System architecture, tools and modelling for safety critical automotive applications – the R&D project SASHA

J. Langheim, B. Guegan[1], L. Maillet-Contoz[1], K. Maaziz[2], G. Zeppa[2], F. Philippot[4], S. Boutin[3], H. Aboutaleb[3], P. David[5]

1: STMicroelectronics
2: Delphi
3: Knowledge Inside
4: ESG France
5: Université de Technologie de Compiègne

Abstract:

Keywords: functional safety, model based design, mechatronics, driver assistance, development tools, dependability, system architecture partitioning, multiple core architecture

1. Introduction

The number of safety critical functions in automotive applications is increasing, especially vehicle dynamic controls, driver assistance functions and the introduction of mechatronics in braking, steering or motor control. As a result there is a need for an improved structured approach in development phase to overcome additional complexity. Indeed, functional safety needs to be included at a very early stage in the development process of systems and their components.

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 crucial task in order to maintain the competitiveness of these organizations on the future automotive market.

SASHA (Safety check of Automotive Software & Harware Architectures) is an opportunity to apply ISO26262 development methodology to an automotive safety critical subsystem and for each partner involved in SASHA, to further improve and validate their own product roadmaps to address ISO26262.

However, the introduction of this new standard requires a change in the development process, in the communication between the partners in the supply chain and the development of new tools helping to minimize the impact of additional requirements on the development process through this new standard.

The SASHA project has been set up around six complementary partners

- **Renault**, the vehicle manufacturer and final customer for a safety critical function
- **Delphi** with its diesel engine department, maker of the function
- **STMicroelectronics**,supplier of automotive components
- **Knowledge Inside**, Tools and Services Provider
- **ESG France**, complex electronic system consultant, experienced in automotive and aeronautic systems
- **UTC Compiegne**, research center skilled in hardware and software co-design

Each partner is bringing its own experience in automotive safety critical system to validate new tools and methodologies inspired by new automotive safety standard. The SASHA use case is based on a diesel engine torque control and the whole project focuses on:

- Common reading and modeling of ISO26262
- Model based system engineering
- Semiconductor transaction level model
- Semiconductor new safety oriented architectures
- Supporting the process with ArKItect innovative tool

This article will first expose the challenges of safety critical automotive applications, before presenting the SASHA use case. Then, the Model Based System Engineering (MBSE) approach, which frames the ISO 26262 application in SASHA, will be introduced. The ArKItect tool supporting this methodology is detailed followed by the introduction of the TLM language used for microcontrollers design and validation within SASHA developments.

2. The safety critical automotive challenge

2.1 safety critical systems in automotive

Systems are safety critical, if their failure may have a consequence on the human life. Depending on severity, exposure and controllability of a failure different levels of automotive safety integrity are defined (ASIL = Automotive Safety Integrity Level). Levels range from A to D, with criticality increasing from A to D. Depending on this level, certain rules of development and documentation have to be followed.

Examples for safety critical systems are obviously braking, steering and motor control, but also functions like lighting and windscreen cleaning.

These “basic” functions are used by driver assistance systems (DAS) like autonomous parking systems, adaptive cruise control or emergency braking.

The development of such systems needs today years of validation. Validation driving of a “simple” adaptive cruise control (ACC) system needs more than 3 man years of driving on the roads. ACC is simple in the way that it does not address extreme situations. Only a certain percentage of the braking force is applied, distances to other cars are comfortable and the driver has a lot of time to react in hazardous situations. However, each time a new application is created, new tests have to be performed and non-regression has to be assured between software and hardware modifications. ACC has started in premium vehicles and it took some time to arrive in lower segments. Today, ACC is a common option in many vehicles.

The development of a system that goes further towards “responsibility” of the technical function needs obviously more validation and re-use become more crucial.

Electric vehicles offer today the possibility to brake electrically up to a certain point, where a mechanical brake needs to add braking force. We all have followed with attention the success but also the difficulties of electric and hybrid cars today already available on the market.

Further significant improvements and the introduction of new topologies will further increase the variety of solutions.

For example the introduction of electric vehicles with wheel motors and integrated stability control offer the possibility to limit mechanical braking to an emergency brake.

In case of accident, the state of art of technology, development and the results of the validation have to be proven meaning a lot of documentation to be presented. Everything has to be checked. In many cases original equipment manufacturers (OEM = car manufacturer) have to assemble at posterior their documentation for legal cases. Following harmonized rules and adapted tools help to find synergies between developments and re-use of already existing elements.

2.2 ISO26262 model in SASHA project

The structure of the standard is already process oriented. It follows the V model activities. So each clause can be easily translated into a task. The standard covers the whole life cycle of the system development. It includes the system view which is then broken down into hardware and software views. The safety aspects during the production, the maintenance and the decommissioning are also taken into account.

In order to prove the compliance of the development process to the standard all work has to be documented. Within the SASHA project the focus was at first on the description of all tasks for traceability.

During the development process proposed in SASHA, all these activities and tasks are then implemented inside the tool ArKItect. The process and the different views of the system are merged inside this tool. Current tools on the market usually separate the modelling of the process and the modelling of the system. ArKItect can merge these both views into a single tool. In fact each process step should correspond to a specific view of the item. Each requested information in the work products are captured from the modelling artefact of the item. The tool should at the end be able to generate automatically all requested documents and then deliver the safety case proving that the system respects the safety goals.

An additional feature of the tool is the tailoring of the development of safety process according to the category of the item (proven in use, modification of an existing item or completely new item).

SASHA will also study the possibility to integrate the safety process of the standard inside an existing development process (SPICE, V model …).

2.3 Semiconductor new horizon

Automotive market demands are driving the automotive semiconductor technologies to continuously innovate.

Next-generation automotive systems will increasingly require multiple power technology mix, more computational performance, better power efficiency and greater memory content to meet market requirements in automotive applications, such as functional safety, tougher emission standards or ADAS (Advanced Driver-Assistance Systems) solutions.

Overall semiconductor technology shrinks offer opportunities for new automotive system architectures compliant with new stringent safety automotive standard such as ISO26262. It goes from an optimized system partitioning between smart power devices and full digital embedded flash microcontrollers to the duplication of a complete subsystem to match toughest safety expectation.

Semiconductor companies have been offering automotive grade components for years. The new global trend is to offer "fit for purpose" ASIL level automotive grade components. This new product range will ease the development and certification process of automotive safety critical system suppliers.

Multiple core architecture devices for automotive embedded systems are becoming a standard, thanks also to advance processes such as 55nm automotive embedded flash process. For complex systems or sub systems, it is now very important to offer to developers virtual models of those complex systems in order to allow fast development and integration between multiple teams.
Virtual models are also a convenient environment for fault injections to monitor the reactions of the systems. One objective of safety critical system is to be able to detect faults and to confine them.

3. SASHA use case

Diesel Engine Management controllers have embedded safety related functions especially the torque structure which from the Driver's accelerator pedal will determine the correct fuel injection to provide engine and vehicle acceleration .
The unwanted vehicle acceleration through engine torque increase event has lead to system and software architectures which now embed software monitoring functions in the ECU.

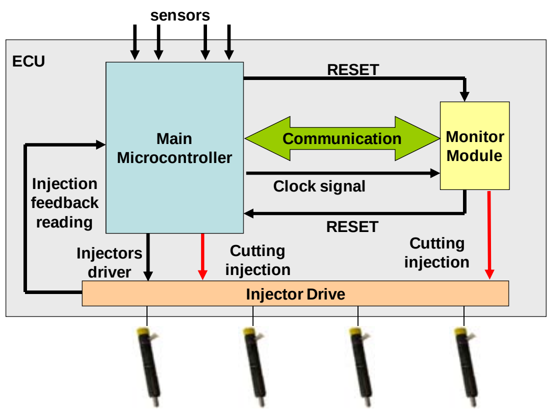

Figure 1 : Global ECU Torque Control

The purpose of the Monitoring Concept as shown in Figure 1 is to monitor the main application to prevent the driver from unwanted acceleration. Moreover its second purpose is to monitor the torque demand computed by main application torque structure.

That monitoring concept includes the actual torque calculation, the permissible torque calculation, the torque comparison and the fault management and related default modes as describe precisely in Figure 2 & 3.

To avoid single CPU errors, monitoring requires multiple levels and especially an independent control of the main controller. (ASICs or secondary micro).

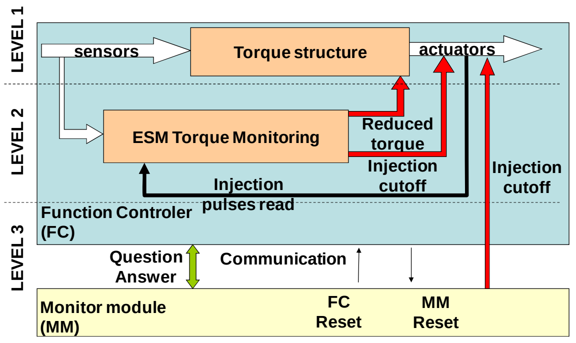

Figure 2 : Torque Monitoring Concept / Functional View

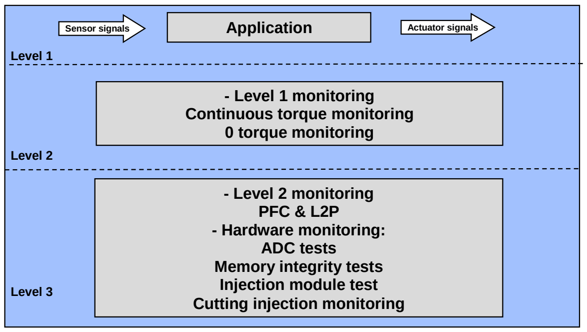

Figure 3 : Torque Monitoring Concept / Abstraction Level

Dual core CPU are an opportunity to reduce cost of multiple components, increase quality and also provide potential new solutions for monitoring concept architecture depending on multicore symmetric or asymmetric design. However, inside the SASHA project, there is no intent to modify the actual monitoring concept solution and architecture, but to use this function as a base for validating the process.

The Torque structure being linked to a set of major functions, only a partial set of those modules will be used for validating the process.

With ISO26262, the various safety related events and safety goals are defined and monitored along all development phases of the software through requirements traceability, from Design to validation. Conformance to the standard will require specific activities, work products and traceability.

Although our example is already an existing support for the ISO26262, other unwanted events, leading to safety goals, will be brought up as requirements, being in the Engine Management Systems or other vehicle ECU's in power train or body controllers.

4. Model Based System Engineering in an ISO 26262 context

4.1 New challenges for safety-critical systems development

In the safety lifecycle proposed by ISO 26262, safety is taken into account since the very beginning of the system development. Therefore, mechanisms have to be set up to support the management of safety objectives throughout the project realization, in order to ensure their monitoring and consistent refinement. One of the emerging challenges in this context, is to be capable of efficiently navigate among the granularity levels needed for system design and validation, and to adapt and clarify the safety requirements for the various detail levels. ISO 26262 provides processes and requirements to frame the design of safety-related systems in road vehicles. It informs about the appropriate method to be used, corresponding to the desired target ASIL deploying the automotive safety lifecycle using semi formal modelling approaches. Moreover, the constraints linked to the development of future safe vehicle functions are well known: higher complexity, shorter time to market, restricted costs, high confidence level expected. Moreover, their increasing complexity stresses new issues as facilitating the communication between domain-specific experts' teams, organizations, stakeholders, certification representatives, enhancing studies experience capitalization or ensuring consistency between the diverse system views exploited during the development cycle. Therefore, the various tasks involved in the safety lifecycle shall be realized with elaborated engineering practices supporting all stakeholders' actions. Consequently, partners of the SASHA project assume that Model Based System Engineering (MBSE) practices are relevant to fulfil project aims and in particular to develop the SASHA framework supporting ISO 26262 deployment. In this section, the MBSE concept used in SASHA will be explained. In particular, the granularity level to reach in the system modelling and the management of safety requirements will be addressed.

4.2 Model Based System Engineering in SASHA project

MBSE is the formalized application of modelling 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 [4]. MBSE enhances classic System Engineering in many domains as communication, preciseness of analysis, results integration or produced knowledge reuse. The expected benefits of MBSE deployment have been summarized in [1]:

- Reduced development risks.
- Enhanced communication.
- Improved quality.
- Increased productivity.
- Enhanced knowledge transfer.

A methodology is needed to implement MBSE. A MBSE methodology can be characterized as the collection of related processes, methods, and tools used to support the discipline of systems engineering in a "model-based" or "model-driven" context [2]. This definition raises the key role of the system model, in direct contact with the methods forming the methodology. In the terminology utilized here the methods are exposing the techniques to use to realize the tasks defined in the System Engineering process. Models and tools are supporting the execution of the method. A valuable survey of MBSE methods can be found in [2]. [1] has drawn up the organization of MBSE models and tools as shown in figure 2. This organization illustrates the central place of a model offering several views on the system: the architecture, the behavior, requirements and constraints explicating physical or trade off study relations verified by the system. The central model is a gateway towards the specific activities as detailed conception concerning domain specific development and analysis tasks including the dependability and safety criterion observation. The model is also used to reflect and respond to requirements emanating from stakeholders, and is a basis to extract viewpoints serving to produce contractual and normative documents. The central model is the crossroads between requirements, the detailed solutions to match them, the analysis to assess their completion and the documentation tracking their expression and fulfillment.

To apply these concepts to the SASHA project, it was necessary to identify the required models for system description and the way to apply ISO 26262 requirements on them. Therefore, a combination of languages, data structures and tools must be provided as SASHA projects outputs.The data structures express the set of artefacts needed for the standard application, each of them represent a concept to manipulate (e.g. a hardware component, a requirement, a test), they are characterized by a set of attributes (e.g. for requirements: associated safety goal, ASIL, associated components ...).

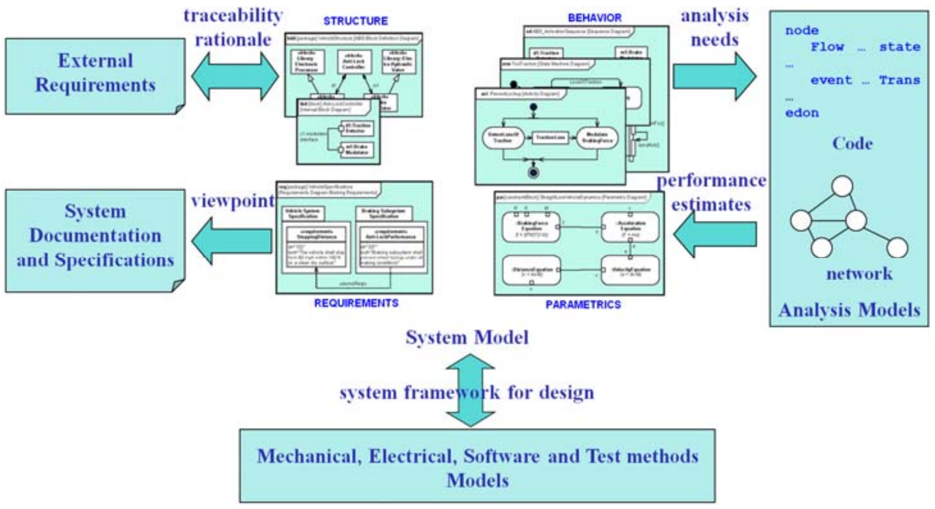

Figure 4: MBSE organization (adapted from [1])

The languages are employed to declare the preceding artefacts and the tools are provided to create the modelling elements, represent, exploit, analyse, maintain and manage them.

Within the SASHA project, the ArKItect tool is supporting the realization of the central model. The tool is based on customizable modelling artefacts. It means that concepts used in the modelling are freely defined to conform to SASHA needs. Therefore, we are constructing a SASHA metamodel comprising the entities we need to execute ISO 26262 activities. Particularly, the central model is to be used to initiate the validation and detailed design steps. Concerning microcontroller architecture definition, thanks to the partners' expertise, we decide to couple the ArKItect model with TLM definitions (presented in section 6). This integration will illustrate the deployment of a MBSE approach, combining the project leading model and domain specific representations for hardware and software design and validation.

5. Developments tools / Arkitect

Traditionally, requirements capture and elicitation follows a top-down approach, while the design and implementation of trustworthy systems follows a bottom-up approach, enabling systems designers and builders to certify desirable safety invariants of the system in a holistic manner.

Given that a norm like 26262 involves many stakeholders, at different levels, to take safety into account since the very beginning of the system design and development, a cooperative tool is necessary to address such system complexity, by following the processes provided by the norm. ArKItect is an innovative tool that fulfils these needs. It is a graphical, user-friendly, and type-based tool that assists system designers and builders in the entire design and development cycle. Its flexibility enables it to implement the norm 26262 for any system, especially engine management system as in SASHA project. In this section, we overview ArKItect, highlight its main features, and illustrate how it could be used in applications.

5.1 ArKItect Features

ArKItect is like SysML, a graphical tool used to support system engineering. It is based on system engineering fundamentals and standards:

- ISO 15288 and EIA632
- System theory: The architecture of complexity [5]

Moreover it has some specific properties that could be noted, since it can:

- Represent graphically a system architecture hierarchically
- Represent graphically requirements (customer needs and system requirements)
- Interface with other tools used to support system engineering (Word, Excel, XML, Doors, Matlab, Simulink…)

It is also a graphics editor for the D2 level Domain Specific Diagram Languages (DSL) which can represent all types of diagrams [6].

D2 – Diagram DSL	At the D2 level, the Diagram DSL represents the language for describing any type of diagrams. It is solution - and platform- independent and contains all criteria understandable by a user who wants to specify diagrams. This level is problem-oriented for specifying diagrams.
D1 – Diagram DSL instance	At the D1 level, a Diagram DSL instance describes a type of diagram. It contains the view model description for producing a type of diagram that is model elements to be displayed with their layout properties. This description respects the language defined by the Diagram DSL. This level contains all data for generating tools producing diagrams.
D0 – Diagram	At the D0 level, we have diagrams expected by end-users in their modeler.

In ArKItect, D2 corresponds to the rule matrix, D1 corresponds to the views, and D0 corresponds to the architecture or model represented.

5.2 SASHA Project
A view is dedicated to the norm 26262 with all its processes.

In Fig. 5 all processes and deliverables are represented. Each deliverable is generated automatically, provided that objects related to the deliverable are represented.
Using ArKItect views-feature, the system is represented at its different levels, depending on the degree of atomicity needed. This is done by modeling the system hierarchically, representing the susbsytems and components. These components could be software and/or hardware components. Thanks to this feature, software and hardware components can be treated independently. For each component/subsystem, the corresponding process in the norm 26262 is applied. Tests at each level are defined, errors injected, and fault trees obtained. Results are represented in the view, and corresponding deliverable is generated. By this way, all the documents needed are generated automatically.

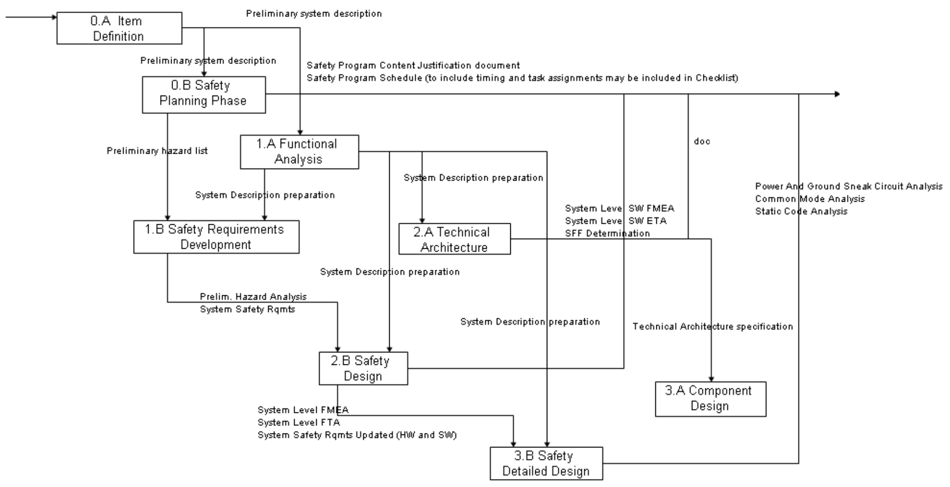
0.A Item Definition
Preliminary system description
Preliminary system description
Safety Program Content Justification document
Safety Program Schedule (to include timing and task assignments may be included in Checklist)
0.B Safety Planning Phase
Preliminary hazard list
1.A Functional Analysis
doc
System Description preparation
System Description preparation
Power And Ground Sneak Circuit Analysis
Common Mode Analysis
Static Code Analysis
System Level SW FMEA
System Level SW ETA
SFF Determination
1.B Safety Requirements Development
2.A Technical Architecture
System Description preparation
Prelim. Hazard Analysis
System Safety Rqmts
System Description preparation
Technical Architecture specification
2.B Safety Design
3.A Component Design
System Level FMEA
System Level FTA
System Safety Rqmts Updated (HW and SW)
3.B Safety Detailed Design

Figure 5: ArKItect processes view

Finally, by using ArKItect collaborative feature, different stakeholders can involve at the same time from different places to work on the project. This enables suppliers from different ranks and constructors to work together efficiently.

6. The Transaction Level Modelling (TLM) approach

6.1 Key features of TLM models

Transaction-level modelling (**TLM**) is a novel technique motivated by the practical need of providing an early virtual prototype [7]. These models are written in SystemC [8], a class library on top of C++, using the TLM standard communication APIs [9], with the following features:

1. *High simulation speed*: 100x to 1000x improvement compared to RTL as a matter of thumb are observed, enabling pre-silicon software development and interactive activities (e.g. software debug)
2. *Model accuracy*: bit accuracy and register accuracy are required, as well as the representation of the system synchronization events [10]. Timing accuracy is not always a strong requirement: functional software development or functional validation may be operated with loosely timed models, whereas time accurate models may be required to optimize software implementation or deal with some real-time aspects of the system.
3. *Early availability*: to enable pre-silicon activities, high level models must be available early. The usual target is 6 months before the availability of RTL models

6.2 Use models
Virtual prototypes based on the TLM approach are used for the following internal activities:

- *Functional verification*. Reliability of the SoCs is obviously a key factor, and is required to gain the time to volume challenge. Therefore, improving the functional verification process, either by reducing the verification time, or extending the verification coverage is a strategic axis. Moreover, getting a unified verification environment at different levels of abstraction (RTL, TLM), improves the process by reusing the same test bench either on the TLM and the RTL models.
- *Embedded software development*. To improve the overall design cycle, early embedded software development enables real hardware/software co-design. The gain is twofold: first, early software development helps to identify specification misunderstanding between software and hardware teams, by sharing the same executable specification. Second, software developers are able to develop and debug their embedded code on a simulation platform that is easier to instrument than the real hardware. This platform can be delivered to all and every developers, and can be used for debug purpose, even when the actual chip is available.
- *System analysis*. In an early phase of the design process, it is also very useful to investigate different functional scenarios.

TLM models might also be delivered to the end customer, as it will enable also its pre-silicon activities, like development of compliance tests or anticipation of system derivatives.

6.3 Contribution of TLM models to safety-critical system development

Developing a TLM model is an important step to get a non ambiguous, executable specification of the System-on-chip. It helps identifying potential inconsistency in the system specification, and it avoids misinterpretation of the specification by the various project stakeholders (architects, software and hardware design teams, etc).
A key question when developing such models is about the consistency between the TLM models, other views of the same specification at other levels of abstraction, and the associated software stack. Using the IP-Xact standard [11], it is now possible to share a representation of an IP and generate header files targeting software activities and TLM model skeleton. The model itself is used as a golden model for functional verification of the RTL IP. As a consequence, non regression test suites can be applied both on the TLM model and RTL IP to ensure functional equivalence. Use of formal methods is also considered to validate the model whatever the input scenario [12,13].

In a safety-critical system development, faithfulness of TLM model is of major importance. Indeed, the execution of the application software on top of the model should match at least with the behaviour observed on the real silicon. A key advantage of the model is its observability and controllability, as it will help software teams to validate the software stack. It is also possible to enable some advanced validation mode, where corner cases can be more easily tested than on the silicon.
To increase the validation of safety-critical systems, TLM models can also be equipped with fault models, to test software robustness. Faults can be injected either on the memory subsystem or on the interconnect model when transactions are going thru. Fault models are built in tight cooperation with technologists, who are collecting experimental results related to physical parameters.
Last but not least, TLM models can also be used to get prepared in a certification process, and anticipate some certification steps with certification authority on the model, before the equipment is effectively available.

7. Results so far

The group has reached a common understanding and interpretation of the ISO26262. The standard has been translated into a spreadsheet and modelized in a flowchart diagram.
The used case has been defined with a focus on the safety critical feature. Safety related events and safety goals have been listed.
Various tools have been exchanged within the group, ArKItect, component models, virtual development environment platforms. Training sessions have been done on ArKItect, electronics component modelization, virtual platform usage.
Focus has been done on methodology re-use.

8. Conclusion

Still at the early stage of the SASHA project, new tools for automotive safety critical system development are showing promising results.
ArKItect is offering a unique environment for specification, model implementation development and validation compliant with latest automotive safety standard.
TLM is a technique to reduce development time and cost for microcontroller implementations. Such development methodology has been already successfully deployed and applied in other industrial sectors such as telecom or consumer. For automotive domain it is a suitable environment to simulate fault injections and monitor the system behaviour under stress. The generally observed anticipation time is about 6 months.
SASHA is a project involving complementary partners leveraging on MBSE know-how in a cooperative and cross experienced environment contributing to a better understanding of the new ISO26262.

9. Acknowledgement

The authors acknowledge the contribution of all involved and sometimes not cited colleagues to this work. Special thanks of course to the French Government with its DGCIS Office, who has accepted to fund this important project.

10. References

[1] [Friedenthal et al. 2008] Friedenthal, S. Moore, A. & Steiner, R. A Practical Guide to SysML : The Systems Modelling Language. *The MK/OMG press, Elsevier.*2008
[2] [Estefan 2008] Estefan, J. Survey of Model-Based Systems Engineering (MBSE) Methodologies, Rev. B. *INCOSE MBSE Initiative, 23 Mai 2008.* 2008
[3] ISO 26262. International Organization for Standardization. Road Vehicles functional Safety. Standard under development.
[4] [INCOSE 2007] International Council on Systems Engineering. *Systems Engineering Vision 2020*. Version 2.03, TP-2004-004-02, Septembre 2007.
[5] H.A.Simon, The Architecture of Complexity, 1962.
[6] B. Langlois, D. Exertier, G. Devda, Toward Families of QVT DSL and Tool, 2006.
[7] F. Ghenassia, Transaction-Level Modeling with SystemC: TLM Concepts and Applications for

Embedded Systems. Springer-Verlag New York, Inc., 2006.

[8] IEEE Standard SystemC Language Reference Manual, December 2005. [Online]. Available: http://standards.ieee.org/getieee/1666/download/ 1666-2005.pdf

[9] OSCI TLM-2.0 Language Reference Manual [online] http://www.systemc.org/downloads/standards

[10] M. Moy, F. Maraninchi, and L. Maillet-Contoz, "LusSy: an open tool for the analysis of systems-on-a-chip at the transaction level," Design Automation for Embedded Systems, 2006, special issue on SystemC-based systems. [Online]. Available: http://www-verimag.imag.fr/~moy/publications/springer.pdf

[11] IP-Xact 1.5 [online] http://www.spiritconsortium.org/tech/docs

[12] M. Moy, "Techniques and tools for the verification of systems-on-a-chip at the transaction level," Ph.D. dissertation, INP Grenoble, December 2005.

[13] Ferro, L.; Pierre, L.; Ledru, Y.; du Bousquet, L.; "Generation of test programs for the assertion-based verification of TLM models", Design and Test Workshop, 2008. IDT 2008

11. Glossary

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
ASIL	Automotive Safety Integrity Level
ESP	Electric Stability Programme
EPS	Electric Power Steering
EV	Electric Vehicle
RTL	Register Transfer Logic
TLM	Transaction Level Model
MBSE	Model Based System Engineering
OEM	Original Equipment manufacturer (Car Manufacturer)
VHDL	**V**ery High Speed IC **H**ardware **D**escription **L**anguage