do you think this Meta Model can be useful for fuel level system/embedded system design?

- Interviewee Sure, it is useful. The fuel level is already designed at Scania then it is useful for understandability of fuel level design especially for someone who is familiar with meta model.

D.9 Interview 9

Background

I would like to have general information about you.

1.1. Would you please let me know if you have had during your working time any involvement with concepts such as: development, information modeling, and meta modeling. If yes, for how long?

- Interviewee I am Licentiate from KTH. I work as a Software Architect. I have 20 years working experience. We use home-made Scania modeling language.

General Information

2.1. Considering the sector you are active, how important is to have an updated Meta Model of the systems?

- Interviewee Yes, the dream is to have a complete model of the system especially for change management

2.2. Does your organization follow any meta models for modeling and development?

- Interviewee We are trying to develop a meta-model.

2.3. Do you follow system engineering process steps at your organization?

- Interviewee Yes, we have a process. It is not a check list it is more agreement within organization. Maybe it is the best to have a strict formal process, we do not have that. We have check points for deliveries. Deliveries are fixed, but how to get there is more flexible. Most of the people in our group are experienced. We have ideas how to do it. There is internal process.
Meta Model

Now I explain about the meta model that I have designed:

3.1. I would like to know if you have any comment or question about the explained model.

- Interviewee I think the structure is basically ok. Most of thing you say is perfectly right.

Factors

Correctness-Syntax

4.1. Do think that the model is correct?

- Interviewee It is correct.

Generality

5.1. This model is designed based on the main object used in Scania systems, specially on fuel level display system. Do you think that this meta model can be considered or work as a general model for embedded system/Fuel Level Display system design?

- Interviewee This is quite a general model. This does not anything specific about fuel level display. I think we could put any of our functions in this model. It should fit in a general way.

Comprehensibility

6.1. Can the explained meta model be perceived logically?

- Interviewee I think so.

6.2. Do you consider the explained Meta Model complex?

- Interviewee Part of it would be quite complex even if it looks quite simple. If you start thinking about it, behind things will be complex.

Semantics

7.1. Can semantics(meaning of the concepts/entities) be interpreted in the same way as they were described?

- Interviewee Yes, I think.

7.2. Is there any other interpretation for the concepts used in this model?

- Interviewee No. If you present it to some one else out of Scania, I wonder they understand function variable?
- **Interviewer** Yes, they understood.

- **Interviewee** I think it needs some explanation. It is not intuitive. If people have not heard this term, it might be confusing.

**Expressiveness**

8.1. Does this meta model describe the same concept in embedded system products?

- **Interviewee** Yes, I think.(explained in previous question.)

**Usefulness**

9.1. Do you think this Meta Model can be useful for embedded system design?

- **Interviewee** Yes, I think you can use it. If you have this generalize way, then you specialize it into your function, I think it can be useful.