Supporting ISO 26262 with SysML, Benefits and Limits

Pierre David, M. Shawky

Supporting ISO 26262 with SysML, Benefits and Limits

P. David & M. Shawky
Heudiasyc, UMR CNRS 6599, Université de Technologie de Compiègne, Compiègne, France

ABSTRACT: This article deals with the issue of deploying efficiently the ISO 26262: the new standard in automotive systems development. The directives enclosed in this norm demands the establishment of a product lifecycle fully integrating the safety assessment activities. To tackle this subject, this paper explores the way of setting up Model-Based Design methodology to express and organize the concepts manipulated during the ISO 26262 process. This attempt is founded on the use of SysML and on the creation of a profile dedicated to ISO 26262 development context. We provide an introduction to Model-Based Design paradigm and its application in a safety relevant context. An overview of ISO 26262 is given, followed by the description of an ongoing project on the subject. Modeling propositions are formulated and the use of diverse SysML diagrams are mapped on the automotive safety lifecycle process.

1 INTRODUCTION

The increasing criticality of embedded systems mission in automotive industry raises safety mastering as a key issue for future road vehicles development. The functionality allocated to such systems concern driver assistance, passive and active safety and vehicle dynamics control, therefore the role of safety analysis during system development continuously grows. To tackle these challenges, automotive industry partners currently set up the ISO 26262 standard, detailing an automotive safety lifecycle supporting the development of road vehicles. This standard built upon IEC 61508, focuses on Electric/Electronic (E/E) Systems but provides a general framework for safety-related systems design. The efficient deployment of this standard within automotive companies is a tedious and crucial task in order to maintain the competitiveness of these organizations on the future automotive market. Therefore, several current projects, as the European funded initiative CESAR[1] and the French project SASHA[2], aim at defining effective tool platforms in order to support the execution of ISO 26262 directives. As members of these two research groups, we work on finding the adequate modeling practices to support the activities of ISO 26262.

The engineering processes involved by the application of such standards, impose on designers to use well-formed methodology that provides efficient and verifiable results for system design and validation. The development environment to set up must show the following characteristics:

- Founded on expressive system representations,
- Founded on unambiguous system representations,
- Providing traceability and configuration management capabilities,
- Showing consistency between system views,
- Providing verification and validation capabilities,
- Supporting the follow-up and respect of costs and time to market,
- Supporting knowledge capitalization and transfer,
- Supporting documentation on the system (e.g. for certification).

The Model-Based System Engineering (MBSE) is nowadays developed to match these expectations (Friendenthal et al. 2008). Therefore we will propose in this article a study on using the SysML language (OMG 2008), which constitutes one of the best languages for MBSE (Estefan 2008), for supporting the process of ISO 26262. We will emphasis on how using SysML artifacts to model the key concept of ISO 26262, and discuss the adequate diagrams and modeling techniques to use to support the various phases of the design cycle.

[1] CESAR: http://www.cesarproject.eu/
[2] SASHA: http://www.pole-moveo.org/

The remaining of this article is organized as follows: in the first section we will make a short presentation of ISO 26262 dedicated to inform the reader with the specific tenets of this very new standard. We will continue this introduction with the indication of SASHA project aims. We will introduce, in section 3, the previous works manipulating MBSE for dependability-critical systems design and validation. Then, in the fourth section, we will underline and detail the key concepts of ISO 26262 and their implementation with the SysML notations. In the fifth part, we will discuss the utilization of the SysML diagrams to support the achievement of ISO 26262 phases.

2 OVERVIEW OF ISO 26262

Currently developed by automotive industry partners, the ISO 26262 aims at becoming the de facto standard organizing road vehicles conception. The proposed engineering process insists on efficiently considering the overall safety of the future car. Therefore, the document gives guidelines to explicitly manage safety throughout product life. The goal is to ensure the safety of the system being developed, but also to adopt a clear communication around its related activities, in order to demonstrate it has been relevantly taken into account. To achieve such ambitions, the standard focuses on defining the artifacts to produce throughout the design. These artifacts have two functions: describing and characterizing the safety of the vehicle on one hand, and producing evidences that safety goals are fulfilled on the other hand. Therefore the instructions given by the ISO 26262 include various plans, analysis, methods, elements and information to produce and report. The ISO 26262 proposes requirements and processes to identify the risks of failures and to set up measures to reduce them to an acceptable level.

The overview of ISO 26262 is given on Figure 1. This representation shows the development process following a "V" shape from concept phase to production and operation. The hardware (HW) and software (SW) level of product development sub phases are also structured in "V" and synchronized with each other during their execution. All these activities are framed by management activities, supporting processes, ASIL-oriented and safety-oriented analyses and guidelines respectively described in part 2, 8, 9 and 10 of the standard.

The central concept of this standard is to achieve the functional safety of the vehicle. Therefore, it proceeds by analyzing the full system, proposing supplementary functions dedicated to safety and validating the whole (system + components executing the safety functions) to assess that the resulting vehicle is acceptably safe. The key concepts used to support these processes will be presented in more details in section 3.

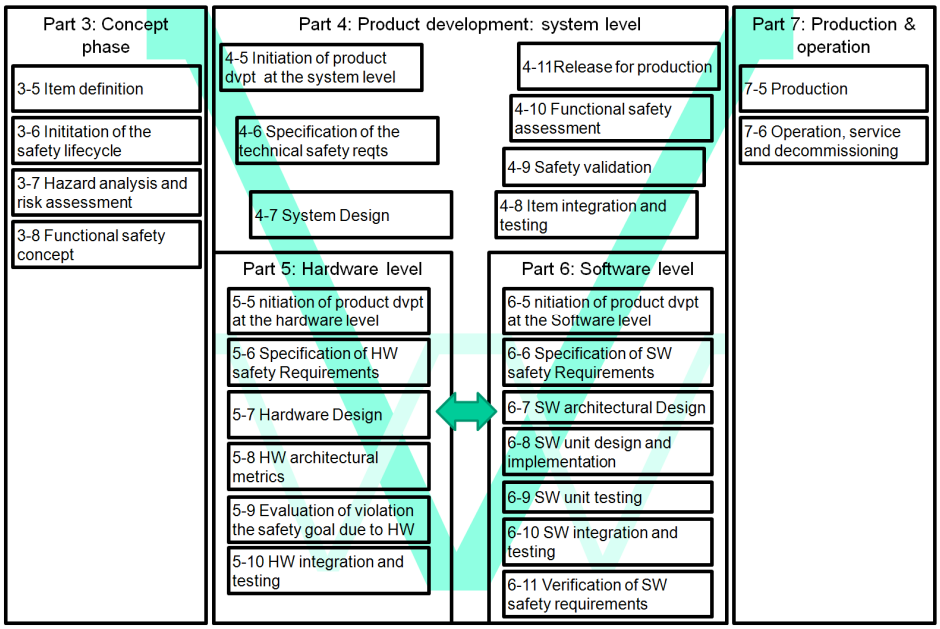

Figure 1. Overview of ISO 26262 Product development process.

The ISO 26262 addresses the overall design phase, it describes activities that demand the intervention of diverse stakeholders of road vehicles conception or production. One main issue that emerges from this situation is the difficulty to master the standard requirements, as well as the correct transfer of information between teams and organizations. Ensuring the right application of the standard, from the car manufacturers, to the electronic component providers, through the OEM (Original Equipment Manufacturer), is a tedious task that needs a clear common understanding and motivation from these very dissimilar partners. The willingness to gather the stakeholders involved in the car design around the application of ISO 26262, is at the source of the SASHA project. This project brings together French representatives of each kind of company participating to car design. Moreover, the consortium benefits from the presence of engineering consultancies and a tool provider. The main objective is to develop tools to support the application of the new safety standards in automotive industry, as well as the methods helping advancing projects while respecting the ISO 26262 requirements. This research project is thus focusing on defining the models that will support this process and the interconnections that can be made between the specific practices of the various specialist teams of the partners.

These preoccupations are topical subjects in nowadays automotive industry. Other research projects as CESAR are also investigating the tools development to sustain ISO 26262 deployment. (Kath et al. 2009) are also pointing out the necessity of supporting ISO 26262 with a consistent tool chain. They expose the *Medini analyse toolchain* devoted to reusable components design. Their communication does not give details on employed models, languages and analysis techniques. Nevertheless, it highlights the necessity of using Model-Based techniques to deploy correctly and efficiently the standard.

3 MBSE BASED ON SYSML FOR DEPENDABILITY

MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases (INCOSE 2007). MBSE enhances classic System Engineering in many domains as communication, preciseness of analysis, results integration or produced knowledge reuse. The SysML language has been specifically defined to support MBSE (OMG 2008). It provides modeling constructs to capture most of systems aspects. It is built on object-oriented principles and shows good abilities for providing well organized and consistent models which are well supporting communication between teams. It also provides modeling constructs permitting an efficient cross identification between system views and a support for Verification and Validation (V&V) activities (Hause 2006). The interested reader can find a presentation of SysML principles in (David et al. 2009b) and a precise proposition of its use for MBSE in (Friedenthal et al. 2008).

3.1 *Adapting SysML with stereotypes and profiles*

SysML, similarly to its parent language UML, provides a stereotyping mechanism permitting a pertinent adjustment of modeling possibilities to projects specific aspects. Moreover, this technique is the basis for profile development, defining adaptation of the language to domain-specific modeling needs. We can mention for example the MARTE profile for real-time applications (Gérard et al. 2007) or the UML/SysML profile for continuous dynamics problems merging UML/SysML and Modelica called ModelicaML by (Pop et al. 2007). Thus, we propose to build a SysML profile for ISO 26262, defining most of the standard notions with SysML artifacts. The profile is primarily built upon a naming policy incorporating the terms used in the standard, then it maps these notions to SysML modeling entities and expresses their features and relationships. This approach, classic for profile construction, is close to the technique employed in (Bernardi et al. 2008) where dependability analysis capabilities are added to the MARTE profile. The goal of such work is to create a modeling language supporting the use of MBSE techniques for ISO 26262 lifecycle realization.

3.2 *SysML and dependability analysis*

Several profiles or UML/SysML extensions for dependability issues modeling can be found in the literature. (Bernardi et al. 2008) propose additional notations to express the dependability attributes of real time embedded systems. Their profile applies for use cases-centered risks analyses, leading to a scenario based evaluation of system dependability. Each component and connector have attached hazards, considering their utilization during the known scenarios, it is possible to compute a risk factor for each use cases described by these scenarios, using Markov Models built from the UML ones. (Zarras & Issarny 2001) realized an UML profile expressing dependability characteristics of software reinforced by OCL (Object Constraint Language) constraints. The HIDE project has valuable contributions on using UML for dependability studies. The project described in (Bondavalli et al. 2001) designed an UML centered System Development environment (SDE) aiming at gathering tools for dependability study.

Our works (David et al. 2009a,b) focused on leading dependability studies from SysML descriptions that were not containing information on the dysfunctional behavior of the system. This was done to prove that it was possible to conduct efficiently dependability analysis, such as FMEA, from a SysML model constituting the central description of a SDE. We showed that such utilization of SysML was in fact enhancing the dependability analysis process and results, in terms of rapidity, consistency and reusability. The resulting method has been called MéDISIS. It is a deductive and iterative approach that includes the following steps:

- Deduction and registration of the dysfunctional behavior with an FMEA, identification of the impacted requirements.
- Construction of a model integrating functional and dysfunctional behaviors with a formal language (e.g. AltaRica Data Flow).
- Analysis and quantification of dysfunctional behavior from the formal model.

4 KEY NOTIONS IN ISO 26262 AND THEIR SYSML EXPRESSION

For a clear application of ISO 26262 standard with modeling tools relying on the exploitation of SysML, an adaptation of the employed vocabulary is recommended. In fact, ISO 26262 proposes a precise vocabulary for all the project artifacts. This glossary is given in the first part of the standard (Vocabulary) and some notions are clarified in the 10th part (Guideline). In order to obtain a relevant follow up of the norm, SysML notations have to be adapted.

4.1 *Defining systems*

The systems tackled by ISO 26262 are complex enough to justify several description levels. The approach proposed by the standard is to progress in the system design by decomposition steps. The system views and analysis are refined into successive granularity models, reducing the complexity of issues to face and progressing towards hardware realization and software coding. The standard defines two main concepts for the system depiction, **item** and **element**:

- **Item**: entire scope under consideration.
- **Element**: any sub-unit of an item.

Thus, an element can be of diverse granularity level, the various kinds of elements:

- **System**: set of elements including at least a sensor, a controller and an actuator.
- **Parts/units**: irresolvable elements, respectively for hardware and software elements.
- **Component**: set of elements neither of system level, nor parts/units. Prefer it to describe sets of specific technology parts/units.

All these notions are differentiated in a SysML model by applying stereotypes to the artifacts representing them.

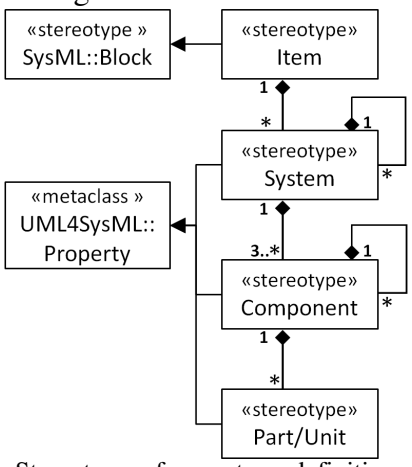

Figure 2. Stereotypes for system definition following ISO 26262 concepts.

We propose (see Figure 2) to define 4 stereotypes: **item**, **system**, **component** and **part/unit**. The last three stereotypes apply to SysML *parts* (beware of the confusion with ISO 26262 hardware **parts**), and the first one apply to SysML *blocks*. Note that it is useless to utilize an **element** stereotype that would "over define" systems, components and parts/units. This construction forces the artifacts stereotyped by system, component and part/unit to have a composite relationship with a *block* and therefore to be SysML *parts*. This is justified since a concept expressed by a SysML *block* refer to a concept out of context, whereas systems, components and parts/units are studied in a precise context within the framework of ISO 26262. Nevertheless, to provide a proper implementation of ISO 26262 these stereotypes must be used with one constraint that could be specified in OCL (Object Constraint Language) or just implemented in the software tool used to model the system: there shall be only one *block* stereotypes by **item** in a model. We also want to highlight the multiplicity of the composite relation between **system** and **component**, which is set to "at least three", and thus imposes on the **system** stereotype to be used for sufficiently important element as prescribed in ISO 26262.

4.2 *Ensuring risks follow up and traceability*

The purpose of ISO 26262 is to provide applicable requirements and processes in order to permit designers to realize the functional safety of their products. Consequently, a great attention is paid in the standard to follow safety goals definition and coverage. The functional safety is defined as the absence of unreasonable risk due to hazards caused by malfunctioning behavior of the system (ISO 26262 part1: 1.51). The overall process to ensure that functional safety will be achieved is constituted as follows:

- Hazard analysis,
- Risk assessment, identify **hazards** that needs risk reduction,
- Formulate a **safety goal** for each remaining hazard,
- Associate an **ASIL** to each safety goal,
- State the functionality to achieve safety goal: **functional safety requirements**,
- State the implementation of the functionality in HW or SW: **technical safety requirements**,
- State the specific safety requirements which will be implemented as part of HW and SW design: **HW and SW safety requirements**.

This process uses the specific concepts indicated in bold in the previous list. In order to ensure a right application of ISO 26262. These elements shall be clearly traced in the models supporting the process. The Safety Requirement (SR) management process is given in the clause 6 of the 8th part of the standard. It indicates that the use of semi-formal notations (as SysML) for requirements specification is highly recommended to reach ASIL C and D, which justifies once more our approach. We implement these con-

cepts in our SysML profile for ISO 26262 with the new stereotypes provided on Figure 3.

The preceding stereotypes define the artifacts to trace, when addressing functional safety. Design artifacts are defined to differentiate the entities to produce (e.g. hazard, requirements), their attributes (e.g. ASIL) and the relationships that can be declared among them (e.g. achievement, allocation). The utilization of the relationships is translated in the entities attributes by "inherited attributes" (preceded by the "/" symbol). This stereotypes model is built on the ISO 26262 recommendations fixing the expected relation and features offered by the notions relative to functional safety.

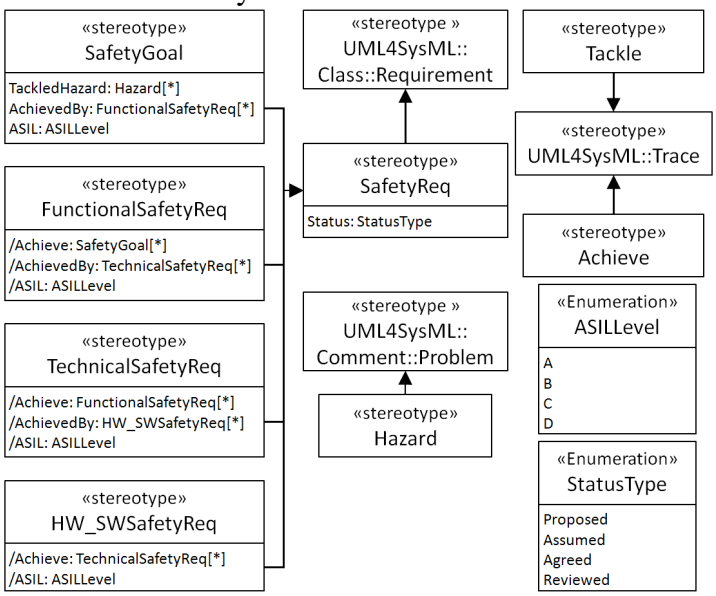

Figure 3. Stereotypes for functional safety management.

As we mentioned in introduction of this section, the management of SRs is the keystone of ISO 26262. The remaining of the standard organizes all the design process realization around this notion. To lead the successive steps of the product realization, other SysML models and artifacts will be useful and will interact with the SR model.

5 SUPPORTING ISO 26262 FROM CONCEPT PHASE TO HARDAWARE/SOFTWARE DESIGN

Beyond a profile realization, we have studied the way SysML modeling constructs and abstraction could support the various workflows demanded by ISO 26262 safety lifecycle. Throughout this process, several actions and analyses are scheduled. Some are parts of the system definition and design, others are the verification and validation of the proposed system and lasts are reporting and management activities. Following the automotive safety lifecycle steps of Figure 1, we will present how to use SysML diagrams to guide developers work and decisions, from concept phase to hardware and software design. Depending on the phase, some diagrams have to be created, modified, analyzed or participate to the documentation process. This first analysis sketches a methodology for the application of ISO 26262 in a SysML context, which reuses parts of the MeDISIS methodology presented in (David et al. 2009a, 2009b) and new elements brought by (Cressent et al. 2010).

5.1 *Concept phase*

The objective of this phase is to initiate the system development, by defining the item and its requirements, analyzing its hazards and formulating its functional safety concepts made to ensure the functional safety. An impact analysis may be incorporated if the system development is an existing system modification, which is not our focus.

5.1.1 *Item definition*

The first task is the item definition beginning by an analysis of the goals and environmental conditions, followed by a declaration of the functional, non-functional and legal requirements. The item boundaries and interfaces must be determined through elements identification, functionality allocations, interaction with environment recognition and inherited requirements declaration. Finally, the operating scenarios must be declared.

This first task is crucial for the remaining of the system development. It calls the use of a lot of System Engineering activities participating to the classical functional analysis. It involves the construction of many types of SysML diagrams covering the modeling axis proposed by the language specification: Requirements modeling, Architecture modeling and Behavior modeling. The main advantage of the use of these diagrams for the realization of this step is the possibility to explicitly link all these system views, thanks to the various allocation constructs detailed in (Hause, 2006). The realization of the Item definition phase, using SysML diagrams, leads to the constitution of:

- **Requirements diagrams** for the identification of functional, non-functional and legal requirements and environmental constraints,
- **Specific Requirement Diagrams** expressing already known SRs (imported from similar previous projects),
- **Internal Block Diagrams** identifying item decomposition, boundaries and interfaces,
- **Activity Diagrams** showing the diverse needed functions,
- **Use Cases and Sequence Diagrams**, defining operating scenarios,
- **Allocations among diagrams artifacts** pointing out functions allocation to HW, requirements allocation to elements, interfaces and functions.

These diagrams are composed of simple SysML objects and of others utilizing our profile stereotypes, in order to identify the elements dedicated to ISO 26262 follow up. Their construction can be done following the directives for SysML models construction given in (Friedenthal et al. 2008) or (INCOSE, 2007). We can note that during this phase, the use of SysML concepts facilitates former projects reuse (e.g. for requirements import) as well as the realization of a consistent definition of the system and its missions. The work output, demanded for this step in the standard, is the item definition. When using SysML, the furnished document will be the SysML model showing all the aspects awaited for item definition.

5.1.2 *Hazard analysis and risk assessment*

This phase clearly initiates the safety survey of the system. Its objective is to identify from the analysis of the functional behavior and the preliminary architecture of the system, the various hazards of the item. To perform this search, ISO 26262 recommended techniques such as brainstorming, field studies or FMEA. This last option is the one we propose to use, since its realization from SysML models can be optimized. Creation of FMEA from a functional analysis performed with SysML is described in (David et al., 2009b), the proposed concepts are directly applicable in an ISO 26262 context. In fact, this phase is the first step of the MeDISIS methodology. However, some minor adaptation of the FMEA report to the ISO 26262 have to be done, by proposing an evaluation of risks based on the criteria given in the standard: severity, probability of exposure and controllability. Then, we propose to translate the results of the FMEA on a Requirement Diagram dedicated to the SRs definition, employing the stereotypes of Figure 3. This construction gives evidences that hazards are tackled by safety goals and that an ASIL is given for each safety goals, by simple relationships analysis between modeling objects. The work products of this phase are the resulting FMEA and the SR Requirement Diagram defining the safety goals. The last awaited work product is the review of the previous analysis and definition. It is clear that the review is aided by the expressivity of the models furnished which shows explicit links between the diverse entities.

5.1.3 *Functional safety concept*

This activity is performed in two main phases that lead to the definition of the functional safety concepts. First, functional SRs are formulated for each safety goals. Then, functional SRs are allocated to elements of the item. The method provided in the ISO 26262 document to perform the task is to choose among known safety mechanisms the one adapted to the safety goal. The declaration of the functional SRs is done on the requirement diagram of SR previously constructed, by defining new requirements employing the functional SR stereotype.

The allocation of the functional SRs to the elements of the item can then simply be made using the *satisfy* relationship, provided by the SysML specification between objects derived from the *requirement* and the *block* or *part* stereotypes. Once again the SysML modeling artifacts allow to obtain a consistent model of the system, unifying in one modeling set the whole treated issue.The work products of this phase are the new version of the SR Requirement Diagram and the set of allocations between elements and functional SR. It is important to note that most of SysML tools provide a tabular representation of allocations existing in the model, under a shape given in the SysML specification. These tables are a great support for the review process concluding the expected work products of the phase.

5.2 *Product development: system level*

The process continues with the activities defining the concrete realization of the functional safety concepts. The product development at the system level refines the technical concepts to be set up to reach functional safety. The architectural diagrams will be extended and detailed while following the realization of the SR declared in the previous phases of the lifecycle. The phase begins by the refinement of project, safety, validation and various assessment plans, which we consider are managed with specialized tools as MS project. These sub-activities will not be detailed here.

5.2.1 *Specification of the technical safety requirements*

The objective of this phase is to build the technical SRs. They are refining the functional SRs towards a description integrating the definition of the functional architecture realizing the safety function. Therefore, this phase is performed by detailing the Internal Block Diagram and precisely the design elements allocated to specific functional SRs. The process will lead to writing the technical SRs added to the diagram of SR, on which their *Achieve* relationship with functional SRs will be modeled.

The second phase is the translation of the technical SRs on the elements properties and interfaces. The properties of the model elements representing the architecture component must be defined. The *part properties* on the system Internal Block Diagrams must be fixed to respect the technical SR. Moreover, Parametric Diagrams shall be used to declare the constraint applied to the elements attributes.

Then, the avoidance of latent faults must be assessed. The system behavior must thus be analyzed in order to spot multiple point failures and their cov-

erage by the detection mechanisms. To perform this task, formal descriptions including the dysfunctional behavior must be utilized in order to extract models of failure propagation as Fault trees. This approach join the second phase of the MeDISIS methodology, which permit to obtain formal description of the system using the AltaRica Data Flow language, that allow the extraction of the Fault Trees characterizing the failure of the whole system. This mechanism is described in more details in (David et al. 2009a).

The fourth sub-phase of this activity is to check if technical SRs are compliant with functional SRs. Preliminary consistency checks can be performed by the modeling tool, for example verifying if the allocation of technical SR to architecture is compliant with the allocation of functional SR to higher level architecture elements, or controlling that all functional SR is at least allocated to one technical SR. We notice that using semi formal descriptions allow various benefic check up possibilities enhancing design quality. The review can be completed by a walkthrough of the system model, facilitated by the SysML models whose organization and internal links accompany the analyst during the inspection.

The expected work products of this activity are mainly the technical SR specification and their review. The specification is represented here by several kinds of diagrams. First, the SR Requirement Diagram is refined and centralizes the declaration of technical SRs. This model is used for the review process to check the compliance and allocation of technical SRs with the existing functional SRs. Secondly, Internal Block Diagrams and Parametric Diagram are provided to specify the realization of the technical SRs by item elements. The attributes and interfaces of the elements specify the awaited features of components that are allocated to the technical SRs. Parametric Diagrams express the constraints that the attributes must verify. Allocation tables between elements and technical SRs can be used once again to support the review process demanded in the ISO 26262.

5.2.2 *System design*

This step is devoted to the refinement of the system architecture specification. Designers have to propose an accurate architecture that meets the technical SRs. The main design of previous phases is augmented with the elements realizing the technical safety concepts. Therefore, elements of the architecture are modified or just added to the existing structure. This corresponds to the modification of architectural diagrams of the model: Internal Block Diagrams. The declaration of elements attributes creates a specification for HW and SW implementation. These designs must be proposed considering the ASIL and previous well trusted design patterns. The ASIL are easily traceable through the model thanks to SysML allocation mechanisms and the reuse is facilitated if components libraries are created within the company.

The introduced modifications imply the necessity to verify the architecture in terms of safety. Systematic faults and random HW failures must be studied using the techniques given in the standard: FMEA, FTA and Markov modeling. These analysis are well supported by the SysML models as we mentioned before using our previous the works of (David et al., 2009 a,b).

This phase is concluded by the system verification against its full requirements, which can be performed depending on the targeted ASIL by a design inspection, simulation or prototyping. The SysML model supports the design inspection as it exposes the whole system. It can be also used to derive simulation models since several tools exist to derive such models for Matlab Simulink (See Artisan Software solutions) or other tools as XaiTool (Peak et al. 2007). The main work product of this step is a refinement of the whole SysML model, showing a concrete structure of the system and identifying the HW and SW components. It takes the shape of the central model of the system development environment produced since the beginning of the concept phase. The dependability analyses performed from this model are part of the review report and show their traceability to the system elements since they are performed from the SysML model.

5.3 *Continuing ISO 26262 process*

The development is continued by a transition to HW and SW level design. The conception of the elements is described in the part 5 and 6 of the standard. The models that are used during those phases must allow to depict domain specific concepts related to the elements technology. Thus, specific languages and modeling approaches have to be used to develop HW and SW components. SysML is no longer the appropriate modeling language for these steps of the design. Therefore, transition mechanisms have to be designed between the SysML descriptions and the needed Domain Specific Languages (DSLs) to continue HW and SW development. Nevertheless, the SysML descriptions have crucial information for the specific models construction. The structure of the language and its interchange format using XML (eXtensible Markup Language) syntax permit to develop many model translations towards DSLs. Examples are given for real-time dependant component development with a transition to AADL in (Cressent et al. 2010) or for AUTOSAR components realization in (Giese et al. 2009). In general, the overall structure of the system, its behavior and the awaited attributes of components are well and expressively defined in the SysML models. These data are translated in the DSL to finalize the development of elements. The transi-

tion with the central SysML model allows to have a consistent link between models and enhance the rapidity of transition to DSL thanks to the various translation automation possibilities offered by SysML.

6 CONCLUSION

The emergent standard ISO 26262 in automotive industry imposes on car manufacturing stakeholders to set up design frameworks efficiently addressing safety issues. The standard strongly recommends the utilization of semi-formal modeling techniques to progress throughout system development. Therefore, we presented in this article the use of SysML to support the deployment of ISO 26262 concepts. First, we provided an initial metamodel establishing the key notions of the standard, using SysML representations. Then, we discussed how SysML can accompany the designer during the concept phase and the system development at system level, by pointing out the SysML artifacts to be used. We showed that SysML possesses the expected representation capacity for ISO 26262 deployment:

- Comprehensive diagrams,
- Hierarchical representation of the system,
- Requirements capture in various shape (textual + accompanying diagrams),
- Views on portions under study,
- Easy reuse of previous studies,
- Good transitioning to DSLs.

Moreover, we mentioned that our previous works merging SysML models and dependability analysis techniques would be valuable for ISO 26262 realization and that they are directly reusable bricks for the constitution of an instrumented framework supporting this standard. We can conclude this paper by presenting the necessary tasks to deploy the use of SysML for ISO 26262 in a specific company. The main task, above the selection of tools and their connections implementation, is the definition of the design method that uses SysML. It is necessary to define which SysML artifacts must be used for each phase and what do they model. A data model must be realized showing how the concepts of ISO 26262 and System Engineering will be modeled using SysML possibilities. The metamodels given in Figure 1 and 2 will be part of this data model but they must be completed with the whole representation needs (e.g. component and interfaces modeling, fault and failures indication). Then directives on diagrams realization must be provided, ensuring a constant quality and expressivity among design teams. This design method is then combined with the adequate modeling tools and connected to the domain specific development methodologies. Future works shall thus focus on the diverse connections needed with DSLs, as well as on the complete definition of a metamodel covering System Engineering notions and ISO 26262 concepts.

REFERENCES

Bernardi, S. Merseguer, J. & Petriu, D. 2008. Adding Dependability Analysis Capabilities to MARTE Profile. *Proceedings of Models'08, Toulouse, France, 28th September – 3rd October 2008.*

Bondavalli, A. Dal Cin, M. Latella, D. Majzik, I. Pataricza, A & Savoia, G. 2001. Dependability analysis in the early phases of UML-based system design. *International Journal of Computer Systems Science & Engineering, Vol. 16 n°5, pp. 265-275, septembre 2001.*

Cressent R., David, P. Idasiak, V. & Kratz, F. 2010. Increasing Reliability of Embedded Systems in a SysML Centered MBSE Process: Application to the LEA Project. *1st M-BED workshop, during DATE 2010, Dresden, Germany, 12 March 2010.*

David, P. Idasiak, V. & Kratz, F. 2009a. Automating the synthesis of AltaRica Data-Flow models from SysML. *Proceedings of ESREL 2009, Prague, Czech Republic, 7-10 September 2009.*

David, P., Idasiak, V. & Kratz, F. 2009b. Reliability study of complex physical systems using SysML. *Journal of Reliability Engineering and System Safety (2009), doi:10.1016/j.ress.2009.11.015*

Estefan, J. 2008. Survey of Model-Based Systems Engineering (MBSE) Methodologies, Rev. B. *INCOSE MBSE Initiative, 23 Mai 2008.*

Friedenthal, S. Moore, A. & Steiner, R. 2008. A Practical Guide to SysML : The Systems Modeling Language. *The MK/OMG press, Elsevier.*

Gérard, S. Petriu, D. & Medine, J. 2007. MARTE: a nex standard for modeling and analysis of real-time and embedded systems. *19th Euromicro Conference on Real-Time Systems (ECRTS 07), Pisa, Italy, 3rd July 2007.*

Giese, H. Hildebrandt, S. & Neumann, S. 2009. Towards Integrating SysML and AUTOSAR Modeling via Bidirectional Model Synchronization. *Workshop MBEES, Braunschweig, Germany, 22-24 April 2009.*

Hause, M. 2006. The SysML modeling language. *15th European systems engineering conference, September 2006.*

IEC 61508. 1998-2005. International Electrotechnical Commission. Functional Safety of Electrical /Electronic /Programmable Electronic Safety-Related Systems. Parts 1 to 7.

International Council on Systems Engineering. 2007. Systems Engineering Handbook. Version 3.1, 2007.

ISO 26262. International Organization for Standardization. Road Vehicles functional Safety. Standard under development (2009).

Kath, O. Schreiner, R. & Favro, J. 2009. Safety, Security, and Software Reuse: A Model-Based Approach. *Proceedings of the 4th International Workshop in Software Reuse and Safety, RESAFE 2009, Washington, D.C., United States, 27th September 2009.*

OMG 2008. OMG Systems Modeling Language (OMG SysML) V1.1. *1st November 2008.*

Peak, R.S. Burkhart, R.M. Friedenthal, S.A. Wilson, M.W. Bajaj, M. & Kim, I. 2007. Simulation-based design using SysML Part 1: a parametric primer. *INCOSE intern. Symp., San Diego, 4 may 2007.*

Pop, A. Akhvlediani, D. & Fritzson, P. 2007. Towards Unified Systems Modeling with the Modelica ML UML Profile. *International Workshop on Equation-Based Object-Oriented Languages and Tools, Linköping University Electronic Press, Berlin 2007.*

Zarras, A. & Issarny, V. 2001. A UML-based framework for assessing the reliability of software systems. *Proc. Intern. Symp. ICSE workshop on describing software architecture with UML, Toronto, Canada, p 36-46, 2001.*