

## Model-based Development of Safety-critical Functions and ISO 26262 Work Products using modified EAST-ADL

Bülent Sari<sup>\*1</sup>, Hans-Christian Reuss<sup>2</sup>

<sup>1</sup>Electronics Powertrain Technology, ZF Friedrichshafen AG, Friedrichshafen, Germany

<sup>2</sup>Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart (FKFS), Stuttgart, Germany

### ARTICLE INFO

Article history:

Received: 22 March, 2017

Accepted: 06 June, 2017

Online: 23 July, 2017

Keywords:

model-driven approach

safety

ISO 26262

EAST-ADL

### ABSTRACT

Safety is becoming more and more important with the ever increasing level of safety related E/E Systems built into the cars. Increasing functionality of vehicle systems through electrification of power train, in future even more by autonomous driving, leads to complexity in designing system, software and safety architecture. ISO 26262 aims to reduce the complexity and to approve the traceability of the different safety activities. This paper presents an approach about model-based development of system, software and safety architecture using Electronics Architecture and Software Technology - Architecture Description Language (EAST-ADL), being in line with the relevant standard ISO 26262. In particular, we briefly discuss how the main safety related activities, such as hazard analysis and risk assessment, developing functional and technical safety concepts and performing safety analysis can be performed model-based and how the activities can be related with system and software development. The state-of-art is also provided and compared with the proposed approach.

## 1. Introduction

This paper is an extension of work originally presented in Electrical Systems for Aircraft, Railway, Ship propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC 2016) [1]. Nowadays, in a premium vehicle up to 100 processing units (ECUs) are installed, which are capable of computing complex algorithms. The embedded software of a premium automobile contains up to 100 million lines of source code. On the contrary, the new "Boeing Dreamliner 787" needs for all onboard systems around 6.5 million lines of code [2].

This comparison shows how complex the software is already in today's vehicles. Complexity and the scale of the software will continue growing [2]. Major reasons for the large amount of software are the electrification of the automobile and already available advanced driver assistance systems. It is believed that the ratio of the cost of electric and electronic components to the total production cost of a vehicle could rise up to 35% by 2020 and up to 50% by 2030 [3]. With the increase of electrification, the proportion of safety-critical systems is also growing. Malfunctions like "unintended blocking of the drive axle while driving", "power

assisted steering acts in the wrong direction", or "false tripping of the airbag while driving" are a few examples that can lead to life-threatening injuries [4].

Increasing functionality of existing or new vehicle systems both result in increasing proportion of E/E systems and an increase in complexity of the system, software and safety architecture. Engineers from different fields need to be involved. The system and software architectures have to satisfy the different requirements in any phase of development. At this point the question arises, how the industry is responding to the increase in safety and non-safety related functions without sacrificing the quality of the software.

An approach, how the engineers deal with these challenges, uses model-driven system, software and safety development. This research presents an approach for model-based development of safety critical functions and model-based development of ISO work products. EAST-ADL is used as the basis of this approach. However EAST-ADL should be modified in a way to bring the system architecture and safety architecture together, because ISO 26262 requires consistency and traceability by the development of safety critical systems. But the current EAST-ADL specification is not enough to implement these requirements, because currently the

\*Corresponding Author: Bülent Sari, Electronics Powertrain Technology, ZF Friedrichshafen AG, Friedrichshafen, Germany | Email: [buelent.sari@zf.com](mailto:buelent.sari@zf.com)

system architecture and safety architecture are separated in different models and there is not direct relation between system architecture models and safety models. The system architecture is developed within the abstraction levels of EAST-ADL and the safety models are realized within the dependability model. It is a big challenge and also requirement of ISO 26262 (Part 4 – Figure 2, Part 10 - Figure 8 and Figure 9) to show the dependencies between hazards and risks, safety goals, functional and technical safety requirements and safety functions. By safety assessments, the developers should be able to show which safety goals are implemented by which safety functions and it should be proven that the safety functions and safety goals have the same Automotive Safety Integrity Level (ASIL). Therefore it was essential to modify EAST-ADL in a way to bring these models together in order to provide consistency check, traceability of safety functions and also generate the ISO26262 work products such as safety concept automatically. The extensions enable to show the relationship between the safety goals and the safety functions in order to prove easily which function is implemented for which safety goal. So it is easier to prove the completeness of the safety activities.

The second main contribution of the paper is to verify and to validate the technical safety concept and functional safety concept as required from ISO 26262-4:2018, Clause 7.4.8 within simulation of safety requirements in the earlier development phase of project. So it is possible to improve the safety concepts earlier and also to find out the systematic errors in the earlier phase of development. The modeling of a complete system is an extensive project and requires domain specific knowledge. Furthermore, knowledge is needed about various tools, because the architecture is described using several tools. Current approach can be replaced within developed approach based on domain-specific language with ADL [6, 7]. By this it is possible now within the extensions and improvements that all information is created within a system model that describes the complete system, safety and software architecture and offers the consistency, traceability between system and safety architecture and enables to verify the safety concepts in the earlier development phase.

Section 2 shows how EAST-ADL is modified for this purpose.

## 2. Description of the Approach

The developed approach in Figure 1 shows how the system, software, and safety development are combined and thus the complexity of the system is reduced by a coherent architecture.

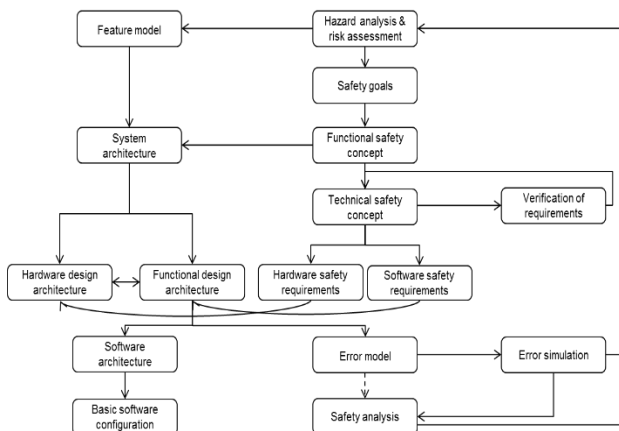


Figure 1 Model driven approach for system, safety and software development

The developed method consists of five main parts, which are shown in Figure 2:

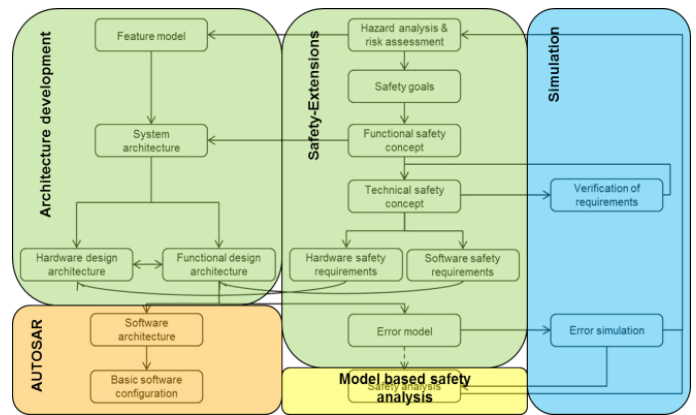


Figure 2 Main parts of the approach

First part is the architecture development [7]. This part is about the creation of feature model, system architecture, functional design architecture (FDA) and hardware design architecture (HDA). It is also possible to allocate the functions from FDA model to the corresponding hardware elements of HDA model. This part is extended within additional safety attributes in order to combine the system architecture and safety architecture and thus it is possible to realize the relationship between system architecture and safety architecture in order to prove traceability and consistency. The further details will be explained by chapter 2.1.

Second part is safety extensions [8, 9]. This part deals with the model-based creation of ISO 26262 work products as an extension to architecture development. Firstly, hazard analysis and risk assessment is performed. Secondly, the safety goals are derived from hazard analysis and risk assessment. In order to fulfill the safety goals, functional safety concept and technical safety concept are created in the following step. This part is also extended with additional safety attributes to realize a relationship between safety model, system model and requirements model. Safety model enables to realize the safety work products model based such as Hazard Analysis and Risk Assessment (HARA), functional and technical safety concept. The extensions enable now additionally to generate these safety work products automatically from the models using the developed scripts. The further details will be explained by chapter 2.2.

Feature model can contain both non-safety-critical and safety-critical properties of the system. After hazard analysis and risk assessment, the safety-critical aspects of the system will be considered as being part of the feature model.

In the system architecture, the safety goals and corresponding safety functions of the system are taken into account. In the functional and hardware design architecture, the safety-critical functions are detailed.

Third part is AUTOSAR [10]. This is about the software architecture and basic software configuration. The software architecture can be created from FDA model, which contains the necessary software features.

Fourth part is model-based safety analysis. In this step, the fault trees (FTA) of fault models are automatically generated from error model of ADL or simulation model with external tools [11] [12].

The last part is simulation and verification. This part is developed within this research. In this step, error simulation and verification of the requirements are carried out and this becomes possible already in the earlier phases of development. On the one hand, the safety requirements are verified with usage of a simulation environment. On the other hand, it is possible to determine the system impacts of causes with error simulation. Thus, the defined top events of the system from hazard analysis and risk assessment can be approved. The further details will be explained by chapter 2.4.

The developed approach allows for consistency and traceability of individual steps. The method also permits the efficient tracing from the software architecture to the feature model and from the safety analysis to the hazard analysis and risk assessment.

In the following subsections the details of the main parts will be described.

2.1. Architecture development and AUTOSAR

The architecture development can be realized with UML, SysML or EAST-ADL. For our tasks, the architecture description language EAST-ADL (Electronics Architecture and Software Technology - Architecture Description Language) is investigated.

The description language EAST-ADL represents an ADL, which was specified initially in research project EAST-EEA (Electronic Architecture and Software Technology – Embedded Electronic Architecture) and developed further to EAST-ADL2 in other research projects ATESS1 (Advancing Traffic Efficiency and Safety through Software Technology), ATESS2 [7] and MAENAD (Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles) [8], SAFE (Safe Automotive Software Architecture) [9] and Synligare [21]. In addition the language was adapted to AUTOSAR [10], such that AUTOSAR can be used for the detailed description and the implementation of the software architecture. It is used to describe electronic systems in vehicles to facilitate the development of vehicle electronics [5]. Thus EAST-ADL is a domain-specific language.

The purpose of EAST-ADL is to provide the engineers facilitation for the representation and description of electronic systems in vehicles in a standardized form [5]. It can be used during different activities for the modeling of functional requirements, safety work products, as well as for analysis and design purposes [13].

The metamodel of EAST-ADL is organized in four different abstraction levels. Each of them fulfills specific roles. Each level considers a "different phase of vehicle development" [14], represents the complete system and provides different perspectives of the whole EEA [5].

The 4 levels “vehicle level, analysis level, design level and implementation level” are described in detail in the following Fig. 3:

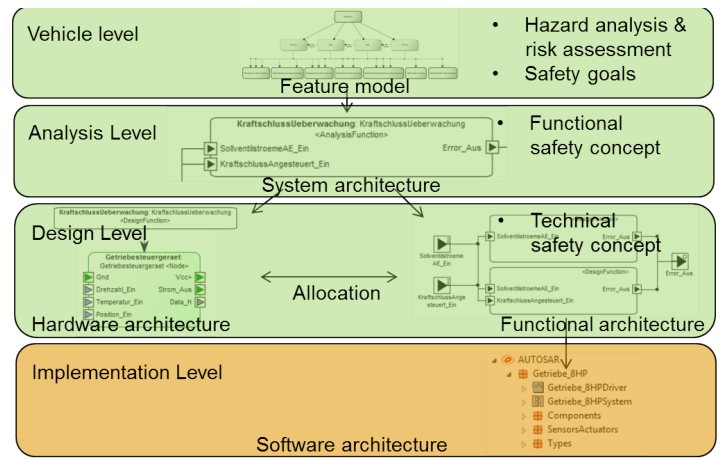


Figure 3 The EAST-ADL levels and safety activities of the approach

As mentioned before, the library elements of these abstraction levels were extended further as follow in Fig. 4. The library elements of EAST-ADL abstraction layer contain the additional information about corresponding safety goal, safety requirements and ASIL classification which enable the relationship between safety model and system model. Additionally the attribute “verified” shows whether the function is already verified in the earlier development phase. This information is very useful for safety case generation and also required from ISO 26262-4:2018, Clause 7.4.8.1.

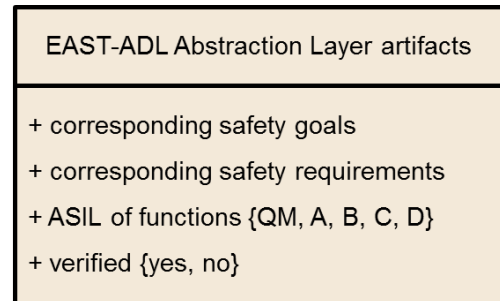


Figure 4 The extensions of EAST-ADL Abstraction layer

At vehicle level, the vehicle or system characteristic is described with a lot of features. In this step, the question 'what' must be realized with the architecture development, but not the question "how". To detect and to eliminate risks as early as possible, it makes sense to perform the hazard analysis and risk assessment in parallel on the same level [15].

The analysis level describes the realization of the features in the functional analysis architecture (FAA) in form of functions. Here we find a top-down decomposition of features on functions and on abstract sensors and actuators [16].

The design level consists of FDA and HDA. The FDA comprises the functions of the FAA, which can be further detailed and decomposed. In addition, an allocation graph is modeled, in which the function blocks of the FDA are allocated to the hardware components of the HDA [18].

The implementation level describes one-to-one the software architecture of the FDA in AUTOSAR [17].

2.2. Safety Extensions

EAST-ADL offers the dependability package with safety extensions, which enable to create the work products of safety development process such as hazard analysis and risk assessment or safety goals, as it is shown in Figure 6.

The dependability model is additionally designed to support the developer in creating safety requirements and performing safety analysis, as well as in the modeling of systematic errors and their propagation, which leads to failures.

As mentioned before, within this research the library elements of dependability model were extended further as follow in Fig 5.

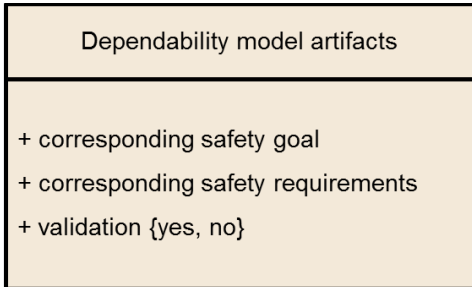


Figure 5 The extensions of EAST-ADL dependability model

The developed extensions within this research enable the generation of safety concepts using the developed automation script. They are created automatically from modified requirements model and dependability model.

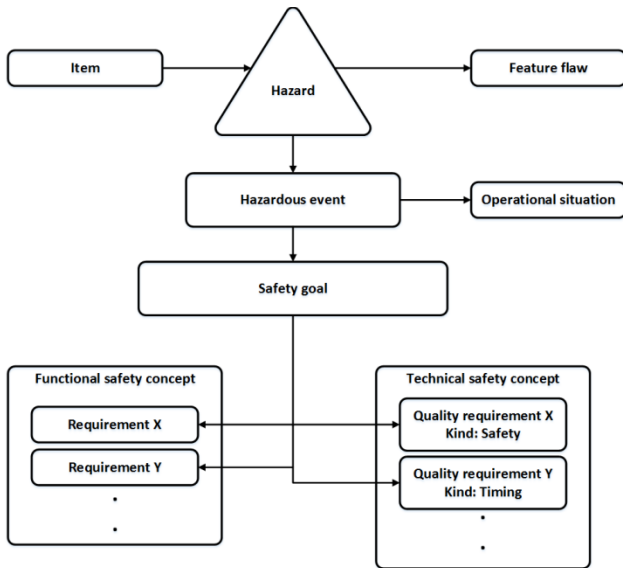


Figure 6 Dependability model

During the development of a system, requirements should be defined, which can be modeled in a separate requirement diagram as shown in Fig 5. Requirements model may include both non-safety-critical and safety-critical requirements. Therefore, it is recommended to begin with the requirements modeling on the system architecture level.

The requirements model in EAST-ADL is based on the SysML (Systems Modeling Language), adapted to the metamodel of EAST-ADL [5]. User-defined features can be added to the

requirements, as it is also possible in DOORS (Dynamic Object Oriented Requirements System). Information such as status, author and responsible person are added, in order to achieve better traceability of the changes. Requirements model can be exported in the format ReqIF (Requirements Interchange Format) to facilitate exchanges with other requirements tools. As mentioned before, the requirements model is extended further as in Fig. 7 and 8 with safety related features such as ASIL classification, corresponding safety goal or decomposition relevance.

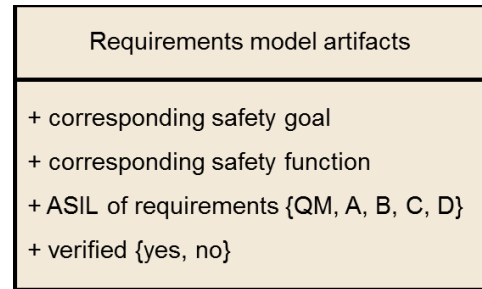


Figure 7 The extensions of EAST-ADL requirements model

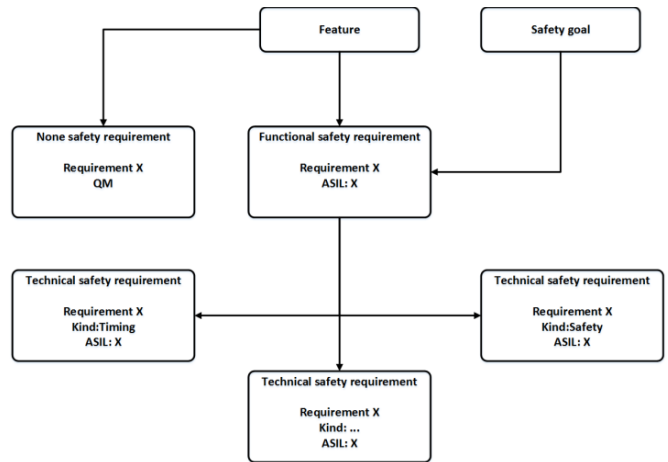


Figure 8 Requirements model

The extensions within this research enable the generation of safety concepts using the developed automating script as follow in Fig. 9.

```

Safety_Concept()
'Functional and Technical Safety Concept'
newline
foreach .Requirement
{
  SA='0';
  $Decomposition='0';
  'Safety Goal : ':Safety_Goal;
  $Safety=:Safety_Goal;
  newline
  $Reqname= :Name;
  'Functional Safety Requirement : ' :Name; ' '
  newline
  SC='0';
  'ASIL of ' :Name;
  ' (Safety-Requirement) : ' :ASIL;
  SA=:ASIL;
  newline
do .QualityRequirement
{
  'Technical Safety Requirement : '
  :Name; ' '
  $FunsafetyReq=:Name;
  'ASIL of ':Name;
  ' (Requirement_Model) : ':ASIL;

```

Figure 9 Safety concept script and generated document

It is possible to generate error models automatically from functional design architecture model (FDA). Error description and error logic to the individual subsystems are modeled afterwards.

Because vehicles in general and especially electrical vehicles consist of complex systems, it is very difficult to determine all measures to prevent a hazard. Of course, Original Equipment Manufacturers (OEMs) and their system suppliers are committed to reduce all risks to an acceptable level. This means that for all hazards and risks relevant safety goals, safety concept, safety requirements and safety measures have to be defined. For this reason, ISO 26262 requires to carry out the safety analysis based on fault models in the automotive industry. For example with the help of the fault tree analysis it is possible to find the error causes for a particular hazard [19].

In EAST-ADL, it is possible to model errors and their propagation in an error model to recreate a failure behavior [20]. On the basis of error models, additional tools such as HiP-HOPS [11], ALTARICA [12] can be used to perform model-based safety analysis.

Figure 6 shows how the error model and error logic can be created.

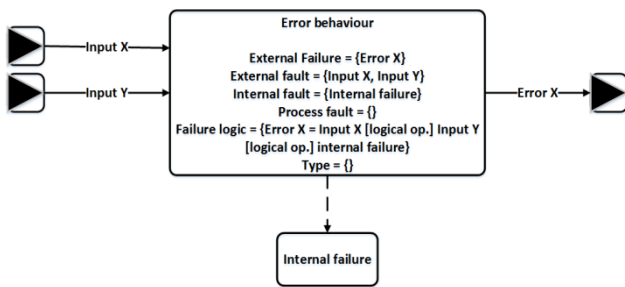


Figure 10 Error Model

### 2.3. Model based safety analysis

It is possible to use commercial tools such as HiP-HOPS [11] to perform the safety analysis. These tools are capable of generating fault tree analysis (FTA) and failure modes and effects analysis (FMEA) automatically from an error model [11].

Safety analysis enables to find out possible error causes which should be detected and avoided with appropriate safety measures. For this purpose, safety measures and requirements are defined in order to introduce countermeasures. Additionally, functional and technical safety requirements can be defined or extended based on the results of the safety analysis.

### 2.4. Simulation

For verification of the requirements and error simulation, a simulation environment such as Simulink (Matlab), Dymola, CarMaker, etc. can be used. For instance, the power train of an electric vehicle can be simulated with the help of a simulation environment and error models can be designed and implemented. The behavior of the vehicle can be simulated with error models in a way where the error causes such as failure sensor values can be used as stimulation signal. The simulation is used for the verification of the requirements, for detection of the errors in the earlier phase of development and for detection of the system impacts of errors. If it is found that the safety requirement cannot

be implemented as intended, then the requirement should be redefined. This is shown in Figure 11.

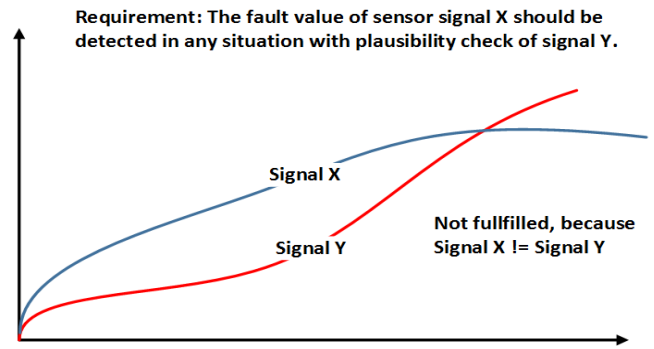


Figure 11 Verification and simulation

Fig. 12 shows the evaluation procedure of the defined hazards. The simulation is performed within the error causes in order to find out the system reaction of these failures. If the vehicle behavior is the same as defined hazards, then HARA is approved, otherwise HARA should be extended further as required from ISO 26262-4:2018, Clause 7.4.8 in order to develop the necessary safety measures to detect and prevent the possible hazards.

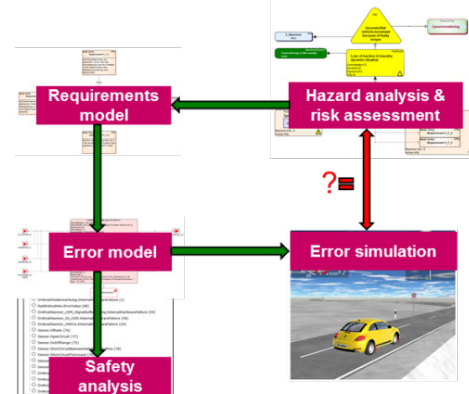


Figure 12 Error simulation

## 3. Use-Case

In order to show the benefits of the developed model-driven approach, an example is created which shows the safety-critical function development. It starts with hazard analysis and risk assessment and ends with verification of safety goals and safety requirements. Figure 13 shows a typical dependability model of a system. In this concrete case an electrical machine is considered as an item. Hazard is defined as “Uncontrolled vehicle movement because of faulty torque”. ASIL of hazardous event is classified based on the operational situation. In this case the hazardous event “Loss of traction in boundary dynamic situation” leads by driving a curve to a dangerous situation. Therefore the ASIL of this hazardous event is classified as ASIL D (C3, S3, E4). At the next step safety goal “No faulty torque in boundary dynamic situation” is defined with the safe state “Electric machine is torque-free, vehicle is rolling freely” in order to detect and avoid the failure. After this functional safety concept and technical safety concept are created in order to achieve the safety goal. Functional and technical safety requirements are linked from the requirements model and shouldn't be specified again in the dependability model.

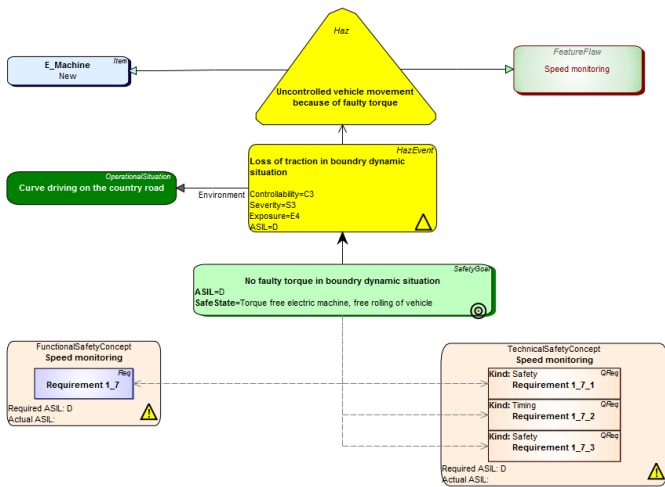


Figure 13 Dependability model – power train example

Functional and technical safety requirements are specified with associated information within the requirements model. Their allocation to architectural elements is realized just by using the satisfy connection. In this case the safety goal is realized with the function “Speed monitoring” (FAA) as shown in Figure 14.

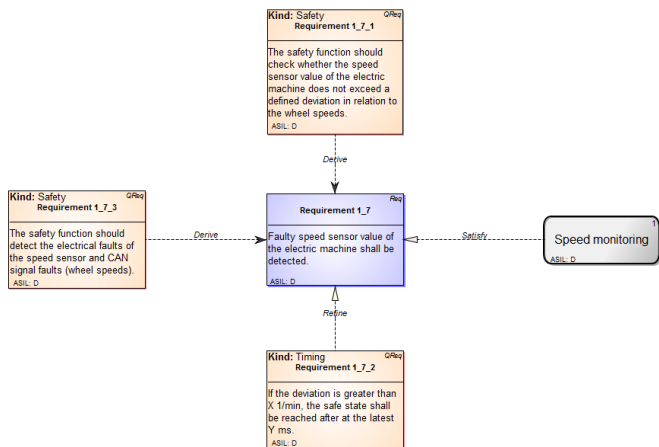


Figure 14 Requirements model – power train example

Functional and technical safety concept can be created automatically from the dependability model and requirements model using the developed script as shown in Figure 15.

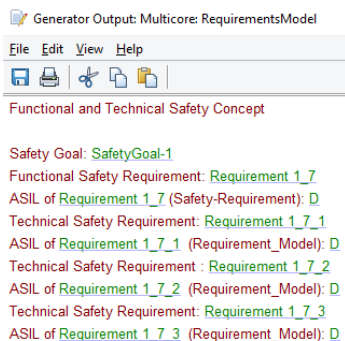


Figure 15 Functional and Technical Safety Concept – power train example

Figure 16 shows the FDA model. The safety requirements are implemented within the subfunctions of the FDA model which are

extended within safety attributes to enable consistency and traceability check.

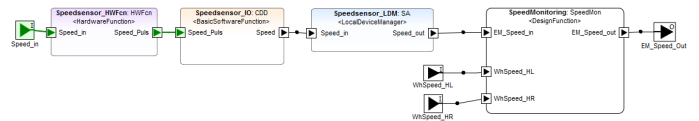


Figure 16 Functional design model – power train example

Error model of the hazard is generated automatically from FDA-Model. Afterwards the error logic of subsystems will be described individually with possible fault causes, considering the error propagation. Error logic contains all possible output failures of every submodule, with regard to internal hardware failures, internal software failures and input signal failures, as it is shown in Figure 17.

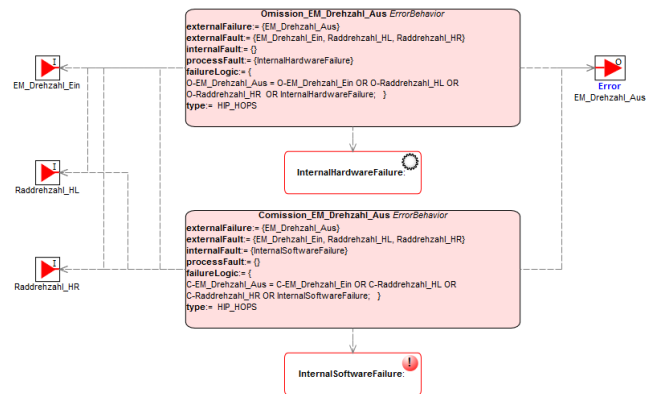


Figure 17 Error model logic – power train example

The additional tool HiP-HOPS enables to perform model based safety analysis automatically from the existing error models. This tool is capable to generate cut sets, fault tree analysis (FTA) and failure modes and effects analysis (FMEA). Figure 18 shows the generated FTA for the top event “faulty torque”. The causes such as sensor failures, internal software failures and internal hardware failures, which lead to the top event, are listed in the FTA. In this case a qualitative safety analysis is realized. The tool also enables to perform a quantitative analysis.

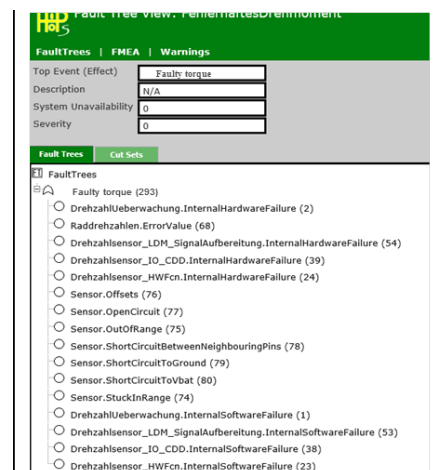


Figure 18 Model based safety analysis with HiP-HOPS – power train example

Finally, the error simulation is performed, in order to verify the system effects (top events) which were defined before within the HARA. For this reason the simulation is stimulated within the error causes (basic events) regarding fault tree analysis. The basic events should lead to the top events. After simulation the results are compared with the HARA results. If they are same, then the defined hazardous events are correct and thus verified. But if the vehicle behavior is other than defined events, the HARA should be extended. If necessary, a new safety goal should then be defined, in order to avoid new detected system impacts. In this example a concrete hazardous event (faulty torque due to E-Machine speed sensor failures) leads to loss of traction, which causes the vehicle to enter an instable state. In the boundary dynamic situations, e. g. when driving a curve, the driver cannot control the vehicle anymore. This may lead to a fatal accident. Figure 19 shows both the verification results of technical safety requirements and validation of safety goals regarding the error simulation.



Figure 19 Verification and Validation – power train example

#### 4. Conclusion

The presented approach supports developers in creating a model based system, software and safety architecture of E/E systems. Various tools are used currently for different special development fields. The architecture description language EAST-ADL, the main component of this approach, replaces these tools in a way that the entire development of system and safety architecture can be realized within EAST-ADL. The proposed approach has the advantage to verify the functional safety concept and technical safety concept and also to validate the hazard analysis and risk assessment within the developed simulation environment in the earlier development phase of project.

The biggest advantage of this method is to prove the traceability of the different development steps and safety work products with the help of extended library elements and developed scripts. Once the user became familiar with the method, it is easy for him to understand the overall architecture. Figure 20 highlights the relationship between the various levels and shows the traceability of different steps.

The approach enables to understand the system, software and safety architecture of an item very quickly. When applying this approach, requirement model, dependability model and library

elements of EAST-ADL is extended within safety features (such as ASIL classification of requirements) in order to model the safety-critical requirements with necessary attributes and in order to create functional and technical safety concepts automatically using developed script and in order to combine system and safety architecture. The user can thus find out quickly which requirements are implemented for which functions.

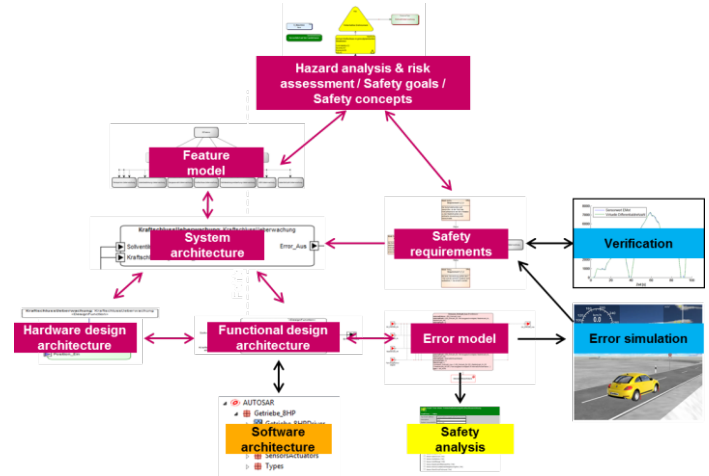


Figure 20 Traceability of the development steps

Another major advantage of the approach is the mapping of the EAST-ADL levels to the work products and requirements of ISO 26262. Already, at the initial phase the relationship between EAST-ADL and ISO 26262 is very significant. With dependability package of EAST-ADL, safety aspects of ISO 26262 can be early considered in the architecture development workflow. Thus, the hazard analysis and risk assessment is performed to find out the safety goals.

Dependability models allow the modeling of the necessary requirements for achieving the defined safety goals. Subsequently, the functional and technical safety concept can be created. So the user gets an overview, which safety concept is formulated for which safety goal. EAST-ADL offers the possibility to generate the error models automatically from FDA-models. With help of the approach and external analysis tools, safety analysis can be performed automatically from error models.

Systematic errors can be detected by the verification of the requirements and error simulation and it is also possible that safety measures are specified early in the development phase.

Finally, advantages of the presented approach can be summarized as follow:

- Modeling of safety-related functions in an architecture description language.
- Achievement of an efficient and consistent model-based development of automotive embedded system.
- Model based automated creation of ISO 26262 work products from hazard analysis and risk assessment to safety requirements using developed scripts.
- Combination of system and safety development to achieve the traceability and to show the relationship between safety goals and safety functions considering ASIL.

- Verification and validation of functional safety concept and technical safety concept in the earlier development phase of project.
- Early detection of systematic errors and system impact of errors.

## References

- [1] B. Sari, H.C. Reuss, "A model-driven approach for the development of safety-critical functions using modified Architecture Description Language (ADL), *Electrical Systems for Aircraft, Railway, Ship propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC 2016)*.
- [2] R. N. Charette, "This Car Runs on Code." Hyperlink: [http://www.real-programmer.com/interesting\\_things/IEEE%20SpectrumThisCarRunsOnCode.pdf](http://www.real-programmer.com/interesting_things/IEEE%20SpectrumThisCarRunsOnCode.pdf), Februar 2009. Accessed January 30, 2017.
- [3] PWC DEUTSCHLAND: Autoindustrie treibt Chipnachfrage an. Hyperlink: <http://www.pwc.de/de/automobilindustrie/autoindustrie-treibt-chipnachfrage-an.html>. Accessed January 30, 2017.
- [4] AK-L\_Orientation-list-V1.2\_2010-11-25\_DE (AA-I3/AK 16 - Functional Safety).
- [5] H. Blom, H. Lönn, F. Hagl, Y. Papadopoulos et al.: EAST-ADL – An Architecture Description Language for Automotive Software-Intensive Systems – White Paper Version 2.1.12. Hyperlink: [http://www.maenad.eu/public/conceptpresentations/EAST-ADL\\_WhitePaper\\_M2.1.12.pdf](http://www.maenad.eu/public/conceptpresentations/EAST-ADL_WhitePaper_M2.1.12.pdf). Accessed January 30, 2017.
- [6] P. Cuenot, D. Chen, S. Gerard, H. Lönn et al.: Managing Complexity of Automotive Electronics Using the EAST-ADL. In: *Engineering Complex Computer Systems, 2007. 12th IEEE International Conference on, 2007*.
- [7] ATESS2: ATESS2. Hyperlink: <http://www.atesst.org>. Accessed January 30, 2017.
- [8] MAENAD: MAENAD. Hyperlink: <http://www.maenad.eu/>. Accessed January 30, 2017.
- [9] SAFE: SAFE. Hyperlink: <http://www.safe-project.eu/>. Accessed January 30, 2017.
- [10] AUTOSAR: AUTOSAR. Hyperlink: <http://www.autosar.org/>. Accessed January 30, 2017.
- [11] HIP-HOPS: HiP-HOPS, Automated Fault Tree, FMEA and Optimisation Tool. Hyperlink: <http://hip-hops.eu/>. Accessed January 30, 2017.
- [12] ALTARICA: ALTARICA. Hyperlink: <http://openaltarica.fr/>. Accessed January 30, 2017.
- [13] ATESS2: EAST-ADL Overview. Hyperlink: [http://www.atesst.org/home/liblocal/docs/ConceptPresentations/01\\_EAST-ADL\\_OverviewandStructure.pdf](http://www.atesst.org/home/liblocal/docs/ConceptPresentations/01_EAST-ADL_OverviewandStructure.pdf), 2010. Accessed January 30, 2017.
- [14] M. Hillenbrand, „Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik / Elektronik Architekturen von Fahrzeugen“, Ph. D. Thesis, KIT Karlsruhe, 2011.
- [15] ATESS2: EAST-ADL2 Vehicle Level. Hyperlink: [http://www.atesst.org/home/liblocal/docs/ConceptPresentations/02\\_EAST-ADL\\_Vehicle\\_Level.pdf](http://www.atesst.org/home/liblocal/docs/ConceptPresentations/02_EAST-ADL_Vehicle_Level.pdf), 2010. Accessed January 30, 2017.
- [16] ATESS2: EAST-ADL Analysis Level. Hyperlink: [http://www.atesst.org/home/liblocal/docs/ConceptPresentations/03\\_EAST-ADL\\_Analysis\\_Level.pdf](http://www.atesst.org/home/liblocal/docs/ConceptPresentations/03_EAST-ADL_Analysis_Level.pdf), 2010. Accessed January 30, 2017.
- [17] ATESS2: EAST-ADL Overview Implementation Level. Hyperlink: [http://www.atesst.org/home/liblocal/docs/ConceptPresentations/05\\_EAST-ADL\\_Implementation\\_Level.pdf](http://www.atesst.org/home/liblocal/docs/ConceptPresentations/05_EAST-ADL_Implementation_Level.pdf), 2010. Accessed January 30, 2017.
- [18] ATESS2: EAST-ADL Overview Design Level. Hyperlink: [http://www.atesst.org/home/liblocal/docs/ConceptPresentations/04\\_EAST-ADL\\_Design\\_Level.pdf](http://www.atesst.org/home/liblocal/docs/ConceptPresentations/04_EAST-ADL_Design_Level.pdf), 2009. Accessed January 30, 2017.
- [19] ISO 26262 “Functional safety of road vehicles”, 2011.
- [20] EAST-ADL ASSOCIATION: EAST-ADL – Domain Model Specification – Version V2.1.12. Hyperlink: [http://east-adl.info/Specification/V2.1.12/EAST-ADL-Specification\\_V2.1.12.pdf](http://east-adl.info/Specification/V2.1.12/EAST-ADL-Specification_V2.1.12.pdf), 2013. Accessed January 30, 2017.
- [21] Synligare: Synligare (literally “more visible” in Swedish). Hyperlink: <http://synligare.eu/HomePage.html>. Accessed January 30, 2017.