MODEL-BASED AND SCALABLE FUNCTIONAL SAFETY ENGINEERING METHODOLOGY FOR ON- AND OFF-HIGHWAY VEHICLES

Szymanski Dariusz1; Dexters Bert; Descas Yoann; Van Vlimmeren Marc
1Flanders’ Drive, Belgium

KEYWORDS – intelligent development methods, functional safety, ISO 26262, model-based systems engineering, SysML.

ABSTRACT

This paper presents opportunities for the intensive use of behavioral and structural models during safety-related systems engineering. The key outcome of the presented research activities is the detailed mapping of the SysML model diagrams and physical behavior models for specific steps of the safety engineering lifecycle. Innovative approaches for performing quantitative safety analyses based on the input of SysML models are also discussed. Finally, the positive effect of the new methodology on the efforts required for safety-related engineering processes is elaborated.

The presented study is based on an engineering methodology developed together with leading Flemish industrial partners for designing a safety-related E/E system according to international standards. This methodology can be applied not only to passenger cars but also in agricultural, (earth moving) machinery and off-highway domains. This so-called Flanders’ ASIL methodology (FLAME) describes the processes, work products, roles and responsibilities and presents links to the requirements of relevant functional safety standards. The methodology is supported by a web-based tool and can interact with other development tools. The synergies and gaps identified across the vehicle and machinery domains are also highlighted.

TECHNICAL PAPER

The fact that the model-based approach is accepted for Systems Engineering purposes is reflected in the fully recognised name of the methodology: Model-Based Systems Engineering (MBSE) and in the number of publications [7], [13]. The popularity of MBSE dates back to ‘90s [2]. Despite the fact that safety engineering is an important part of systems engineering processes and that it also uses modelling, e.g. for failure mode and effects analyses (FMEA [10]) and fault tree analyses (FTA [11]) [5], the MBSE approach consisting of the modelling of both the behaviour and structure is still rather in a try-out phase, looking for the best possible solutions [8], [9], particularly for safety engineering purposes. This situation is also reflected in the ISO 26262 standard, which names the model-based approach but only for the development of in-vehicle software (part 6, Annex B, [1]). The standard also proposes the semi-formal notation for requirements specification purposes, pointing out neither specific methodology nor language (part 8, clause 6.4.1, Table 1, [1]).

The purpose of this study was to discover the opportunities and threats that the use of standardised modelling languages may bring to the safety engineering process, in particular for the development of items that are in fact mechatronic systems. The approach presented in this paper is based on the assumption that the generic technology is used, rather than that the technology is customised to the specific needs of a particular industry. That was the reason for choosing the SysML [12] language, which is in fact a semi-formal notation for creating models.

The steps and stages of the safety engineering processes are described by the ISO 26262 standard but there is no easy answer as to how these steps should be implemented. One interesting question is whether SysML-based modelling can be helpful in doing this.
GENERIC SAFETY METHODOLOGY

FLAME is a unique methodology providing for a full description of the generic safety engineering processes. Thanks to its scaling capabilities it is also possible to generate safety engineering processes dedicated to different application domains and to take into account the required Safety Integrity Level. The methodology itself is described using a dedicated database containing references to standards clauses and template documents. Information about roles, responsibilities, work products and input products is also covered. Finally, FLAME not only entails visual process descriptions of what-to-do during the safety engineering process but also a detailed description of how-to-act in the different steps of safety lifecycles.

Figure 1. Main user interface of FLAME portal.

The extension of this scalable FLAME with the model-based approach is a further improvement of the how-to-act part of the methodology. However, this model-based extension is limited to the core safety engineering processes and to one automotive domain only. This means that the process description of organisational processes, project management processes or, supporting processes are not model-based; the same applies to the core engineering processes for agricultural and off-road domains.
MODEL-BASED APPROACH IN FLAME

For validating the model-based approach, the Concept Phase process [1] was followed for real-life applications. The parts covered are: Item Definition, Hazard Analysis and Risk Assessment (HARA) and Functional Safety Concept [1]. As for the use case, not only the item model was developed, also a subset of methodologies consisting of Word and Excel templates was created, derived from FLAME and from the SysML model of the safety process itself (see Figure 1). In fact, the model concerns a linked item-and-process model (see Figure 2).

Figure 2. Core safety engineering process presented in FLAME portal.

Figure 3. Main flow of the safety engineering process (modelled with the help of Eriksson-Penker extensions [3]).
Both structures of the item and process models strictly reflect the requirements of the ISO 26262 standard, which is shown Figure 5 and Figure 6.

The Item Definition phase is the first phase of any safety engineering process, in fact it plays the role of defining the requirements engineering set-up. This phase is extremely important for the effectiveness of the whole process. Mistakes made during this phase will in the most optimistic scenario cause additional iterations in the safety lifecycle. During this phase, the following diagrams were used: internal block diagrams (IBD), block definition diagrams (BDD), state machine diagrams (STMD), requirements diagrams (RQD) and use case diagram (UCD). The respective diagrams used are mapped in Table 1, including the dedicated Excel tool for conducting the Hazard and Operability Study (HAZOP [4]) for shortfall states identification.
The use case diagrams were used for depicting the item’s functionality. What is important here is that the persons/users are the actors. The use case diagrams were also used for describing the functionality required from other items; here, the item concerned is the only actor. Finally, these diagrams were used for describing the functionality required from the item concerned by other items; here, the other items are the actors. The item’s functionality is also described in great detail with the help of requirements diagrams, which were also used for describing the relevant legal requirements and environmental constraints. These requirements diagrams could be replaced by mere text specifications but then the possibility to create links between requirements and use cases as well as architectural components would go lost. The state machine diagrams were used for describing the item’s Operating Modes [1] and states and for describing the shortfall states [1] (failure states). To ensure that proper distinction is made between modes and states, an appropriate colour scheme was applied.

In the presented methodology, the Preliminary Architecture [1] was included in the Item Definition. Block definition diagrams as well as internal block diagrams were used to describe the Preliminary Architecture. These diagrams not only described the component and communication architecture but also the internal structure of signals and messages. The block definition diagram also served the preliminary allocation of functions. Another important application of internal block diagrams was the description of the effects of the system’s behaviour on other systems. This description was deduced from modelling the flows of physical quantities between systems. Finally, Operational Situations (OSS) [1] were described by way of requirements diagrams. During this stage, the SysML models served as a valuable input but the final structure of the Item Definition document was governed by a carefully designed Word document based on the ISO 26262 standard. The linked model is a very valuable supporting tool to connect the written report to the SysML model as it acts as the proverbial glue between both.

Table 1. Mapping of diagrams and tool deployment for the Item Definition steps.

<table>
<thead>
<tr>
<th>ID1</th>
<th>UCD Actors: users, engineers, authorities</th>
<th>Purpose and functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID 2</td>
<td>STMD</td>
<td>Operating modes and states</td>
</tr>
<tr>
<td>ID 3</td>
<td>RQD</td>
<td>Legal operational and environmental requirements and constraints, assumptions on item’s behaviour</td>
</tr>
<tr>
<td>ID4</td>
<td>FD’s xls tooling</td>
<td>STMD</td>
</tr>
<tr>
<td>ID 5</td>
<td>BDD</td>
<td>Elements of item</td>
</tr>
</tbody>
</table>
HAZARD ANALYSIS AND RISK ASSESSMENT PHASE

During this phase, state machine diagram (depicting the shortfall behaviour) and the requirements diagram (depicting the Operational Situations) served as inputs for the analysis performed with an Excel tool (see Figure 6). The results of this analysis were the Safety Goals (SG) [1], which were introduced into the model using the newly designed Safety Goal stereotype. The stereotype was derived from the requirements class. Even though using the new stereotype helps a bit in describing the model, its application is not critical to the methodology. As you can see, this stage of the safety process was mainly based on a tool external to the SysML model.

Table 2. Mapping of diagrams and tool deployment for the HARA steps.

<table>
<thead>
<tr>
<th>ID4</th>
<th>Reused</th>
<th>Behavioural shortfalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID10</td>
<td>Reused</td>
<td>OSs</td>
</tr>
<tr>
<td>H1</td>
<td>RQD</td>
<td>SGs</td>
</tr>
</tbody>
</table>
FUNCTIONAL CONCEPT PHASE

At this phase of the safety lifecycle [1], the previously established SysML model was worked out in detail, following the architecture-to-design path. The Operating Modes provided by state machine diagrams and the Preliminary Architecture, both from the Item Definition phase, served as inputs for the Fault States Analysis, which was performed with the help of activity diagrams (ACD) taking into account Safety Goals from HARA. Functional Safety Requirements [1] were derived from the Fault State diagram. This was done with the help of a use case diagram with the Safety Goal being the actor. The Functional Safety Requirements are shown in the requirements diagram. The general view of diagrams and tool deployment for this phase is presented in Table 3. In addition to Functional Safety Requirements, extended shortfall state machine diagrams were used to describe the Functional Safety Concept. The latter diagrams add a safety mechanism [1] (i.e. additional functionality) to the original state machine diagrams to achieve the Safety Goal and to maintain the Safe State [1]. At this stage, the safety mechanisms were allocated to the Preliminary Architecture. The block definition diagrams and internal block diagram from the Item Definition stage were reused for that purpose. The additional information was added to the diagrams using an appropriate colour scheme. To conclude this stage, Validation Criteria [1] were derived for every Functional Safety Requirement. The Validation Criteria and their relationship with Functional Safety Requirements were depicted with the help of requirements diagrams.

Table 3. The mapping of diagrams and tool deployment to Functional Safety Concept steps.

<table>
<thead>
<tr>
<th>H1</th>
<th>Reused</th>
<th>SGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSC1</td>
<td>ACD</td>
<td>Faults State Analysis</td>
</tr>
<tr>
<td>FSC2</td>
<td>RQD</td>
<td>Functional Safety RQs (FSRs)</td>
</tr>
<tr>
<td>FSC3</td>
<td>STMD</td>
<td>Transitions to Safety State</td>
</tr>
<tr>
<td>FSC4</td>
<td>UCD</td>
<td>Fault detection</td>
</tr>
<tr>
<td>FSC5</td>
<td>BDD</td>
<td>Allocation of FSRs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

01
ASIL: QM
Safe State: A yaw rate below acceptable limits
Corrective torque applied to the wheels on... [Delta between value and the average]
Emergency operational interval
FSR001.105 [Safe State not achieved]

02
ASIL: QM
Safe State: A yaw rate below acceptable limits
Corrective torque applied to the wheels on... [Delta between value and the average]
Emergency operational interval
FSR001.105 [Safe State not achieved]

03
ASIL: QM
Safe State: A yaw rate below acceptable limits
Corrective torque applied to the wheels on... [Delta between value and the average]
Emergency operational interval
FSR001.105 [Safe State not achieved]

04
ASIL: QM
Safe State: A yaw rate below acceptable limits
Corrective torque applied to the wheels on... [Delta between value and the average]
Emergency operational interval
FSR001.105 [Safe State not achieved]

05
ASIL: QM
Safe State: A yaw rate below acceptable limits
Corrective torque applied to the wheels on... [Delta between value and the average]
Emergency operational interval
FSR001.105 [Safe State not achieved]
CONCLUSION

The presented hybrid approach for introducing the SysML-based MBSE into the safety engineering process proved to be both pragmatic and effective. The approach includes the use of written documents with a carefully designed structure based on ISO 26262 requirements, an automated Excel tool, a SysML linked process-and-item model and the FLAME process. The main advantage of this approach is that the model is self-guiding for future users. Not only the process steps are described, the required diagrams with exemplary content, being the realisations of the respective steps, are linked to the steps as well (see Figure 7).

Other advantages include the potential reduction of the workload following the reuse of parts of the model. However, this workload reduction should not yet be expected in the initial stages of the process. Analyses show a higher consistency and quality and descriptions are easier to understand for the target audience.
Furthermore, the question raised at the beginning of this paper can be answered positively. The SysML-based approach, which is rooted in the UML architecture-to-design software development, does indeed make it easier to follow the architecture-to-design path. Also the concept of using activity diagrams instead of FTA for analysing the consequences of faults proved to be successful.

The initial assumption to use a generic, non-customised SysML profile was not fully observed. One customised stereotype for the introduction of Safety Goals was indeed designed. However, the model will still be operational without introducing this customisation. The lessons learnt from the use case do show a couple of problems that were encountered and are worth mentioning.

One of the problems is that models can still suffer from a lack of consistency because SysML is not a formal notation. The methodology requires the support of web-based FLAME for planning and management activities (see Figure 1). Features such as colouring capabilities, requirements engineering and versioning are very tool-dependent. Despite the use of the SysML model, the methodology is in its current state not entirely model-driven. It strongly depends on reports and their structure. Text reports play a key role in the methodology for gathering and disseminating information. As a result, the version management of the model and its associated documents is a challenging task.

Taking into account FLAME’s scalability, the model-based Item Definition phase of the safety engineering process appears to be applicable to different domains, other phases however are rather domain-dependent.

REFERENCES