A Process and Data Model for Automotive Safety-Critical Systems Design

Hugo Guillermo Chalé Góngora, Ofaina Taofifenua, Thierry Gaudré
RENAULT, Systems & SW Engineering, 1 avenue du Golf, 78288 Guyancourt, France
hugo.chale-gongora@renault.com, ofaina.taofifenua@renault.com,
thierry.gaudre@renault.com

Abstract. This paper presents the formalization of an innovative design process for automotive safety-critical systems. The objects and data used or produced throughout the different steps of the system design process (e.g. requirements, safety goals, functions, components, validation and verification activities…) are formalized in an integrated data model (or meta model). Besides the novel aspects of the design process, which will only be briefly mentioned in this paper, the originality of the approach lies on the combination of two normally independent models: a Systems Engineering data model and an Automotive Safety data model. The latter stems from the future ISO26262 standard relative to the safety of automotive embedded systems. The results presented in this paper are part of the Systems Engineering deployment initiative at Renault and represent the very first efforts aiming at the compliance with the future ISO26262 standard.

Introduction

For the last several years, car manufacturers have had to face an always-increasing list of stakes and challenges. In the strongly competitive worldwide market of today, a car manufacturer has to offer to its customers relevant, innovative, reliable, environment-friendly and safe services. All this must be done at very competitive costs while complying with more and more stringent regulations and tighter deadlines. The extensive use of mechatronic and software technologies is often the only solution to meet these challenges. This trend, however, increases system complexity and consequently increases the risks due to systematic (process) and random (hardware) failures. These risks are of even more serious consequences when we deal with safety-critical systems.

The emergence of the ISO26262 automotive standard, which deals with the functional safety of embedded electric/electronic systems within road vehicles, brings along new requirements and constraints with which the systems as well as the processes allowing their development will have to comply. Although ISO 26262 is concerned with E/E systems, it provides a framework within which safety-related systems based on other technologies can be considered (ISO 2008). This standard is undoubtedly acting as a catalyst for the research of new processes, methods and tools to cope with these new requirements.

This paper presents an answer to this automotive systems “safety dilemma” from the perspective of processes, methods and tools, the purpose of which is to allow us to prepare the arrival of the ISO26262 standard. More precisely, this paper presents the formalization of an innovative design process for automotive safety-critical systems. The paper also explains the method used to formalize the system design process and it explores the expected outcome of this formalization.

In the first part of the paper, we remind the challenges concerning safety in complex safety-critical systems and make a brief introduction to the ISO26262 standard. Then we outline the system

design process conceived in order to meet the previously mentioned challenges and we stress its particularities. Next, the difficulties encountered during the implementation of the process are explained. The formalization of the design process, which results in the definition of a formal data model, originates in part from these difficulties. We then introduce the basic concepts of system and safety ontology and we finally explain the construction and the structure of the data model (its terms, concepts and relationships). The paper concludes with an outlook of the different features made possible by the utilization and the exploitation of the data model.

The Context

Safety in Complex Systems. One of the consequences of the growing complexity of automotive systems is that performing hazard analyses by traditional methods has become very complicated, thus, time-consuming and expensive given the time scales of typical vehicle systems development cycles. One of the reasons for this is that current automotive systems integrate elements that are heterogeneous in nature (software, mechanical, electrical, electronic). New methods for analyzing systems and new design processes supported by adequate tools and methods are thus necessary to face this complexity.

Concerning design processes, the ISO26262 automotive standard presented hereafter defines a system life cycle, the activities that one must perform during the different phases of this life cycle and the support processes that are necessary for these activities. The standard also supplies a specific method for automotive hazards analysis that leads to the identification and evaluation of safety goals called ASIL, for Automotive Safety Integrity Levels. Furthermore, the standard defines specific rules to decompose these safety goals so that they can eventually be allocated on the system architecture and its components. ASIL are used to specify the safety requirements that must be satisfied in order to attain an acceptable residual risk. These requirements deal in particular with validation activities and with measures of conformity that allow guaranteeing the satisfaction of the required safety level.

However, few guidelines or references are given regarding the adaptation and the application of the standard and how safety requirements should be implemented. So little or no advice is given in terms of recommended methods and specific techniques to comply with the standard. The process and data model described in this paper aim at covering these open issues and focus, in particular, on the lifecycle phase dealing with system design as described in the standard.

The ISO26262 Standard. ISO26262 is the adaptation of IEC61508 to comply with needs specific to the application sector of E/E systems within road vehicles. This adaptation applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic, and software elements that provide safety-related functions (ISO 2008). IEC61508 is an international generic standard for the functional safety of programmable electrical, electronic and programmable electronic (E/E/PE) safety-related systems (IEC 2000). Its generic scope has helped IEC61508 become a reference in all the main industrial sectors and has made it the object of numerous adaptations that take into account the specificities of these different sectors (McDermid 2001). We can cite, for example, IEC61511 for industrial processes, IEC61513 for the nuclear power sector, IEC62061 for machines, EN50126, 50128 and 50129 for the railroad sector and, finally, ISO26262 for the automotive sector.

ISO26262, presently in "Draft International Standard" version, should be published as an international standard in July 2011. It remains largely in compliance with IEC61508 in its

substance but diverges in its structure. One important evolution consists of the fact that the main functionalities of the system can be considered *a priori* as safety-related functions; i.e. all the functionalities of the system are analyzed in order to determine whether they are safety-related. Not surprisingly, we find in ISO26262 the definition of safety integrity levels, which determine the activities to be performed according to each integrity level in order to justify an acceptable safety level of the system design. There are numerous adaptations in ISO26262, concerning primarily the system lifecycle, that deal with the specificities of the automotive domain.

ISO26262 defines four ASIL: A, B, C and D. These levels are determined by combining the following criteria: severity, probability of exposure and controllability. Severity is a qualitative measurement of the consequences of a car accident. Classes of severity S0, S1, S2 and S3 correspond respectively to "no injury", "light and moderate injuries", "severe and life threatening injuries (probable survival)" and "life-threatening injuries (uncertain survival), fatal injuries". Probability of exposure is a qualitative measurement of the possibility of the system (and the user) being in a situation where the occurrence of the accident is conceivable. Classes of probability of exposure E0, E1, E2, E3 and E4 correspond respectively to "improbable", "very low probability", "low probability", "medium probability" and "high probability". Whereas those classes have no quantitative values associated, they should be inwardly understood as though separated from one another by one order of magnitude. Finally, controllability is a qualitative measurement of the capability of the user to avoid a dangerous situation. This criterion is specific to the automotive domain where *the user* (the driver) can exercise a certain control on a *permissive system* (the vehicle does not inhibit unforeseen behaviors). Classes of controllability C0, C1, C2 and C3 correspond to "controllable in general", "simply controllable", "normally controllable" and "difficult to control or uncontrollable". These three criteria allow determining in a systematic way the ASIL of a system or of one of its features as shown in table 1 below.

Table 1. Automotive Safety Integrity Levels (ISO 2008)

Severity	Exposure	Controllability		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

If the evaluation leads to an ASIL quotation lower than level A, a QM quotation (for Quality

Management) is assigned to the event and no safety requirements are defined for the system. This is systematically the case for classes S0, E0 or C0, not shown on table 1. A QM quotation means that a Quality Management Process is mandatory and sufficient to meet the safety goal.

Two other topics of the automotive domain are considered in ISO26262: the human factor and the relationship between car manufacturers and their suppliers. As previously mentioned, the user can have unexpected or unwanted behaviors (e.g., crossing downtown at 100Mph). This type of risks, specific to the automotive domain and relatively non-existent in the nuclear power, aerospace or railways sectors where systems and procedures authorize only foreseen behaviors in precise contexts, partially justify the "customer-driven" approach mentioned below. However, the question of how to handle these risks still remains little approached. Concerning the relationship between car manufacturers and suppliers, ISO26262 defines all the activities to be performed by both parties, but it does not define who should execute this or that activity. The share of responsibilities between the car manufacturer and its suppliers is thus left open; ISO26262 imposes only to define this share of responsibilities at the beginning of the project.

One important element to note, which is a big strength of ISO26262 compared to its predecessor, is that every normative part of the standard depends on the safety integrity levels. Hence, the compliance with the standard will be obtained and verified in a systematic way, contrary to IEC61508, which could lead to different interpretations. In other words, ASIL leads to the specification of a necessary set of safety requirements, which, if satisfied, allow asserting the absence of unacceptable risks.

The emergence of the ISO26262 standard in the automotive industry can be perceived either as a source of concern and apprehension or as an opportunity to improve current Systems Engineering processes and working methods. On this last point, ISO26262 is rather exiguous about the methods and the tools that could allow executing the activities it describes. The rest of this paper presents some of the answers to these questions that remain open.

The System Design Process

Background. The systems engineering process applied at Renault is based on the requirements described in ISO/IEC 15288 (ISO/IEC 2002) and its French equivalent NF Z 67-288 (AFNOR 2003). We follow in particular the Technical Processes of the system lifecycle: stakeholder requirements definition, requirements analysis, architectural design, implementation, integration, verification and validation. This process is in great part applied from the vehicle system level and will be briefly presented here in order to have a general view of the process. However, the greater part of this paper will focus on the specificities of our approach that deal with safety aspects.

Customer services and other non-functional vehicle characteristics (e.g. weight, volumes) are first specified at vehicle level. For example, the “vehicle braking service” shall allow the customer to decelerate, stop and maintain the vehicle still when stopped. Each customer service is refined into requirements that are allocated to sub-systems of the vehicle. For instance, the requirements of the “vehicle braking service” are allocated to the “braking system”, to the “lighting system” (which manages the brake lights) and to the “dashboard system” (which manages the braking system information for the driver). A first Preliminary Hazard Analysis (PHA) is performed at vehicle level on each customer service to define vehicle level feared customer events (FCE) and to evaluate their ASIL. The FCE with the higher ASIL are managed at corporate level for all vehicle projects to assure their consistency and completeness. This list can be updated with new FCE

depending on the results of the analyses carried out when introducing new services or new technologies into the vehicle.

Particularities of the Process. One of the distinctive features of the process presented in this paper is the point of view adopted for the implementation of safety-related requirements: Safety must be integrated as early as possible in system design. The approach proposes first to perform Preliminary Hazard Analysis (PHA) on system requirements in order to identify and evaluate hazards for customers and stakeholders. Then other methods, such as Faults Tree Analysis (FTA) or Failure Modes Effects and Criticality Analysis (FMECA), are carried out firstly and foremost on the functional architecture to identify all hazards in the architecture, to evaluate how safety requirements could be best implemented and to assess if their implementation is correct, such as prescribed by the ISO 26262 standard. In other words, hazards analyses are not only performed on the final system or on an advanced definition of it.

This choice is mainly justified by the fact that performing safety analyses on a completely defined physical architecture means that analyses will start too late in the development process and, in any case, after many important design choices and decisions have been taken. This means that modifications to the system design imposed by hazards analyses can only be made at very high or unacceptable costs. In this kind of sequential process, designers often opt for component add-ons to the preliminary system design as a solution (Stringfellow 2008).

Another important feature of our approach, related to the previous one, consists of the fact that, unlike other approaches proposed in the literature, we actually propose an approach that is not only "safety-driven", but also and especially "customer-driven". For a detailed description of the implementation of the "customer-driven" approach, the reader can refer to Chalé Góngora et al. (2009). Let us just outline that safety aspects are linked to the main functionalities of the system, i.e., to the services and behavior expected by the system users, since the first hazards analyses are precisely performed on these same expected services. In our approach, all safety-related aspects are integrated from the beginning of the development process so that system designers can "naturally" take into account safety requirements, just as they would do with the rest of system requirements coming from other stakeholders. This means that system designers will basically apply the same systems engineering process: stakeholder requirements definition, requirements analysis, architectural design, implementation, integration, verification and validation. The only notorious differences lie on the specificities in terms of implementation or solutions and on the nature of the verification & validation activities put into place for this kind of requirements.

Finally, by identifying the safety requirements (by using a particular attribute, for instance) we can highlight all safety-related elements or characteristics of the system design for possible needs of qualification or certification. For instance, when defining the system validation plan, we can pay particular attention on the most critical elements of the system design, in order to be able to concentrate the verification and validation effort on these critical elements or on their integration within the system. The paragraphs below outline the safety-related aspects of the design process.

Safety Process Outline. The system design process presented here can be considered as a first answer to the emerging ISO26262 standard, as previously explained. The first step of the process concerns the elaboration of the system technical requirements (STR), which must meet the stakeholders' requirements and, in particular, the customer services. In its first version, the STR document contains mainly non safety-related requirements but may also include some safety requirements from previous similar system projects if they exist.

The second step of the system process is the execution of preliminary hazard analyses (PHA) on the requirements, which result in the definition of feared system events (FSE). The occurrence of a FSE in a specific operational context may cause at least one of the FCE identified during the customer services PHA at vehicle level, but can also cause new FCEs due to the failure of a system component. The ASIL quotation of each FSE is done in consistency with the FCE ASIL quotation. Each couple FSE and its ASIL defines a Safety Goal, the top-level safety requirements at system level that are then integrated into the STR. In parallel, the same set of system technical requirements is used in the standard systems engineering process to design the system architecture.

As in any model-based approach, it is also necessary to produce a more or less formal description or model of the system under study. The central object used in the next steps of the process, in particular FTA and FMECA, is the functional architecture model. In our approach, these analyses are not carried out in a purely analytical way but by applying a more exhaustive method based on model execution and on systematic fault injection on all inputs and internal flows of the functional architecture. By doing this, we assume that (1) all the functions in the architecture might (and will) malfunction and (2) will produce failing outputs, without further dwelling on the causes of such failures. So, we focus on the effects of these failures on the architecture (i.e. propagation paths) and on the services and behavior expected by the system users (i.e. occurrence of FSE and FCE). The system architecture and validation plan are thus upgraded by taking into consideration two concurring inputs: a first set of safety requirements coming from PHA that are refined and allocated on the architecture and on its components, and a set of safety requirements, mechanisms and measures obtained from FTA and FMECA. Finally, supplementary verification and validation activities are defined and carried out according to ASIL quotations as defined in ISO26262. Table 2 below presents a synthesis of the system design process integrating safety activities.

Table 2. Synthesis of the system design process integrating safety aspects

Step	Activity	Details / expected outcome
S01	System Specification	Elicitation and analysis of stakeholders requirements Outcome: System Technical Requirements document (STR)
S02	System Preliminary Hazard Analysis (PHA)	PHA on system requirements in order to identify FSE and FCE Safety Goals definition: ASIL quotation of FSE and FCE Prescription of first set of safety requirements on the system and on the system design process. Outcome: Safety Goals couples (FCE or FSE, ASIL), First set of safety requirements
S03	System Architecture Design and modeling	Description and modeling of the Functional and Physical System Architectures satisfying system requirements Outcome: System Architectures description, First models of the System Architectures

Step	Activity	Details / expected outcome
S04	Validation plan specification and modeling	Definition of the validation plan from safety- and non safety-related system requirements, in particular from system use-cases. Outcome: Specification of validation actions (tests, checks…)
S05	FSE and faults-injection mechanisms specification and modeling	Specification and modeling of FSE (observers of the unwanted system behaviors) and fault-injection mechanisms on all input and internal flows of the system architecture. Outcome: Specification and models of FSE and potential faults to inject
S06	Validation plan integration	Integration on the system architecture models with models of fault-injection mechanisms, models of the FSE (observers of the unwanted system behaviors), models of system use-cases (stimuli for the system architecture model) Outcome: Integrated test plan models for safety analysis
S07	FSE causes analysis	Generation of fault trees based on the results of test plans execution Analysis of fault trees in order to identify critical elements on the fault-error-failure propagation paths Outcome: Faults trees (top of the trees being the FSE), List of critical system elements
S08	Safety mechanisms specification, safety requirements refinement and allocation on system architecture and components	Based on the results of fault tree analysis Definition of safety mechanisms: safe states, new functions (detection, diagnosis), new components (redundancies, vote mechanisms) Safety requirements refinement and allocation on system components (ASIL allocation on components) and definition of corresponding validation activities Outcome: Specification of safety mechanisms and requirements, Validation plan completed with new tests and checks.
Loop	Modification of system specification, system architecture descriptions and of all models	According to the results of step S8, iterate on steps S01, S02, S03, S04 and S05: modification of all the descriptions and models of the system in compliance with the new requirements Outcome: Updates of (if necessary) - System specification - System architecture design and models - Tests plan, Fault injection and FSE specification and model - Integrated models for safety analysis

Steps S01 to S04 correspond to standard systems engineering process activities, while steps S05 to S08 can be seen as system safety activities that enhance the standard process. The table presents the logical order in which the process activities should be performed, but the process is rather incremental and iterative, where activities can be carried out in parallel. For any given iteration, if safety goals are not fulfilled according to the results of the execution of the test plan (S07 in table 2), the design process is repeated. The last activity on table 2 marks the end of one main iteration. Iterations stop when the system architecture meets the safety goals.

To help to understand the iterative aspect of the process, figure 1 shows a simplified "dynamic" view of the design process, in which iterations and parallelism inside the main iteration are described. The order in which the activities are executed is given by the number associated with the arrows on the diagram.

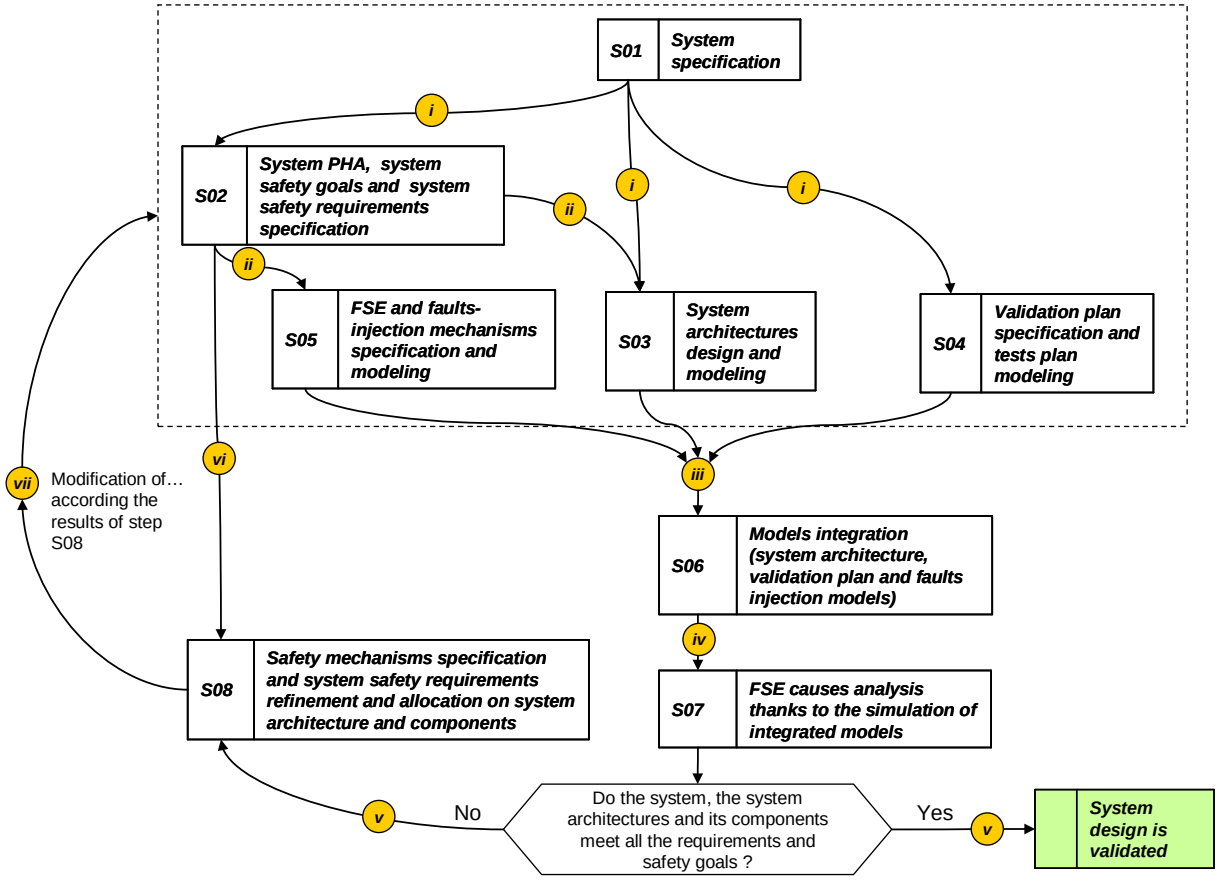

Figure 1. « Dynamic » view of a simplified design loop

Difficulties in Process Implementation. The model-based design process outlined in the previous section calls for different design objects that have to be described as clearly as possible (e.g. requirements, system architectures, safety goals, system use-cases, FCE's/FSE's, fault trees, safety concepts). Our first implementations of the process were document-centric and depended largely on testing and simulation (Chalé Góngora et al. 2009). Although these first attempts yielded quite satisfactory results in building safe system architectures, the creation of the different objects of the process was somewhat troublesome and relatively time-consuming. The reason for

this is that the objects were modeled by means of transformations of *ad-hoc* data and information contained in the different documents that were transmitted from one process step to the other.

The main difficulty in implementing the process consisted thus in the lack of semantic consistency among the different modeled objects. This need for a better formalization is furthermore stressed by the fact that car manufacturers rely heavily on third parties to develop vehicle systems. A better formalization of processes and of the process objects would certainly contribute to avoid confusion and misinterpretations in the development of systems. All this led us to the conclusion that the use of formal and informal (but consistent) models can commit to a common semantic model and to a system and safety ontology whose purpose is to better understand all the aspects of safety-critical system design. These concepts are developed in the rest of this paper.

System and Safety Data Model

Rationale. As stated above, in the competitive market of automotive industry, the product development process tackles conflicting challenges, namely, time to market, increased product quality and safety, reduced costs and integrated innovations. One argument for conforming to systems engineering processes lies in the opportunity to improve the product development process to meet those challenges.

Thomke (2000) demonstrates that “front-loading” effectively transfers problem discovery and problem solving to earlier stages of product development where the cost (both in time and money) of dealing with those problems is lower. Moving problem discovery and problem solving upstream implies that knowledge necessary for the engineers to perform informed decisions has also been moved upstream. Burr (2004) makes clear that the “integration of new methods into existing and new system environments” is heightened in reducing the loss of knowledge at the process interfaces. More specifically, the complex relationships between the individual parts of a system are often implicit and hard to manage.

Adapting the ideas in Chen-Burger (2001) to the level of system engineering in the automotive industry, we can identify their *Domain-Model* as the system under consideration, the other models (i.e. abstract views of the system) are interrelated in that they must describe the *same* system. In systems engineering, the system evolves from a concept to a realization. The Domain-Model comes to existence later in the development process; therefore this concept is moved to the ontological level in a *light-weight ontology* repository of common (i.e. inter-disciplinary, inter-model) formal knowledge. Finally, the Domain-Model can be seen as an instance of the light-weight ontology.

Following these ideas, we chose to develop the data model as an ontology using the entity-relationship paradigm. As explained above, we expect the ontology to improve the development process in a number of ways. First, making implicit knowledge explicit further enables the front loading of activities, while engineers remain capable to make informed decisions, and help to identify areas susceptible to automatic or semi-automatic procedures used in Model-Based Design. Secondly, committing to the ontology makes it possible to work at inter-disciplinary level, the foundation coming from the fact that the system is indeed the common rally point to different professions. For instance, analysis such as multi-model consistency and coherence gives confidence that it is the *same* system that is being built. And last, incorporated safety concepts from ISO26262 help to demonstrate the compliance with the aforementioned standard.

The terms ontology and data model will be used interchangeably in the remainder of the paper.

Ontology vis-à-vis Systems Engineering and Safety Domains. An ontology is a formal, explicit specification of a shared conceptualization (Studer 1998). The definition is explained as follows: *formal* means that the ontology is machine readable; *explicit* means that concepts and how they are constrained is explicitly defined; *shared* indicates that the ontology captures consensual knowledge; *conceptualization* refers to an abstract, simplified model of concepts in the world. Ontology is an active field of research; the interested reader can refer to Gruber (2008) for a more detailed definition and to Changrui (2006) for a discussion on ontology checking. Also of interest are collaborative approaches, which we find corroborates the automotive industry context, found in Sebastian (2008). We chose to develop a data model that uses the entity-relationship paradigm, concepts being entities and relations being relationships. The data model is consistent with systems engineering and makes explicit the principal concepts of a system and the relationships between those concepts. We give particular attention to safety critical systems and so, as a provision to the upcoming ISO26262 International Standard, the data model incorporates safety concepts and relations. The data model has been inspired by multiple sources in literature, the main ones being INCOSE (and its French chapter AFIS), ISO26262 and the systems engineering data model of Renault.

The following is the agreed INCOSE definition of a system: “A system is a construct or collection of different elements that together produce results not obtainable by the elements alone [...] The value added by the system as a whole [...] is primarily created by the relationships among the parts; that is, how they are interconnected” (Rechtin 2000). We retain that the fundamental concepts that we need to capture are parts and their interconnections composing a whole and producing results; i.e. *results* (system requirements) and *structure* (system architecture).

ISO26262 defines safety as “the absence of unreasonable risks” for the users (e.g. the final customer, after-sales personnel) and the environment (e.g. pedestrians, other vehicles). In other words, safety is attained by diminishing the risks until an acceptable level. The standard provides a set of requirements that, if respected, demonstrate that acceptable effort has been undertaken to diminish those risks inherent to the *system* under consideration. Therefore, the safety domain can be manipulated using the same system concepts, i.e. structure and results, but this time from a safety point of view.

The next section presents the design of the data model and defines some of the concepts and their relationships that are sufficient to illustrate the generic properties of the data model and its correctness.

Design of the Data Model. In the previous section we introduced the domains of the concepts that we want to manipulate. We concluded that the concepts were tightly related as the main object of interest is the system itself. The system concepts are therefore presented first. Then they are used as the basis on which the safety concepts are built upon. As far as possible and when useful, the data model will be presented following the logic of the design process on Table 2. Even though most of the concepts in the design process are present in the ontology, its full scope exceeds the communication purposes of this paper. The mechanisms of an ontology go beyond the simple entity relationship paradigm. This is not trivial and needs to be given thorough consideration. In the remaining of this section, elements of the ontology, i.e. concepts and relationships, are italicized. Concepts and relationships are respectively represented with upper- and lower-case characters (e.g. *Concepts* and *relations*).

Figure 2 shows the system ontology where a box represents a concept and an arrow, a relationship. For readability reasons, neither inherited relations nor decomposition relations are shown. A detailed description of figure 2 is given below.

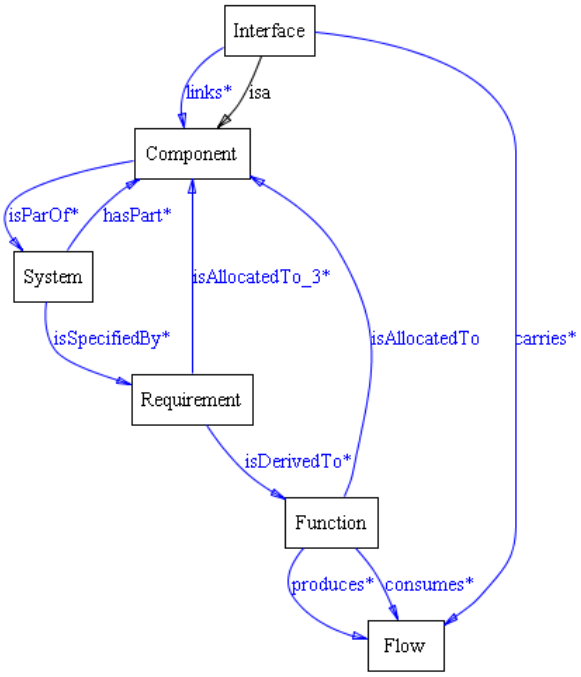

Figure 2. System ontology.

A *System* is a fundamental concept used in different domains. We give the System concept the INCOSE definition stated in the previous section. Coming back to the design process, the first activity is S01-system specification. A system specification is a set of requirements, the fundamental concept being *Requirement*. A requirement formulates what the system is and/or what it does. A system is specified by its requirements, thus the definition of the *isSpecifiedBy* relation. S02 only has safety concepts so we move to S03-System Architecture Design. Two kinds of architecture are defined: functional and physical. *Function* is the key concept of the functional architecture. EIA632 (EIA 1999) definition of a function: “a task, action, or activity performed to achieve a desired outcome” is taken. Functions are derived from requirements; hence the *isDerivedTo* relation that is interpreted as: a requirement can be derived into a function. Functions are structured by the flows they manipulate. A *Flow* is a non-broken circulation (of information, energy or material). A function *consumes* or *produces* flows. The physical architecture of a system is defined by its components and how they are interconnected. A *Component* is part of a system: *isPartOf*. The inverse relation is that a system is composed of components: *hasPart*. A requirement can be allocated to a component: *isAllocatedTo*. A function can be allocated to a component: *isAllocatedTo*. The interconnections of components are captured in the *Interface* concept. An interface is a type of component. The class paradigm is useful here to describe taxonomy structural concept. Using this paradigm a concept in the ontology is read as a class. A class can have subclasses and the *isa* relation means that an element that belongs to a subclass also belongs to the super class. The subclass also inherits the relations of the super class. An interface is

a subclass of component; the *isa* relation is used. An interface *carries* some flows and *links* components. Important knowledge is captured using relations templates <hasSub> and <isSubOf> that states that a concept can be decomposed into concepts of the same type. Those relations are inverse of one another. System, requirement, function and component concepts share this property and we define the relations *hasSubSystem*, *isSubSystemOf*, *hasSubRequirement*, *isSubRequirementOf*, *hasSubFunction*, *isSubFunctionOf*, *hasSubComponent* and *isSubComponentof*. No other fundamental system concept is introduced in the remaining activities of table 2.

Now that we have defined a system ontology, we go over the design process once more to capture safety concepts and relations. Activity S02-System Preliminary Hazard Analysis introduces the key elements from the safety domain. We have *Feared_System_Event* (FSE) and *Feared_Customer_Event* (FCE) that are self explicit. General examples of feared system event and feared customer event are "all brake calipers blocked in open position" and "impossible deceleration by the main braking system vehicle running", respectively. An FSE being the cause of FCE(s) is formulated with the relation *causes*. Those events have been identified from the system requirements; the relation *identifies* formalizes this. Then, an *ASIL* (presented in the first part of this paper) *isAllocatedTo* each FCE and then to each FSE with the FSE ASIL depending on the highest ASIL of the FCE's that causes the FCE. Subclasses of ASIL are *QM, ASILA*, *ASILB*, *ASILC*, *ASILD* (not presented on the figure). A *Safety_Requirement* is a subclass of requirement. It is a requirement with a safety characteristic. Each couple (FSE or FCE, ASIL) is defined as a *Safety_Goal* which is a subclass of safety requirement and defined as a top-level safety requirement. We use the relations *isComposedOf_1* and *isComposedOf_2* to represent that a safety goal is composed of a FSE and an ASIL. Activities S03 to S09 do not introduce new fundamental concepts in the sense that the ontology now possesses all the necessary concepts to discuss both system and safety. Figure 3 is the simplified system and safety ontology.

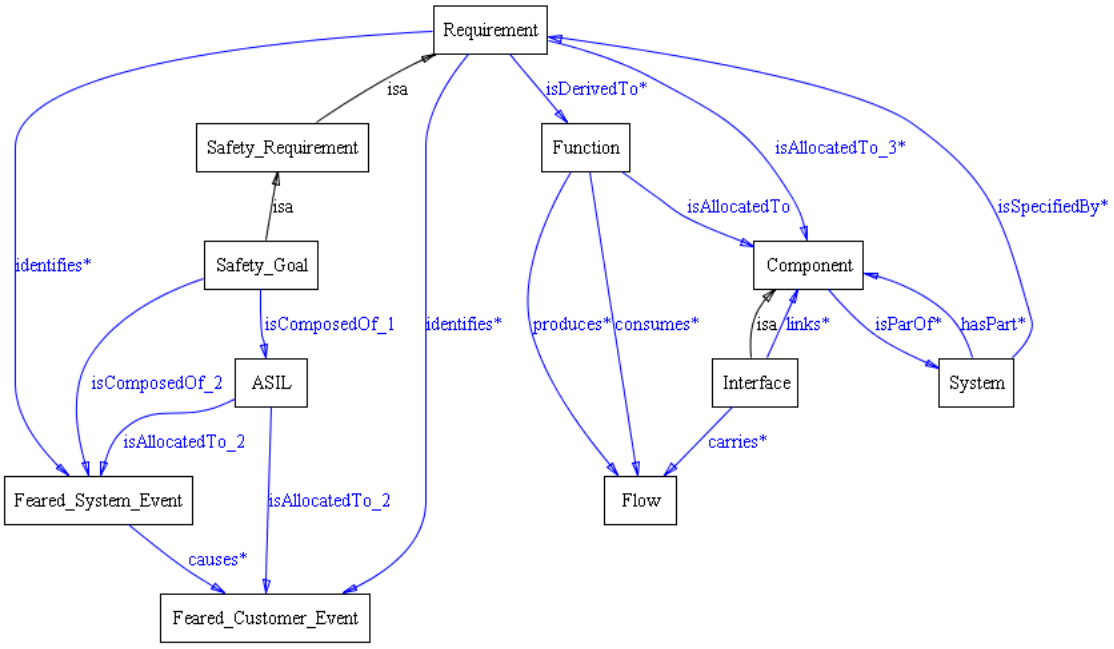

Figure 3. System and safety meta model.

Figure 3 already encompasses a considerable amount of relevant information (some elements have been removed to improve readability), although looking down at figure 3, this is not evident. For instance, the ontology in figure 3 states that the failure of a function is a potential cause of a feared system event, since the function comes from a requirement that is also at the origin of a feared system event and, consequently, of a safety goal. Figure 3 can be used as a meta model to further build the ontology. Activities S03 to S08 of table 2 formulate dependency relationships and new concepts that are not present in figure 3 but introduced in figure 4. For instance, adding concepts *Safety_Related_Function* and *Safety_Related_Component* is done by defining subclasses of function and component. Relations *isDerivedTo_2* and *isAllocatedTo_4* are also made explicit. It results a simplified data model in figure 4.

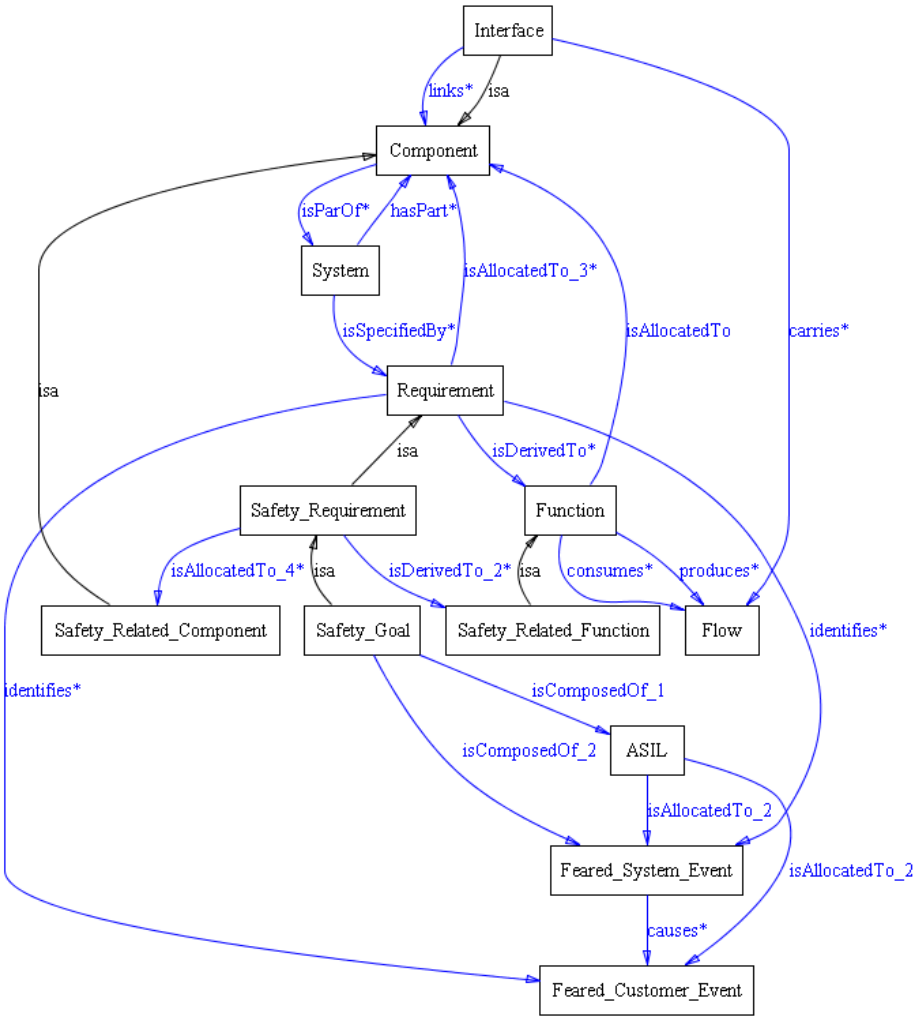

Figure 4. Simplified system and safety data model.

Figure 4 is a lot more understandable than figure 3 albeit a loss of readability. The whole data model is therefore not presented in this paper but the concepts and method on which it has been constructed have been presented. Building up from figure 3 more concepts and relations have been

made explicit making the data model richer and richer while keeping consistency. The ontology language is formal and inconsistencies can be detected automatically. The next step is to use the ontology which is the scope of future work.

Conclusions

The work presented here represents an innovative design process for automotive critical systems and constitutes the first answer to comply with the upcoming ISO26262 standard. The process deals in particular with parts 3 and 4 of ISO26262 respectively entitled "Concept phase" and "Product development: system level". The process provides a rigorous framework in which design choices can be gradually analyzed and validated from the early stages of the development process.

The first implementations of the process revealed difficulties related to a lack of semantic consistency among the modeled objects that were transmitted from one process step to the other. We thus identified the need for a better formalization, which lead us to the development of a data model. One important property of the data model was logically for it to be used as the common semantic model of all the models used internally or by third parties. Ontology, which captures knowledge about a domain at the semantic level, has been used to realize the data model. The modeling activity revealed to be relatively straightforward by using the web ontology language OWL (W3C 2009) and an ontology editor.

The uses of such formal semantic models are numerous and are the heart of our activities for the near future. Among the different possible features that can be supported by the data model, the following three are given as an example. We are first using the ontology as a template to develop specific data models for different automotive systems domains, e.g. braking or steering. Those data models will be central for all the other models realized for a system, as they will enable to check formally the semantic consistency of one model of the system against its central data model, ensuring that the model describes the right system. Second, capabilities are being built-in to query the ontology and to use reasoning. An example of interesting query is to list the ASIL D functions that are not allocated on an ASIL D component. Reasoning is done adding rules to the ontology. For instance, the rule: *if a requirement is derived to a function and the function is allocated to a component then the requirement is allocated to the component* enables to infer all the requirements allocated to all components providing that necessary information has been entered into the ontology. Third, exploiting common formal information in the ontology and reasoning we expect to develop interoperability of the tools used by multiple domains; interoperable behavior such as a change in one tool reflecting in another tool being the objective. Developing an ontology requires a much greater effort than the one required to build a data model, however given the set of possible features that are now within reach, this effort should pay off relatively quickly.

References

Association Française de Normalisation (AFNOR) 2003. NF Z 67-288: Ingénierie des systèmes – Processus de cycle de vie des systèmes. Saint-Denis: AFNOR.

Burr, H., Deubel, T., Vielhaber, M., Haasis, S., and Weber, C. 2003. CAx/engineering data management integration: enabler for methodical benefits in the design process. *Journal of Engineering Design* 16 (4): 385-398

Chalé Góngora, H.G., Taofifenua O., Gaudré T. 2009. Conception de systèmes critiques

automobiles – Etude de mise en œuvre de la norme ISO26262. In Proceedings of 5ème Conférence Annuelle AFIS (Paris, France). Orsay: AFIS.

Changrui, Y., Hongwei, W. and Yan, L. 2006. Extended Ontology Model and Ontology Checking Based on Description Logics. *Fuzzy Systems and Knowledge Discovery*. LNCS, 4223: 607-610. Berlin: Springer

Chen-Burger, Y.H. 2001. Knowledge sharing and inconsistency checking on multiple enterprise models. In 17th International Joint Conference on Artificial Intelligence, Knowledge Management and Organizational Memories Workshop (Seattle, WA)

Electronic Industries Alliance (EIA) 1999. EIA-632: Processes for Engineering a System. Government Electronics And Information Technology Association Engineering Department, EIA STANDARD

Gruber, T. 2008, Ontology. *Entry in the Encyclopedia of Database Systems*, eds. Liu, L. and Özsu, M. T., Springer-Verlag 2009.

International Electrotechnical Commission (IEC) 2000. IEC61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems. Geneva: IEC.

International Organization for Standardization (ISO) 2008. Committee Draft ISO/CD 26262: Road Vehicles – Functional Safety. Geneva: ISO.

International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) 2002. ISO/IEC 15288: Systems Engineering — System Life Cycle Processes. Geneva: ISO.

McDermid, J. A and Pumfrey, D. J. 2001. Software Safety: Why is there no Consensus?? In Proceedings of ISSC 2001 (Huntsville). System Safety Society.

Rechtin, E. 2000. *The Art of Systems Architecting*. New York, CRC Press

Sebastian, A., Noy, N.F., Tudorache, T. and Musen, M.A. 2008. A Generic Ontology For Collaborative Ontology-Development Workflows. *Knowledge Engineering: Practice and Patterns*. LNCS, 5268: 318-328. Berlin: Springer

Stringfellow, M.V., Owens, B.D., Dulac, N., Leveson, N.G. 2008. A safety-driven systems engineering process. In Proceedings of the Eighteenth Annual International Symposium of the International Council on Systems Engineering (Utrecht, Netherlands). Seattle: INCOSE.

Studer, R., Benjamins, V.R. and Fensel, D. 1998. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering* 25 (1-2): 161–197

Thomke, S. and Fujimoto, T. 2000. The Effect of 'Front-Loading' Problem-Solving on Product Development Performance. *Journal of Product Innovation Management* 17 (2): 128-142

World Wide Web Consortium (W3C). 2009. OWL2 Web Ontology Language: Document Overview.

BIOGRAPHY

Hugo Guillermo Chalé Góngora is a specialist in Systems and Software Engineering at Renault. He is in charge of the development and the deployment of model-based methods and tools for the vehicle engineering divisions. During the last years, he has been interested in safety-critical systems design and validation, integrated development environments for software, formal methods

and architecture description languages. In the past, he has worked in the field of industrial systems engineering and, in particular, costs engineering and project management in his native country. He holds a doctorate degree in thermal and energy sciences (Centrale Lyon, France), a post-graduate diploma on internal combustion engines and environment (IFP School, France) and on energy conversion (ENSAM, France) and a mechanical-electrical engineering degree (UNAM, Mexico).

Ofaina Taofifenua is a doctoral candidate at UVSQ (France) for Renault. His research interests include formal methods and their application to the design of mechatronic systems and the security-innocuousness of safety-critical systems. He possesses a Master in Data processing from the University of Bordeaux (France).

Thierry Gaudré is a specialist in Systems and Software Engineering at Renault. He is in charge of the development and deployment of specification methods and requirement management tools for systems and software development. During the last years, he has worked on Requirements Engineering and participated to Systems Engineering training for Renault engineers and technical support for innovative systems projects. In the past, also for Renault, he has worked in the field of Quality assurance and on Dependability of software-intensive systems where he led studies on Verification and Validation techniques by means of statistical tests and formal methods. Thierry Gaudré is a 1992 Engineer graduate from Supélec (France), specialized on Instrumentation and Measurement systems.