Model-based Safety and Reliability Analysis on Functional Level for Hardware

Arne Maximilian Richter

Hamburg University of Applied Sciences, Dept. Computer Science,
Berliner Tor 7
20099 Hamburg, Germany
Email: arne.richter@haw-hamburg.de

I. Introduction

Nowadays we are surrounded by embedded systems. Some of these systems have a huge impact on our daily life and some even on our life itself. Therefore the correct functionality of these systems is important and is part of the system requirements.

To allow an engineer to meet these requirements some methods were introduced which are used to analyse the systems behaviour. Some of these methods have the goal to ensure safety by analysing possible unforeseen states of the system and allowing the engineer to deal with them. Sometimes the same and other methods are used to improve the reliability of the system.

System engineers are now trying to integrate these methods into their development processes to get a better usage out of them. These processes are normally model-based and fall under the topic of the Model-based System Engineering (MBSE). In parallel the methods are used in Model-based Safety Analysis (MBSA). Both kinds of processes are normally separated but work on the same system models. Therefore it just seems logic to integrate both processes into one and allow the engineer to get a direct feedback from the used tools about the system and the results of the analysis ([1], [2], [3], [4]).

The goal of this paper is to give a short overview of the current state of the art and to give a short assessment of how this can be used for the personal goal and the following work. To do so this paper will start in the following section II with the personal motivation for the work. Based on that section III will describe the personal goals and ideas. To complement these ideas section IV will give a short overview over the basics of safety and reliability analyses followed by section V with current approaches from the literature. Section VI will then give a short summary and conclusion about the presented approaches and how they can be used for the following work. As last section the section VII will give an outline about the following steps.

II. Motivation

Personal experiences have shown that the usage of the concepts of the MBSE and the MBSA can be really useful to the design of an embedded system consisting of hardware and software. Deviating from the system engineering the engineering of an embedded system currently seems less advanced in this direction.

While the the system engineering processes are evolving really strong the engineering processes for the integrated circuit diagrams are more or less stuck in their development. This can be confusing since the bigger systems developed by the MBSE processes include these embedded systems. Also adding confusion is the fact that embedded systems can be expressed in the same modelling languages the MBSE uses to develop and express the complete system. As a result the system might be analysed on the top level in a sophisticated way while subsystems goes through less effective tests.

These described problematics sometimes still run through the certification processes. National and international standards like the IEC 61508 for "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems" and the sub standards are requiring safety analyses through all life cycles of a system. The result of the analyses must be documented and the documents are required for the certification of the product. The work of analysis is often seen as kind of additional work for the developer and therefore disliked. Particularly when there is no additional team for the safety analysis it can happen that this work is done after finishing the development. Sometimes it is argued that the time required for a proper analysis is to long and if the design changes the work has to be done twice.

As a consequence design flaws stay undetected. The engineer is normally not able to see all the possible system states and therefore he can easily oversee these flaws. As the result they can stay undetected or will be detected during the final analysis for the certification.

When these flaws are detected in the last step the product is often finished. Prototypes had been build and tested and a lot of time and resources had been put into the development. This forces the development team to find a way to argue that the result is safe enough. Is it possible to explain why the flaw is under examination of the specific standards an acceptable risk the product will be released. If the risk is unacceptable the product goes back to development and causes the company a loose of time and resources.

Either way, all three situations with a flaw just shouldn't happen if they are preventable.

III. Objective

As mentioned at the beginning of section II, the processes of MBSE and circuit design are actually compatible. Also was mentioned in section I, that a lot of work is currently done

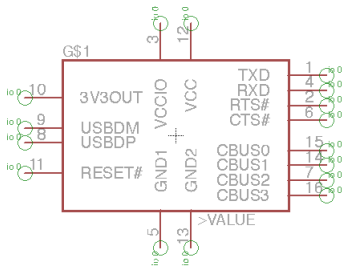

Figure 1. *Example of a component in Eagle.*

in combining MBSE and MBSA. As consequence it seems legit to attempt to integrate the automated tests and analyses of MBSA into the circuit design process.

The total goal is therefore to develop a concept which could be used by common tool for circuit design like Eagle[1]. Eagle and some of its competitors are already supporting a primitive form of hierarchical circuit design. This new approach in design allows the new idea to work based on abstraction concepts known from the software engineering.

Also current tools support connections to simulation programs for testing the analog circuits of the system. This shows the used steps of development. The engineer are doing tests as long as they help him to understand better the system. On the other hand the tools must be easy to use and must give useful and understandable results. To support this way of development the new Eagle competitors MultiSIM BLUE[2] and Scheme-it[3] included these simulations direct in the tool. To allow these simulations with the circuit designs the programs organise the parts in components. Figure 1 shows an example component generated with Eagle. This component is enriched by informations about the required metal plates on the circuit board, simulation behaviour for the analog simulation, and useful informations like vendors and so on.

The objective of the coming work is therefore to enrich the component with additional informations useful for safety and reliability analyses. These informations must be able to describe behaviour in case of internal faults and in case of failure spreading from components nearby. Therefore each port seen in the figure 1 must get additional informations about incoming and outgoing failures. As result automated safety and reliability analyses should be done automatically by the used tool.

Necessary informations for the engineer are in this case the behaviour of components when one related component is failing. The engineer needs to know how the fault of one component in the structure affects neighbour components and the ability of the whole system to execute its functions after that. Therefore the analysis method must be able to understand dependencies between signals like the ones used by Mariani [5] and Griessnig [6]. Both analysed architectures for a safety-critical system under the light of standards like IEC 61508. In detail, Mariani analysed how micro-controllers can be used to ensure safety by finding a way to detect software failures. Griessnig did this for a CPLD-based architecture. Therefore a goal of the work is to get a method that is capable of distinguishing between signal flows and understands redundancy and mitigations in the flows.

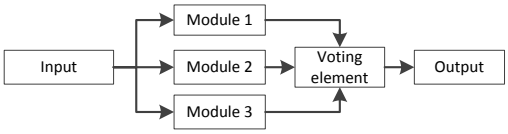

Figure 2. *A simple triple modular redundancy arrangement ([7]).*

In the example of the figure 2 the solution must be able to simulate a fault in one component and follow the direction of the signal. Is for example the module 1 corrupted the voting element must still be able to put out an accurate output. Therefore this single fault in module 1 is acceptable to the system safety and reliability. If another module is failing the system is in danger and the method must be able to detect this possible situation.

The required methods should be capable of understanding these kinds of dependencies and should allow a normal engineer to use the methods freely and most of all allow him to understand the results. The safety of software on the other hand should not be part of the solution. Components using software are often recommended to be seen as components with a 100% chance for a failure. Thereby the analysis of software won't be part of the objective or any sub goals.

IV. Safety and Reliability Analysis

To analyse a system in a proper form some techniques are required. First of all there must be something to analyse. For this purpose it is common to use a model of the system which should be analysed. The section IV-A therefore describes the basics of the common modelling techniques. Then the section IV-B gives an overview over the common techniques for analysing a model.

A. Models

Classic kinds of models are pure text or user defined figures. To solve the problem with the variety of the kinds of models the Object Management Group (OMG) published the standardised Unified Modelling Language (UML) for software ([8]). This solution nowadays even used in the field of hardware and mechanics to describe the system and its behaviour. As a result models described in UML can be used for analysing the system safety. Some research approaches, which will be described in section V, are building on the UML for automated analyses of systems.

With the increasing interest for the UML from not computer scientist the OMG released a second language ([9]). The System Modelling Language (SysML) is based on the UML and uses some of their diagrams. Also it contains additional diagrams which can be used to describe architectures in mechanical and electronically fields.

Friedenthal describes in [10] the current version of the SysML and how to use it for modelling a system. Safety

[1]http://www.cadsoftusa.com/eagle-pcb-design-software/about-eagle/
[2]http://www.mouser.com/multisimblue/
[3]http://www.digikey.com/schemeit

aspects however are not part of the current version of the language. Therefore the UML or the SysML are currently used to develop a system and the models are then analysed by experts. Most approaches to automate MBSA are based in models described in one of these languages.

B. Analysis

Using the provided models the system can be analysed with standard methods. As basis for the analysis the book [11] written by Leveson can be used. Leveson uses modern approaches for safety analyses which are not based on traditional assumptions like ”a system can only have on fault at the time”.

The approaches of Leveson might be the most modern and probably the best at the time. However the ideas are difficult to integrate into an automated process. The current work in this direction is based on classic approaches of safety analysis. Therefore the book [7] written by Storey might be the best approach since the current research is building up on these methods.

As basis of the analyses the HAZard and OPerability studies (HAZOP) can be used. This analytical technique is used to understand the behaviour of components in the system. It is based on ”what-if?” questions asked to experts. An example question from Storey is ”What would be the effect of an increase in temperature?” By using this kind of questions in the correct way the engineer should be able to understand his system in a more profound way and it should allow him to understand the effects analysed with other techniques. Based on these characteristics HAZOP can't be automated but should be done in parallel to improve additional techniques.

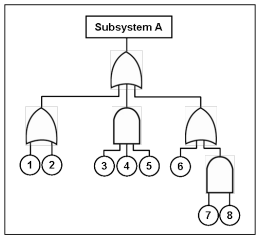

Figure 3. *Example of a fault-tree.*
Source: en.wikipedia.org

Using the knowledge acquired by using HAZOP and comparable techniques the system itself and the connections between the components should be analysed. To do so there are two standard techniques which were also described by Storey in [7]. The first technique is the Fault Tree Analysis (FTA) and the second one is the Failure Modes and Effects Analysis (FMEA).

The FTA uses a top-down approach for analysing a system. It takes a top-level hazard and analyses down which faults must occur to create the failure of the top level. The figure 3 illustrates a possible result. At the top is a possible hazard in the system or subsystem. In a respirator example this could be the stop of the respiration function. Going down the required faults are analysed and displayed as tree using the leafs as faults. In the case if the respirator leaf one and two can be the faults of the motor and a valve and the leafs three to five can be faults of the power supply and two emergency batteries therefore a fault of motor or valve will let the system fail but in case of the power supply both emergency batteries must also fail.

The automated generation of these trees are part of currently active research and the approaches will be discussed in section V-B.

The FMEA is a different approach which analyses the effects of failures in the system. To do so this technique is taking the possible incoming failures and the possible internal faults for a component and analyses the outgoing effects. Therefore this technique is working bottom-up and works in the opposite direction of the FTA.

To complement this procedure Schmittner published with [12] a new approach which added vulnerability analysis to the FMEA. This Failure Mode, Vulnerability and Effects Analysis (FMVEA) allows the analyst to take the security factor into the analysis. This concept is nowadays more and more important since safety-critical systems become more and more connected.

Both techniques are used to ensure safety and reliability in current systems ([11]). The usage of both techniques is required to reach the requirements for a product certification.

Before continuing with the resent approaches the technique of fault injection should be mentioned. Fault injection like done with the MODIFI concept by Svenningsson ([13]) analyses failure propagation. Therefore a component of the system is set into a fault condition. This can be done on an implementation of the system or in the model. The effect of the fault is then analysed and documented. These techniques are more or less tests and should be seen as alternative for techniques like FTA and FMEA.

V. Approaches in Literature

The basics of the last sections are the foundation for the current research approaches. Before going into detail with an approach it should be mentioned that this is a special field of research. The work of Leveson in [11] is for example is based on analyses of incidents that are caused by mistakes in the safety analysis. Most of these incidents are responsible for the death of more than hundred people.

Since this task is so critical the analysis of a system are mostly done in secret and the results are not published (to prevent possible law suits). Also the process of introducing new methods is slowly since they are used on systems like cars, planes, chemical and nuclear plans, and safety-critical medical equipment. These products have a long product life cycle and the quality of the techniques is normally first seen a few years after the start of the active use of the system. Therefore the community is really conservative and some new approaches are based on small changes done to approaches which came out a decade ago.

However there was a lot of research in the last years due to changes in modelling processes. The current research branches will therefore be organised in different sub sections. Section V-A will describe current approaches to deal with MBSA using UML with profiles which enrich the models with important informations. In section V-B an additional UML profile will be presented and an approach to synthesis a fault trees out of the model. Approaches dealing with SysML and FMEAs and FTAs will be presented in section V-C. A concept using a formal language for analysing a system in combination with

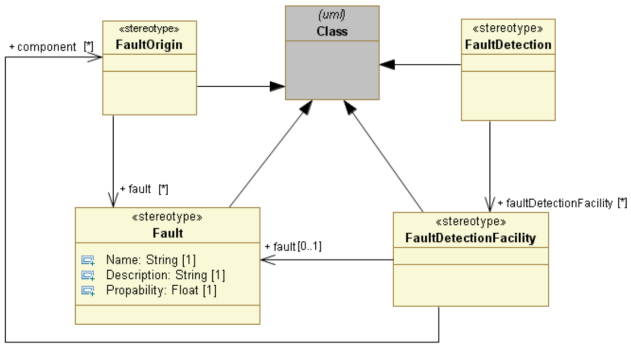

Figure 4. *Lauer's meta-model definition of the architecture model. Source: [2]*

a SysML model will be presented in section V-D. The last section V-E will introduce two concepts using a different modelling technique which is not developed by the OMG.

A. UML Profiles for Embedded Systems

The OMG published a UML profile to deal with real-time embedded systems. The "Modelling and Analysis of Real-Time and Embedded Systems" (MARTE) can be used to enrich UML models with required informations about schedulability and other important informations like concurrency and deadline specifications ([14]).

Based on the MARTE profile Bernardi published with [15] a UML profile for dependability analysis of real-time embedded systems. This profile is based on the OMG approach and is adding additional stereotypes for dependability. This allows the software engineer to let his system safety be analysed by tools.

The approach however is developed for software engineering and is not particularly designed to be used for hardware. On the other hand some of the concepts to model error detection and reparation might be adaptable for the analysation of hardware.

As an alternative the white paper [16] of Douglass presented an approach to describe a system on the structural hardware level which allows an automated fault tree generation. As an example Douglass uses a anesthesia machine for surgeries and describes how the UML profile can be used to generate the fault tree and to calculate the Mean Time To Failure (MTTF) and the total risk of the system.

While Bernardi's approach allows to analyse and verify software described by various UML diagram types the approach of Douglass gives a more simpler solution using UML relationship diagrams. According to Lauer ([2]) Bernardi's MARTE adaptation is also able to provide a tool with all the required informations to do analyses on the structure of a system's hardware. However the approach of Douglass is more determined for this specific objective of the structure analysis.

B. Synthesising Fault Trees using UML Models

In the publication [2] of Lauer a concept of synthesising fault trees out of UML profiles was described. To do the analysis Lauer used a reduced version of the UML profile of Bernardi. The objective of Lauer was to develop an UML profile which is capable of holding informations for techniques like FMEA and FTA. This reduced profile is based on the meta-model shown in figure 4.

This model is using a strongly reduced set of informations compared with the concepts of Bernardi and Douglass. The basic element of this concept is the fault. Faults contain a name, a description, and a probability. Informations like effects, severity, and risk are not included.

In this profile a component has a FaultDetection. This FaultDetection is able to react on several Faults by using FaultDetectionFacilities which are knowing a Fault and possible FaultOrigins. Therefore a component organised as a Class is able to detect faults and deal with them. Finally combined in a model with a few additional informations about the allocation and dependency of components a model can be used to synthesis a fault tree.

Using this profile as foundation of the work Lauer described the required algorithms to synthesis the fault tree. For displaying the fault tree he suggested an open source tool called openFTA[4][5]. Therefore the results can be used with already existing tool support.

C. FTA and FMEA Using SysML

Mhenni introduced a concept of doing MBSA using SysML structural diagrams ([17], [18], [19], [20], [21]). The concept supports techniques like FTA, FMEA, and Functional Failure Analysis (FFA) ([22]). Like the approaches of Douglass and Lauer Mhenni's concept is designed to work for mechatronic systems.

Mhenni's work started with [17] as a first analysis of the work of David [23]. The background designed by David was a process for automated FMEA synthesis out of SysML diagrams. This process should allow the engineer to generate the required tables for a FMEA. The concept however was not designed to do more than just the generation of the tables for the FMEA. The meta-model in David's profile didn't include any kind of mitigation ability or fault detection.

As a result Mhenni defined some requirements for the profile and the process. As goals Mhenni defined among other things the ability to do quantitativ analyses and the ability to deal with failure combinations which weren't include in the work of David.

The result is a profile which was introduced in [21]. This profile can deal with fault detection. It allows the engineer to automatically generate fault trees and prepare tables for an FMEA. Also the meta-model does include fields for informations that can be used to do a quantitative analyses of the model and determine the probability of a fault. To do so the work of Bernardi's [15] was combined with David's concepts from [23].

An independent approach did Pearce together with the earlier mentioned Friedenthal [24]. The objective of this approach was to combine the model with an FMEA. The meta-model of their profile however doesn't support more. It is designed to be synchronised with Microsoft Excel Spreadsheets and doesn't have the complexity of the approaches of Mhenni and David.

[4]http://www.openfta.com/

[5]openFTA and equal tools use standard file formats for storing the informations about the trees.

D. Verification of SysML Models

Arnold introduced with [25] the AltaRica language. This language allows the user to describe concurrent systems. The language was particularly designed to work with safety-critical systems. As a result the language can be used to describe systems and use tools to verify the behaviour in conditions of faults to verify safety and reliability requirements of a system.

As problem with AltaRica describes Helle in [3] the separation of development model and test model. AltaRica models are written after a system has been designed with another language like SysML. Therefore it requires additional work to maintain the actuality of the AltaRica model for verification. To solve this problem Helle introduced a concept to generate the AltaRica model automatically out of a SysML model.

As a proof of concept Helle used a smoke detector system. In the example all components were enriched with failure rates and the system was given a fixed structure. Using two different behaviours for the system the tool was able to calculate the system failure rates and give as a result informations about the verification of the required failure rates.

Differing from the previous approaches the concept of Helle is not preparing informations for a analysis by safety analysts or engineers. This approach requires concrete numbers of failure rates and maximal failure rates. In consequence the engineer can design his system and gets a feedback without consulting safety analysts for following steps.

E. Analyses using Simulink Models

In section IV-B the work of Svenningsson was already mentioned ([13]). The approach was a model-implemented fault injection tool and the result was called MODIFI. MODIFI allows the user to develop his software using Simulink[6] and then let it be tested with automated fault injections. The tool might be developed for safety-critical software but the concepts of simulating hardware faults based on a behaviour model can be useful.

Also using Simulink models Oertel developed with [26] an approach to analyse structure modelled with Simulink. The concept is designed to do FTAs as well as fault injection experiments. To do so Oertel uses a technique called safety contracts. The safety contracts are using a slim semantic to describe dependencies between components and events. As a result it is possible to generate fault trees and do fault injection experiments. The experiments and analyses can be done on models of hardware as well as on models of software. Analyses like FMEA however are not supported by the concept.

Oertel describes the contract system as capable of doing top-down analyses and experiments like a FTA and also button-up analyses and experiments like the fault injection. For analyses like an FMEA additional informations are required which are not included in the contracts. But if the models would be enriched with detailed informations of failure modes and the effects it would be possible to generate an FMEA out of the model.

[6]http://de.mathworks.com/products/simulink/

VI. Conclusion

There are many approaches in research, development, and use. Most of them are based on models build with UML or SysML. Other techniques for modelling are also in use but the most effort seemed to be put into concepts using one of the OMG languages. This speaks for the success in raising market shares of both languages in the field of system engineering.

The ability of both languages to describe complex mechatronic in a standardised way made them to a kind of standard for the industry. The fields they can be used in are various. The presented examples in the papers which were introduced in this work are going from small medical devices over cars to planes and submarines. The ability to deal with this variety combined with equal requirements for safety is probably the factor which pushed the development in the field of the MBSA in the last years.

The problem using the UML or SysML for the final objective of this work is the wide range of supported functions. Both can describe really complex system in sometimes different ways. As a result both languages are complex in use and might bring more features than required for a specific purpose. This doesn't meet the requirements for describing a circuit schema. On the other hand the structure diagrams of the SysML are comparable to the models used in circuit development (see figure 1). Therefore the concepts using the SysML structure diagrams can be transferred into a new approach for circuit development.

The approaches using Simulink look more or less the same like the ones used in circuit development and do support hierarchical organisation of the design like Eagle and the OMG languages.

Despite the importance of a correct model display the informations required for the safety and reliability analyses are more important for the complete process. For these required informations all previous approaches deliver some concepts and suggestions. For instance the UML profiles are designed to enrich UML models but the dependencies and the informations about the safety and reliability can be used as background for work with another model language.

Analysing these profiles more closely for the requirements of the future work the MARTE profile didn't meet any of the requirements. This profile doesn't contain the required informations for modelling the safety aspects. The update to the MARTE profile done by Bernardi can be used since required informations are included in the profile. It also allows to do quantitativ analyses to calculate the total risk for example. For an effective use the profile must be analysed and the additional informations required for analysing software should be removed.

Dougless's work as alternative to Bernardi's work can be used directly. The proof of concept was directly delivered and a manual for practical use is available. Dougless's profile does bring everything required to do FTAs but for being used as base for the future work it must be checked how effective it can be used to do bottom-up analyses since his approach only bases on a top-down FTA.

As a more light weight profile the Lauer approach can be seen. The approach is more handy and focuses more on the faults itself, their detection, and their propagation. Even when

Lauer's profile doesn't get used his algorithms for synthesising the fault trees can be used.

Under the concepts using SysML the profile of Pearce didn't bring any further aspects that can be used in the work. For the submarine in the example it might be an improvement of the workflow but taking the objectives of the future work in account this approach doesn't bring any use. Therefore the focus should be put on the alternatives like the work of David or Mhenni.

Since the work of Mhenni is based on David's and also Bernardi's approaches the work of Mhenni might be one of the most promising approaches. It supports top-down and bottom-up analyses, FTA and FMEA can be done with it, and might be capable of being used for fault injection experiments.

Alternatively the work done by Helle can be used. His work is differing strongly since he doesn't make use of analyses like FMEA and FTA. Using AltaRica with component failure rates, structural, and behaviour models the system can be analysed against concrete failure rates the system should satisfy. The failure rates can be taken from the supplier or from special documents like the military handbook MIL-HDBK-217F[7] which describes failure rates and error probabilities of components like transistors, memory banks, and micro-controllers. For the use in the future work it must therefore be guaranteed that the developer gets these informations. Since it is not normal that the suppliers like Farnell[8] add these informations of their goods it might get problematic. These kinds of informations are normally delivered for bigger machine components like motors but not for small components like motor-driver ICs.

Also an interesting approach was done by Svenningsson with MODIFI. MODIFI might not be interesting for the work for his test techniques it is more interesting for the automatism that is used to generate tests cases. This can be useful in combination with other approaches to generate test automatically. The other approach using Simulink was presented by Oertel. Oertel's concept using safety contracts can be used as alternative to the concepts presented with profiles for UML and SysML. The contract system supports FTA and fault injection and can probably be enriched to do analyses like an FMEA.

In total Oertel's, Helle's, Lauer's, and Mhenni's approaches are all possible basements for future work and should be tested, analysed, and compared in detail.

Some points were mentioned in the different publications and should be a focus part during the comparison. For example did Lauer and Oertel describe in [2] and [27] that one of the biggest issues the variability of the components is. It is common practices to use the same sub components in different projects. Therefore the technique needs to be able to deal with this variability and should be designed in a way that every system needs to be set up from the beginning without any kind of library for components.

Also it should be analysed how the techniques can deal with multiple errors. For that Xiao describes in [28] a basic principe and also Young ([29]) and Leveson ([11]) promote that a technique must be able to deal with multiple errors. Therefore it must be taken in account for the up coming work.

[7] http://www.sre.org/pubs/Mil-Hdbk-217F.pdf

[8] http://de.farnell.com/

VII. Following Steps

Based on the research the following steps can be defined. As first step the possible failure modes which should be detected and analysed must be determined. These can be failure modes like a delay in a data transmission between two components or an occurrence of over current. These are the necessary basics for the later project.

Aware of the required failure modes the approaches of Oertel, Helle, Lauer, and Mhenni should be tested and analysed. Questions are if they detect all the required failures, how they deal with multiple errors, in which way they support variability, the amount of work they require for doing the analyses, and how easy they are to understand. Particularly the last points are important for the practical field. If the system is too complex to understand or there is too much work to do with no adequate result it may never deploy on the market.

An additional question which should be checked is how the system should deal with security. As mentioned by Schmittner ([12]) security plays nowadays a more important role in the development of embedded system. Connected cars and maybe connected traffic lights need to deal with security aspects. The question is thereby if the hardware should be involved as a factor and should a controller conntected to the internet be seen as a component with additional failure modes like an external takeover.

After dealing with these questions and defining the requirements for the final technique the final steps consist of the development of the necessary profile for the components, selecting the chosen analyses and tests, and combine them in a way that they can be used by tools like Eagle.

References

[1] M. de Miguel, J. Briones, J. Silva, and A. Alonso, "Model based integration of safety analysis and development," in Object and Component-Oriented Real-Time Distributed Computing, 2006. ISORC 2006. Ninth IEEE International Symposium on, April 2006, pp. 2 pp.–.

[2] C. Lauer, R. German, and J. Pollmer, "Fault tree synthesis from uml models for reliability analysis at early design stages," SIGSOFT Softw. Eng. Notes, vol. 36, no. 1, Jan. 2011, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1921532.1921558

[3] P. Helle, "Automatic sysml-based safety analysis," in Proceedings of the 5th International Workshop on Model Based Architecting and Construction of Embedded Systems, ser. ACES-MB '12. New York, NY, USA: ACM, 2012, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/2432631.2432635

[4] A. Garro and A. Tundis, "A model-based method for system reliability analysis," in Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium, ser. TMS/DEVS '12. San Diego, CA, USA: Society for Computer Simulation International, 2012, pp. 2:1–2:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2346616.2346618

[5] R. Mariani and P. Fuhrmann, "Comparing fail-safe microcontroller architectures in light of iec 61508," in Proceedings of the 22Nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, ser. DFT '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 123–131. [Online]. Available: http://dl.acm.org/citation.cfm?id=1302493.1302731

[6] G. Griessnig, R. Mader, C. Steger, and R. Weiss, "Design and implementation of safety functions on a novel cpld-based fail-safe system architecture," in Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on, March 2010, pp. 206–212.

[7] N. R. Storey, Safety Critical Computer Systems. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1996.

[8] Object Management Group, "Introduction To OMG's Unified Modeling Language," July 2005, checked on 14.08.2013. [Online]. Available: http://www.omg.org/gettingstarted/what_is_uml.htm

[9] ——, "OMG Systems Modeling Language," checked on 14.08.2013. [Online]. Available: http://www.omgsysml.org/

[10] S. Friedenthal, A. Moore, and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.

[11] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety (Engineering Systems). The MIT Press, Jan. 2012.

[12] C. Schmittner, T. Gruber, P. Puschner, and E. Schoitsch, "Security application of failure mode and effect analysis (fmea)," in Computer Safety, Reliability, and Security, ser. Lecture Notes in Computer Science, A. Bondavalli and F. Di Giandomenico, Eds. Springer International Publishing, 2014, vol. 8666, pp. 310–325. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10506-2_21

[13] R. Svenningsson, H. Eriksson, J. Vinter, and M. Törngren, "Model-implemented fault injection for hardware fault simulation," in Model-Driven Engineering, Verification, and Validation (MoDeVVa), 2010 Workshop on, Oct 2010, pp. 31–36.

[14] Object Management Group, "The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems," January 2013, checked on 14.08.2013. [Online]. Available: http://www.omgmarte.org/

[15] S. Bernardi and J. Merseguer, "A uml profile for dependability analysis of real-time embedded systems," in Proceedings of the 6th International Workshop on Software and Performance, ser. WOSP '07. New York, NY, USA: ACM, 2007, pp. 115–124. [Online]. Available: http://doi.acm.org/10.1145/1216993.1217012

[16] B. P. Douglass, "Analyze system safety using UML within the IBM Rational Rhapsody environment," IBM Software Group, White paper, June 2009. [Online]. Available: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&htmlfid=RAW14124USEN

[17] F. Mhenni, J. Choley, A. Riviere, N. Nguyen, and H. Kadima, "Sysml and safety analysis for mechatronic systems," in Mechatronics (MECATRONICS) , 2012 9th France-Japan 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM), 2012 13th Int'l Workshop on, Nov 2012, pp. 417–424.

[18] F. Mhenni, N. Nguyen, H. Kadima, and J. Choley, "Safety analysis integration in a sysml-based complex system design process," in Systems Conference (SysCon), 2013 IEEE International, April 2013, pp. 70–75.

[19] F. Mhenni, J.-Y. Choley, and N. Nguyen, "Extended mechatronic systems architecture modeling with sysml for enhanced safety analysis," in Systems Conference (SysCon), 2014 8th Annual IEEE, March 2014, pp. 378–382.

[20] F. Mhenni, N. Nguyen, and J.-Y. Choley, "Automatic fault tree generation from sysml system models," in Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on, July 2014, pp. 715–720.

[21] F. Mhenni, J.-Y. Choley, and N. Nguyen, "Sysml safety profile for mechatronics," in Mecatronics (MECATRONICS), 2014 10th France-Japan/ 8th Europe-Asia Congress on, Nov 2014, pp. 29–34.

[22] Y. Papadopoulos, J. McDermid, R. Sasse, and G. Heiner, "Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure," Reliability Engineering & System Safety, vol. 71, no. 3, 2001, pp. 229 – 247. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832000000764

[23] P. David, V. Idasiak, and F. Kratz, "Reliability study of complex physical systems using sysml," Reliability Engineering & System Safety, vol. 95, no. 4, 2010, pp. 431 – 450. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832009002671

[24] P. Pearce and S. Friedenthal, "A practical approach for modelling submarine subsystem architecture in sysml," in Proceedings from the 2nd Submarine Institute of Australia (SIA) Submarine Science, Technology and Engineering Conference, October 2013, pp. 347–360. [Online]. Available: http://www.omgsysml.org/A_Practical_Approach_for_Modelling_Submarine_Sub-system_Architecture_in_SysML-Pearce_Friedenthal.pdf

[25] A. Arnold, G. Point, A. Griffault, and A. Rauzy, "The altarica formalism for describing concurrent systems," Fundam. Inf., vol. 40, no. 2,3, Aug. 1999, pp. 109–124. [Online]. Available: http://dl.acm.org/citation.cfm?id=2378994.2378996

[26] M. Oertel, O. Kacimi, and E. Böde, "Proving compliance of implementation models to safety specifications," in Computer Safety, Reliability, and Security, ser. Lecture Notes in Computer Science, A. Bondavalli, A. Ceccarelli, and F. Ortmeier, Eds. Springer International Publishing, 2014, vol. 8696, pp. 97–107. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10557-4_13

[27] M. Oertel, M. Schulze, and T. Peikenkamp, "Reusing a functional safety concept in variable system architectures," in Proceedings of the 7th International Workshop on Model-based Architecting and Construction of Embedded Systems co-located with ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Valencia, Spain, September 30th, 2014., 2014, pp. 16–25. [Online]. Available: http://ceur-ws.org/Vol-1250/paper3.pdf

[28] N. Xiao, H.-Z. Huang, Y. Li, L. He, and T. Jin, "Multiple failure modes analysis and weighted risk priority number evaluation in fmea," Engineering Failure Analysis, vol. 18, no. 4, 2011, pp. 1162 – 1170. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1350630711000288

[29] W. Young and N. Leveson, "Systems thinking for safety and security," in Proceedings of the 29th Annual Computer Security Applications Conference, ser. ACSAC '13. New York, NY, USA: ACM, 2013, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/2523649.2530277