Safe design of the Autonomous Driving control function

R.Cuer *,**, L. Pietrac*, E. Niel*, S. Diallo**,
N.Minoiu-Enache**, C. Dang-Van-Nhan**

* Institut National des Sciences Appliquées de Lyon,
20 avenue Albert Einstein 69 100 Villeurbanne FRANCE
**RENAULT, 1 avenue du Golf 78 280 Guyancourt FRANCE

Abstract: The autonomous vehicle is meant to drive by itself without any driver intervention. The Autonomous Driving (AD) function is based on the Electric/Electronic architecture of the vehicle constituted of sensors, actuators, ECUs (Electronic Control Units) and communication networks. The focus of this study is on the different states of the AD function, implemented in different ECUs. Traditionally the system design process distinguishes between the systems engineering process and the safety process. In this application, the first process specifies the functional requirements for the AD function while, in the second one, three redundant sub-functions are considered to ensure a continuous service under failure. Each of the two processes might have its own constraints and planning. So, the safety requirements might come often too late to be taken into account in the systems engineering process without major impacts on the design of the vehicle. More than other functions, with respect to its complexity, the AD function imposes to consider the safety requirements at the beginning of the systems engineering process. To achieve this, a state model of the AD function has been built. It allows integrating functional and redundancy aspects, formalizing the approach and formally verifying requirements of interest. The built model will ensure the consistency between the two design processes, functional and safety.

Keywords: Autonomous vehicle, safety analysis, systems engineering, requirements analysis, design systems, discrete-event dynamic systems, redundancy control

1. INTRODUCTION

The autonomous vehicle causes a break in the automotive embedded systems design mainly because it is no more possible to count on the driver reaction in order to keep the vehicle safe in case of failure. This paradigm change will surely deeply impact the design process. But the autonomous vehicle design must also take into account the constraints of the existing and be built on the know-how. The autonomous vehicle design can be carried out following the classical automotive engineering process. Indeed, a system in charge of the new function "Autonomous Driving" (called AD system) is added in a specific type of vehicle. This approach is consistent with the introduction of other ADAS functions (Advanced Driver Assistance Systems) like Adaptive Cruise Controller or Automatic Parking. However, if these systems were already safety critical, the challenge is higher for the AD system because the driver is no more the ultimate safety barrier.

Until now, the automotive systems design mainly relies on know-how, experience, proven-in-use methods and solutions. These systems are designed to meet as many requirements as possible (functional, performance, costs…). However, the safety requirements, resulting from risks analysis, are often actually taken into account in a late design stage. The safety requirements compliance is then justified by tests. This process is appropriate and sufficient in most cases because the safety requirements actual impacts on designed systems are under control. Nevertheless, the safety requirements related to the AD system will considerably influence its design.

To cope with this problem, a formal approach based on the already applied one is proposed. It allows taking into account safety requirements at a very early phase of the design process. For this purpose, a formal behaviour model of the AD system, compliant both with safety requirements and with functional requirements, is built.

The paper is organized as follows. First, we introduce and describe the specific industrial context in which the works take place. The main issue and related sub problems are then explained. In a second section, the approach to address the raised issue is presented. The existing process in the automotive industry is here described. This allows precising the issue and defining work areas. These work axes permit to establish the state of the art. Our proposal is based on the elements found in the literature and the usual process in the automotive industry. In the final section, an application of the approach is illustrated. The obtained outcomes show its relevancy notably because required actions to improve the existing process are determined.

Our main contribution is the proposal of an approach that modifies the conventional design process in automotive industry, and permits to exhibit potential weaknesses in the specifications at a very early design phase. Furthermore, ways to solve the raised problems are defined too.

2. INDUSTRIAL CONTEXT

2.1 A conventional system design process

The design of an automotive embedded system follows the usual V-cycle model. This model describes the activities of the systems engineering process. In the upstream phase of this cycle, the models of the systems (design phase) are developed and studied, while the second phase focuses on the systems themselves (realization, integration). This study concerns more particularly the activity of Verification and Validation but has also strong interactions with the risks analysis, the technical requirements formulation, and the design of the functional and organic architectures. As part of this study, we focus on the verification of the so-called safety requirements, this means requirements provided from risks analysis. This activity is made in the context of the safety process, which is different from the systems engineering one. Indeed, the objective of the systems engineering process is to design as soon as possible and as cost-effective as possible the system in question, while the safety process aims at guaranteeing the system safety under any circumstances. Consequently the planning of the two processes is difficult to synchronize. The AD system might follow this design process but its specificities, described thereafter, force to modify the current approach.

2.2 The Autonomous Driving system

A current vehicle is equipped with a large number of Electric/Electronic (E/E) components, such as: battery, sensors, ECUs, actuators, wire harnesses... The E/E equipment allow implementing some functions in the vehicle, organized in a functional architecture. The autonomous vehicle can be seen like a conventional vehicle, completed by a new function (Autonomous Driving) carried out by a system called "AD system" and that can be active or not. This means that this new type of vehicle can be either driven by the user or autonomously. The AD system is then part of vehicle functional architecture like represented in the Fig. 1:

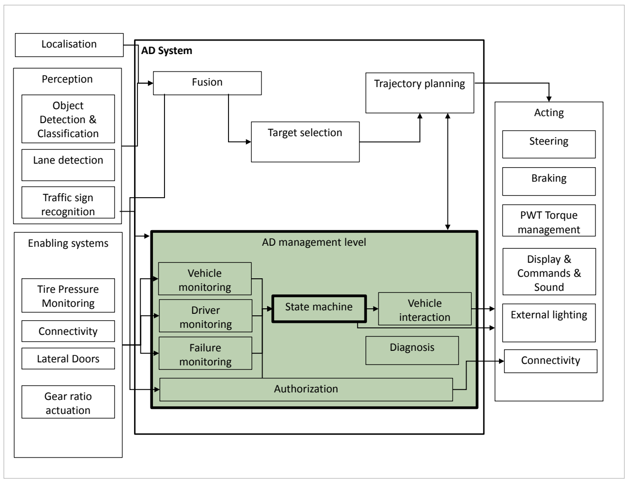

Fig. 1. Simplified functional architecture

The Fig. 1 shows how the AD system is integrated in the functional architecture of the vehicle. The AD system takes the information on the vehicle states and on the driver behaviour as well as the environmental conditions evolutions as input data. From this, a trajectory is continuously determined and sent to the actuators (acting). Furthermore, the AD function can be in different states (active, available…). The functional block called *AD management level* is in charge of this aspect. More specifically, the sub-block *State Machine*, determining in which state the AD function must be, is at the heart of the work done. In particular, a specific state of the AD function, called MRM, standing for Minimal Risk Manoeuver, consists in realizing different pre-determined manoeuvers to keep the vehicle safe in case of failure occurrence. This aspect is specified by safety requirements. It highlights the importance to consider the risks analysis results at the first step of the AD system design.

2.3 Issue

A main issue of this study is precisely to integrate the safety process in the existing systems engineering process. The goal is to take into account as early as possible the safety requirements in the AD system design process. This wide topic raises other difficulties: homogenisation between safety requirements and functional requirements (provided from the systems engineering process); and early integration of the functional redundancy concept. Eventually, the approach proposed must be formalized to ensure rigor and reusability.

3. APPROACH PROPOSED

3.1 Description of the usual approach

On the whole, the usual approach consists in developing a system model compliant with the functional requirements (and justify this compliance by simulation and tests). Then, the compliance of the model obtained with the safety requirements is verified (by simulation and tests too). Moreover, all the requirements are only textually expressed (at the design phase). This approach brings then two main problems: safety requirements that are verified too late can sensibly impact the design (functional and organic architectures) and the safety requirements are not necessarily consistent with the functional ones. To investigate the synchronization between the two processes, a literature review structured in four axes is conducted.

3.2 Related works

The first investigated area is the integration of the safety aspects in the systems engineering process. This topic has been largely studied. (Guillerm et al., 2010), (Lamy et al., 2014) or (Cressent et al., 2011) treat this issue in different fields (aeronautic, manufacturing systems, aerospace). The focus is made on the processes integration, notably by taking into account specific standards requirements. The particularities of the domains studied (constraints of costs, planning, performance…) make the adaptation of these works difficult in the present context. Nevertheless, the same type of study has been already led in the automotive field (Taofifenua, 2012). However, the ontology built, permitting to formally link safety and functional viewpoints, concerns only requirements structure and inter-links. In our case, the requirements content has to be considered too, and not only their structures.

The second issue is relative to the selection of the relevant requirements and their translation in formal properties. Some works address this issue. (Chapurlat, 2013) build a framework allowing to select requirements that are verifiable by model checking and translates them in formal properties. In automotive domain, the problem has also been handled (Mrasek et al., 2016). (Bitsch, 2000) defines a classification of typical safety requirements and their translations in formal properties. All these works can be helpful in the context of this study. Nonetheless, the main followed idea consists in defining boilerplates in order to proceed to the translation. The proposed boilerplates are valid only for the specific application context and the global objectives of the works. Thus, one can inspire of these approaches to translate the requirements but some adaptations to the specific constraints, objective and input data have to be done.

The third topic concerns the homogeneity between the requirements and the behaviour model. In the most of works dealing with formal verification for industrial applications, formal properties are well deduced from requirements. But the system functional behaviour model (compliant with functional requirements) is an input data. More precisely, the way to build functional behaviour model from functional requirements is not detailed (Sharvia and Papadopoulos, 2015), (Kang et al., 2013), (Hajjar, 2013). There is then a lack of approaches to build complete behaviour model (compliant with both safety and functional requirements) in the context of a real case study.

The fourth axe is the verification of formal properties and the use of the obtained results. Many studies address the topic of formal verification in an industrial context and the technic used is generally the model checking (Sharvia and Papadopoulos, 2015), (Kang et al., 2013), (Gan et al., 2014), (Apvrille and Becoulet, 2012). The model checking gives a binary result: either the property is verified or the property is not verified and a counter-example is exhibited. This trace is not necessarily easy to analyse and understand. So the use of the model checking results can be a difficult task. Only few works propose methods to cope with this problem (Mrasek et al., 2016) and the solutions are highly dependent on the application context (field, aim, tool). So, to perform formal verification, one can rely on many cases already studied in the literature. However, there are few elements about the usage of the obtained results.

In conclusion, the development of the proposed approach can draw on consolidated elements of the literature (integration of the safety aspects in the systems engineering process, classification and selection of relevant requirements, formal verification). Nonetheless, the present study also explores less classic problems like building system behaviour model in the context of an industrial case study.

3.3 Improvements proposed

Based on the current approach, three ways of improvement are proposed:
- take the safety requirements earlier into account by building a system behaviour model, formally verifiable even in the design phase
- improving both requirements formulation and system modelling at the beginning of the system design
- make the functional point of view and the safety perspective consistent.

To highlight the interest of the proposed approach, this one is applied in the last section. Since the method has been elaborated to design AD function, we chose this function as example. Nevertheless, the approach is also relevant for other automotive safety critical systems, and particularly for next generation systems.

4. EXAMPLE OF APPLICATION

4.1 Safety point of view

All the safety requirements of interest are contained in a document called *Functional Safety Concept*, resulting from the risks analysis (see 1.1)**.** The first criteria applied to select relevant requirements concerns the allocation. Only the requirements allocated to the *AD management level* are considered (see Fig. 1). It is here important to note that actually three redundant sub-functions (named Main AD, Sub AD and AD-3) ensure the global AD function. The requirements content explains how this redundancy works. The second filter consists in retaining only requirements that are relative to a state change. Finally, 73 safety requirements were sorted out of about 350 initial requirements (20 percent). From the requirements, states and transitions for each sub-function are deduced and the following model is obtained in this way:

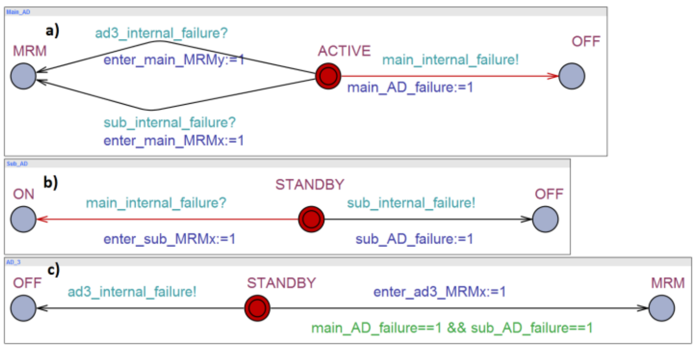

Fig. 2. Safety behaviour model, a) for Main AD, b) for Sub AD, c) for AD-3

The requirement selection is done manually, from the two criteria above-mentioned. For this step, the tool UPPAAL is used because it is a user-friendly and widespread software permitting to proceed to formal verification (Liu and Zhu, 2011), (Lamy et al., 2014), (Apvrille and Becoulet, 2012). These automata describe the following functioning. When there is no failure (nominal functioning), the main AD is in the *ACTIVE* state, the sub AD is in the *STANDBY* state and the AD-3 is also in the *STANDBY* state. If an internal failure of the main AD (resp. sub AD) occurs (*main_internal_failure!* resp. *sub_internal_failure!*), the sub AD (resp. main AD) detects it (*main_internal_failure*? resp. *sub_internal_failure?*) and switches to the *ON* state (resp. *MRM* state) by activating a MRMx (for the MRM x) (*enter_sub_MRMx:=1* resp. *enter_main_MRMx:=1)* while the main AD (resp. sub AD) switches itself *OFF*. The same principle applies to handle common cause failures (affecting both main AD and sub AD) and AD-3 failures.

To finish the build of the safety behaviour model, it is necessary to verify if each selected requirement is well respected. So the formal method of model checking is applied to carry out requirements verification. The treated requirements are for this purpose translated into CTL* (Full Computation Tree Logic) formulae, processed by UPPAAL. The way adopted to translate each selected requirement is the following:

Example of requirement: "In case of a sub AD internal failure, the main AD shall switch itself off".

The following statement corresponds to the inferred formal property:

"*A[](sub_AD_failure==1 imply Main.AD_mgt_level.OFF)*"

All requirements have been translated in this way. Then, the corresponding properties have been put in the verifier tab of UPPAAL in order to be checked. The correctness of the established behaviour model (with regards to the selected safety requirements) has been proven. This permits to obtain the safety behaviour model of the AD system and to conclude the step 1.

4.2 Functional point of view

The same type of approach is adopted by taking as input data the functional requirements resulting from the technical requirements formulation (see 1.1). The relevant requirements are selected in the same way as the first step. 70 functional requirements were selected out of about 175 initial requirements (40 percent). Contrarily to the safety approach, the functional requirements deal with the global AD function. Thus only one entity is considered, called AD_function. It should be noted that, for both safety and functional perspective, additional assumptions about selected requirements had to be formulated to build state models. These hypotheses concern for instance requirements completeness or parameters determination. Finally, the following automaton is obtained:

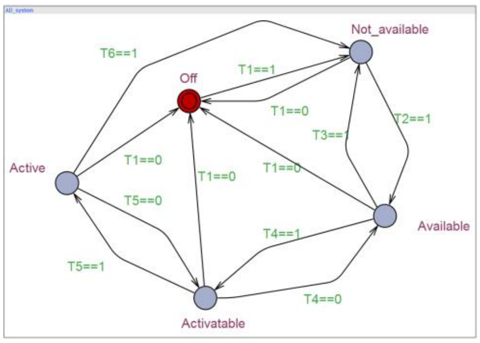

Fig. 3. Functional behaviour model

Each transition Ti is defined from parameters appearing in the selected requirements. These parameters concern mainly environmental conditions, vehicle internal conditions and driver behaviour. According to the values taken by the parameters, the transitions can be fired or not.

The safety behaviour model and the functional behaviour model for the same system are then different in terms of states, transitions and level of abstraction. But none of the two models allows verifying all addressed requirements. To achieve this goal, a unique behaviour model has to be built. This model, composed of four automata (main AD, sub AD, AD-3 and global AD function) will permit us to verify both functional requirements and safety requirements.

The transformation of requirements expressed in natural language into formal state models actually relies on a detailed method, not shown here due to lack of space. An important step of this method is a confrontation with experts' opinion, which secures the transformation. However, the risk of errors remains because of manual nature of the work done. Thus a perspective is to automatize certain stages of the method (for instance steps based on determined and fixed criteria).

4.3 Complete behaviour model

Two problems have to be solved in order to build a complete behaviour model of the AD function. First, the states and transitions of the two behaviour models previously obtained shall be homogenised. Then, a behaviour model of the global AD function integrating the safety aspects has to be proposed. As for precedent steps, this work has been done manually and some assumptions had to be made.

For the first issue, states and transitions from the functional behaviour model are added to the safety behaviour model. The three automata of the Fig. 2 are completed with the states of the functional behaviour model (Fig. 3). To that end, some hypotheses have to be formulated. In fact, each state of the functional behaviour model actually corresponds to a combination of states of the three automata in Fig. 2. For each state of the functional behaviour model, three corresponding states have to be defined. For instance, the state *Available* for the global AD function breaks down into the state *Available* for the Main AD automaton, *Available* for the sub AD automaton and *Available* for the AD-3 automaton. These states have been added to the three automata of the Fig. 2. In this way, three complete automata taking into account both functional requirements and safety requirements are obtained. The three complete automata contain 7 states (Main AD), 6 states (Sub AD) and 3 states (AD-3). They are too complex to be shown in this paper.

The behaviour of the global AD function shall also be determined. In fact, all the starting requirements shall be verified. Yet the selected functional requirements are related to the whole AD function. So these requirements can be verified only on an automaton of the global AD function. It corresponds to the second topic mentioned above. To address this issue, different points of view are crossed.

First, all possible combinations of states (called meta-states) of the three complete automata are listed (126 meta-states). Each meta-state has been deemed to be functionally possible or not, according to some hypotheses. After that, a synthesis automaton is built by making the synchronized product of the three complete automata. This operation is the second formal method used in the context of the proposed approach. The automaton resulting from automata composition is composed of 9 states. Lastly, nominal meta-states are defined: they correspond to the expected meta-states of the global AD function (for instance, meta-state Off = (Main AD=Off, Sub AD=Off, AD-3=Off)).

So, three binary criteria (meta-state functionally possible/impossible, reachable/not reachable by synchronized product, nominal/non nominal) are defined and permit to exhibit eight possible cases. For each case, actions are required that have both consequences on the modelling itself and also on the starting requirements. For example, the meta-states that are functionally possible, reachable by synchronized product but non nominal must be studied. Indeed, these meta-states have to be specified by new requirements, completing the starting ones. This last point is very important because it shows two main interests of our approach: point out the potential weakness of the specifications at a very early phase of the design process and propose ways to solve the underlying problems.

5. CONCLUSIONS

The new safety requirements due to the autonomous vehicle delay the manufacturers from introducing this innovation like other recent driver assistance systems. Thus, the current design process has to be adapted to take into account these changes as soon as possible. In this context, a modification of the current design approach is proposed. It permits in particular to consider the safety requirements and to reinforce the requirements formulation at a very early design phase. The approach application shows the effectiveness of the method. In fact, it allows detecting some problems with requirements (consistency, completeness). Then, actions to solve those problems are determined. Moreover, the approach definition makes it reusable in the future and will facilitate further safety proofs (for all the V-cycle). However, the approach is, for now, not enough integrated within the broader context of MBSE (Model Based Systems Engineering). Besides, most of steps are still done manually but certain may be thereafter automatized. Lastly, the physical implementation is already partially known. It causes particular constraints, some of which might be ever considered.

At least two perspectives are now planned. The first one concerns the development of the AD system complete behaviour model. Indeed, the construction of this model causes modifications on starting requirements. After those changes made and validated, the compliance of the complete behaviour model with the modified requirements shall be formally verified. The second longer term perspective is the allocation of the AD function in an organic architecture. Indeed, this architecture is already largely defined. Yet the allocation brings new issues, which can be partly already addressed.

REFERENCES

Apvrille, L., Becoulet, A., 2012. Prototyping an embedded automotive system from its UML/SysML models. Proc. Embed. Real Time Syst. Softw. 87–124.

Bitsch, F., 2000. Classification of Safety Requirements for Formal Verification of Software Models of Industrial Automation Systems, in: Proceedings of the 13th Conference on Software and Systems Engineering and Their Applications. Citeseer.

Chapurlat, V., 2013. UPSL-SE: A model verification framework for Systems Engineering. Comput. Ind. 64, 581–597.

Cressent, R., Idasiak, V., Kratz, F., David, P., 2011. Mastering safety and reliability in a model based process, in: Proceedings - Annual Reliability and Maintainability Symposium. 1–6.

Gan, X., Dubrovin, J., Heljanko, K., 2014. A symbolic model checking approach to verifying satellite onboard software. Sci. Comput. Program. 82, 44–55.

Guillerm, R., Demmou, H., Sadou, N., 2010. Safety evaluation of complex system, in: IEEE International Systems Conference. 559–562.

Hajjar, S., 2013. Safe design method of embedded systems based on COTS (PhD. dissertation). INSA de Lyon.

Kang, E.-Y., Enoiu, E.P., Marinescu, R., Seceleanu, C., Schobbens, P.-Y., Pettersson, P., 2013. A methodology for formal analysis and verification of EAST-ADL models. Reliab. Eng. Syst. Saf. 120, 127–138.

Lamy, P., Charpentier, P., Petin, J.-F., Evrot, D., 2014. , in: Formal Methods Applied to Industrial Complex Systems: Implementation of the B Method. John Wiley & Sons.

Liu, X., Zhu, Z., 2011. Construct Aspectual Models from Requirement Documents for Model-driven Development of Automotive Software. Electron. Notes Theor. Comput. Sci. 274, 33–50.

Mrasek, R., Mülle, J., Böhm, K., Becker, M., Allmann, C., 2016. Property specification, process verification, and reporting – A case study with vehicle-commissioning processes. Inf. Syst. 56, 326–346.

Sharvia, S., Papadopoulos, Y., 2015. Integrating model checking with HiP-HOPS in model-based safety analysis. Reliab. Eng. Syst. Saf. 135, 64–80.

Taofifenua, O., 2012. Ontology centric design process: Sharing a conceptualization (PhD. dissertation). Conservatoire national des arts et metiers-CNAM.