HAL
open science

SysML safety profile for mechatronics

Faïda Mhenni, Jean-Yves Choley, Nga Nguyen

To cite this version:

Faïda Mhenni, Jean-Yves Choley, Nga Nguyen. SysML safety profile for mechatronics. 10th France-Japan/8th Europe-Asia Congress on Mecatronics (MECATRONICS), Nov 2014, Tokyo, Japan. 10.1109/MECATRONICS.2014.7018622 . hal-01910896

HAL Id: hal-01910896

https://hal.science/hal-01910896v1

Submitted on 9 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SysML Safety Profile for Mechatronics

Faida Mhenni, Jean-Yves Choley
LISMMA
SUPMECA
Saint-Ouen, France
faida.mhenni,jean-yves.choley@supmeca.fr

Nga Nguyen
LARIS
EISTI
Cergy, France
nga.nguyen@eisti.eu

***Abstract*—Safety analysis of mechatronic systems is a time-consuming activity, because of the complexity of these systems that involve different fields of engineering. It is desirable to carry out safety assessment methods as soon as possible in the design process in order to reduce errors, cost and time to market of the system. Our paper addresses this problem by proposing a safety profile that is integrated directly with the modeling elements of the system via SysML, a model-based systems engineering language. Failure modes of each function and each component, their causes, their effects as well as their severity are modeled via stereotypes or tag definitions that extend the existing UML elements. These failure data can be 1) entered directly by systems engineers when possible; and then 2) generated automatically for safety experts' work; and also 3) updated from safety analysis results. Our integrated systems engineering and safety analysis process helps to narrow the gap between these two disciplines by ensuring the consistency in the whole process. A case study with an electromechanical actuator is given to illustrate the process as well as the safety profile.**

I. Introduction

Nowadays systems often have to provide the end-user with more functionality while being confined in a compact volume and being less energy consuming. Mechatronics offers the opportunity to design such smart systems. The complexity of mechatronic systems mainly relies on the merging of very different technologies such as mechanical devices to actuate, embedded software and electronic hardware for command and control purposes, with respect to an appropriate automation strategy. Mechanical parts, electronic components and software routines are on their own error sensitive, thus needing thorough safety studies in order to make them safe and reliable. When it comes to the integration of all these components in a so-called smart mechatronic system, usual safety assessment methods are not self-sufficient and have to be widely extended and supported by new design tools and methodologies [1], [2]. Since systems engineering is almost becoming compulsory for designing such complex artificial systems [3], an integrated safety analysis and systems engineering process is our proposal. This means that there is a need to customize the usual systems engineering framework in order to have it adapted for safety expert's activities. The Unified Modeling Language (UML) [4] is a general-purpose modeling language in the field of software engineering, which comes with the ability to be customized using the profiling extension mechanism. Numerous system design activities use UML profiling to have built-in domain-specific functionality available in a dedicated design framework, such as for RTES (Real-Time Embedded Systems) design with MARTE (Modeling and Analysis of Real Time and Embedded systems) or for systems engineering domain with SysML. Thus, our goal is to provide the safety experts and systems engineers with tools and methodologies relying on a domain-specific profile for safety analysis for mechatronics, based on SysML, a widely used systems engineering language.

The paper is organized as follows. Section 2 discusses several famous UML extensions for domain specific applications. Section 3 presents our methodology for the integration of safety analysis in systems engineering process. Section 4 explains the proposed Safety Profile using together with SysML models to support the methodology. An electromechanical actuator example is studied in Section 5 and conclusions are given in the last section.

II. Related Work

As a generic UML extension mechanism, a profile is a collection of stereotypes, stereotype attributes and constraints applied to specific UML modeling elements (classes, activities, ...) in order to customize UML for specific domains and platforms. For this purpose, stereotypes extend UML vocabulary to create new domain-specific modeling elements derived from generic UML ones. Four examples of profile namely SysML, MARTE, Mechatronic UML and UPDM are presented thereafter.

SysML is a profile proposed by OMG (Object Management Group) with INCOSE specifications for systems engineering [5]. It offers new specific diagrams such as Requirement and Parametric diagrams. It also provides modified UML 2.0 diagrams such as Activity diagrams, Block definition diagrams and Internal block diagrams. SysML is now well-known as an efficient systems engineering language, though very few guidelines have been provided by OMG and INCOSE, thus being slowly adopted by potential industrial users. When used with other profiles such as MARTE, this profile can be relevant for mechatronic systems modeling [6].

Replacing the former UML profile for Schedulability, Performance and Time Specification (SPTP), MARTE is the UML 2.0 OMG standard for Real-Time Embedded systems modeling and Analysis (RTEA), dealing with software and hardware aspects. It provides non-functional property modeling such as performance, scheduling; adds time features; defines concepts for software and hardware platform modeling; and provides quantitative analysis [7].

In [8], Zohaib et al. discuss experiences about applying MARTE profile for Real Time Embedded Systems (RTES) design. Three industrial experimentations are explained. The first one deals with architectural modeling and configuration,

applied to ICSs (Integrated Control Systems), which are heterogeneous systems-of-systems. In these systems, software and hardware components are integrated to control and monitor physical devices and processes, such as process plants or oil and gas production platforms. The request of the industrial partner (FMC Technologies Inc.) was mainly for capturing software and hardware components interactions, enhancing consistency between software and hardware modeling and, finally, enabling automated configuration and configuration reuse. The second experience deals with model-based robustness testing, using also stereotypes from Robustprofile from Simula Research Laboratory [9]. The project aimed at supporting automated, model-based robustness testing of SATURN, a video conferencing system from CISCO Systems Inc. The last experience deals with Testing RTES using MARTE environment models on a large and complex seismic acquisition system with tens of thousands of sensors and actuators in its marine environment (WesternGeco and Tomra). Models were used to generate an environment simulator, test cases, and obtain test oracles.

MARTE is also a UML profile for AADL (Architecture Analysis and Design Language), a Domain-Specific Language (DSL) that deals with the hardware platform and the physical environment of intensive embedded software systems. It can be used to model application tasks and communication architectures, thus allowing modeling and analysis of coupled software and hardware RTES aspects.

Mechatronic UML is a specific profile for mechatronic systems. It is derived from the safety-critical software development domain with the main objective of bringing model-based design, domain testing and simulation, and formal analysis to the mechatronics area [6]. The aim behind this is to guarantee highly safety-critical system properties. The profile restricts the usage of UML to certain types of diagrams and extends these diagrams to be able to model hybrid and self-adaptive systems. The use of formal semantics to model the components allows formal analysis. This profile aims at being more efficient than approaches such as SysML that does not adequately support modeling of time, and MARTE that does not provide the needed architectural abstraction for hardware [6].

The fourth profile is UPDM (Unified Profile for DoDAF/MODAF) [10]. DoDAF (Department of Defense Architecture Framework, US), MODAF (Ministry of Defense Architecture Framework, UK) and NAF (NATO Architecture Framework) use this profile to define architecture frameworks that are domain specific and define practices for creating, interpreting, analyzing and using architecture descriptions, as described in ISO/IEC/IEEE 42010 [11].

Concerning the integration of safety analysis and systems engineering, there are a lot of works that have been carried out by different researchers [12]–[17]. The main safety techniques studied in these papers are Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) [17]. Since the major objective of this paper is the SysML Safety Profile, please refer to our previous paper for a more detailed comparison of these related research works [18].

III. Integrated Systems Engineering and Safety Analysis Process

In this section, the integrated process of systems engineering and safety analysis is presented through a set of steps.

- **Step 1: Requirements definition and analysis**- In this step, system functionalities as well as its external interfaces are described by a set of requirements. Several SysML diagrams such as use case diagrams and block definition diagrams for the system context can be used to help in the identification of these requirements. For more detail about requirement definition and analysis please refer to the steps of the *black box analysis* in [3].
- **Step 2: Functional architecture definition**- Based on the functional requirements identified in step 1, one or more functional architectures are identified during this step. The final result is a hierarchical model of the breakdown of the system main function(s) into sub-functions. In SysML, functions are represented by activities and the functional breakdown is modeled through a set of activity diagrams, each activity diagram representing the breakdown of a given function (activity) into sub-functions. Activity diagrams also show the progressive transformation of input flows into output flows.
- **Step 3: Functional risk assessment**- In this step, a functional Failure Mode and Effects Analysis (FMEA) is used to identify potential hazards caused by failures and their effects. In this work, a tool has been developed to automatically generate partial FMEA based on the XML Metadata Interchange, XMI [19] file generated from the SysML model. The generated FMEA contains the data-sheet with the list of the functions and a generic list of failure modes. The safety expert then performs the analysis and completes the FMEA with the relevant data. All these safety information are then incorporated into the SysML model via the safety profile extension that will be explained in section IV. The gap between safety analysis and design modification is shorten, thanks to this integrated model.
  At the end of this step, safety requirements are derived and added to the set of requirements. The rule is that for each failure mode with hazardous effects, at least one safety requirement is added. Design changes can be done from this early design stage at the functional level to eliminate or reduce identified risks. Risk effects mitigation can be obtained by eliminating or modifying high risk functions, adding new fault tolerance mechanisms like diagnosis and reconfiguration functions, etc. Each time that the functional architecture is modified, the FMEA shall be updated to take into account the new changes. The previous steps iterate until a satisfactory solution is identified.
- **Step 4: Logical architecture definition**- Once the functional architecture is defined taking into account the results of the safety analysis in step 3, one or more logical architectures are defined by allocating components to functions. A Block Definition Diagram

(BDD) describes the components of the system and an Internal Block Diagram (IBD) describes the interactions between the components. The logical architecture defined at this step already takes into account safety aspects since it integrates the results of the functional safety assessment performed in step 3.

- **Step 5: Component-level risk assessment**- When the structure of the system is defined, the safety analysis results are updated and a component level risk assessment is performed. For this purpose, a component FMEA is generated from the XMI file like in step 3. To ensure consistency with previous safety analysis, the generated FMEA, in addition to the components, contains in front of each component the functions allocated to the component as well as the failure modes identified at the functional level as a reminder. The safety expert then identifies the failure modes at the component level and performs FMEA analysis. If there are identified risks at a non acceptable level, then these risks shall be eliminated or reduced to an acceptable level by performing changes to the design. Once again, these safety data are saved back in the same SysML model.
- **Step 6: Fault tree analysis**- The final step is the fault tree analysis. Fault trees are used for both qualitative and quantitative analyses. In our approach, fault trees are automatically generated from SysML IBDs describing the system architecture. Information from the previous FMEA analysis is taken into account to create fault tree with specific failure modes. Fault trees can be generated in a graphical form for qualitative analysis purposes like fault propagation studies and critical paths identifications. They can also be generated in an appropriate format for existing fault tree analysis tools. For more details about fault tree generation please refer to [18].

IV. Safety Profile

During the design phase, the designers (systems or domain engineers) may have relevant information concerning safety especially if they are integrating new concepts or innovating technology. In this case, they are recommended to transmit these data to safety experts. And in the opposite direction, it is important for a safety expert to pass on safety analysis results to systems engineers to take into account in the system design. In order to integrate safety information directly into SysML models, we have used the extension mechanism of UML to create a so called Safety Profile. A profile allows adaptation or customization of UML meta-models to a specific platform, domain or method through stereotype and tag definition concepts [4]. The stereotype is the primary extension construct that extends an existing meta-class. A stereotype may have properties that are referred to as tag definitions. In our case, the Safety Profile is built from stereotypes and tag definitions that represent FMEA artifacts such as failure modes, causal factors, system effects, probability, severity, etc. The relationships between this information are modeled by a class diagram given in Fig. 1.

Since in our methodology, in the functional decomposition step (step 2, section III), a system function is represented by an activity, it is straightforward to consider Function as a stereotype extending the Activity meta-class. A system component is a SysML block, so the "Component" stereotype will extend the Class meta-class of UML. Because each activity may have several parameters and each class may have several attributes, we propose to use Parameter and Attribute as extended meta-classes for FailureMode stereotype. By doing so, we can represent the fact that each function and each component may have different failure modes. The other information about a failure mode such as rate, severity, causal factors, detection methods, etc can be simply considered as the tag definitions of the Failure Mode stereotype. Fig. 2 gives a partial profile diagram of our Safety Profile. In a more general framework, we model also the redundancy mechanism as well as dysfunctional behavior information such as degraded and failed states, safety requirements with their formal test cases, etc. In the scope of this paper, we only present the FMEA view of the Safety Profile. It is also noted that there is no unique solution for the safety profile. We prefer to use a simple and efficient solution that allows us to represent all needed information while not overloading the XML Metadata Interchange file generated from the SysML model.

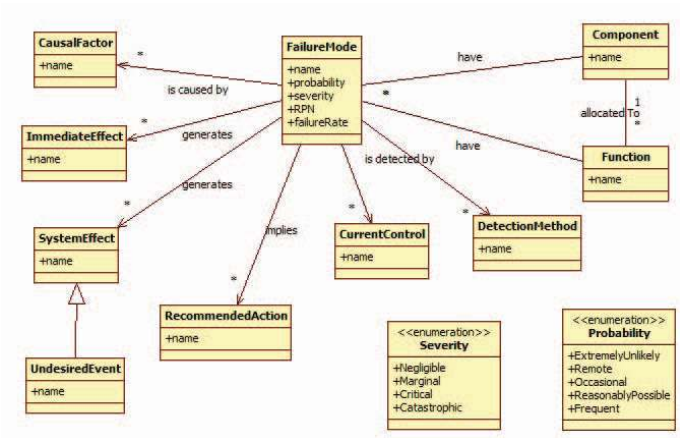


Fig. 1. Class Diagram for FMEA Artifacts

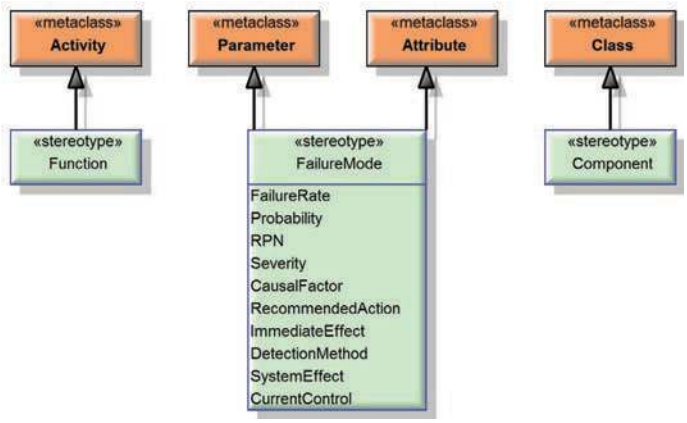


Fig. 2. Safety Profile Diagram

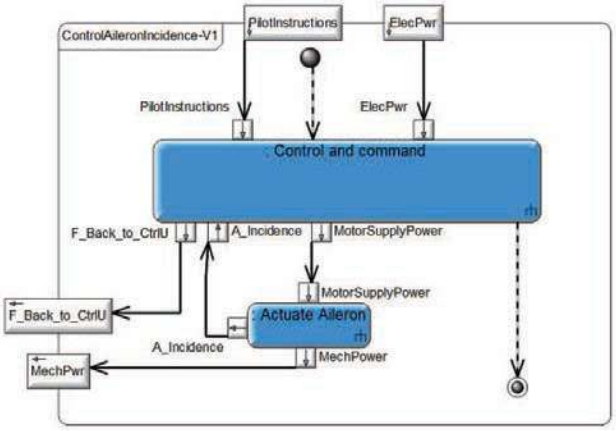
V. Case Study

In this section, a case study is presented to illustrate the proposed method. An example of electromechanical actuator

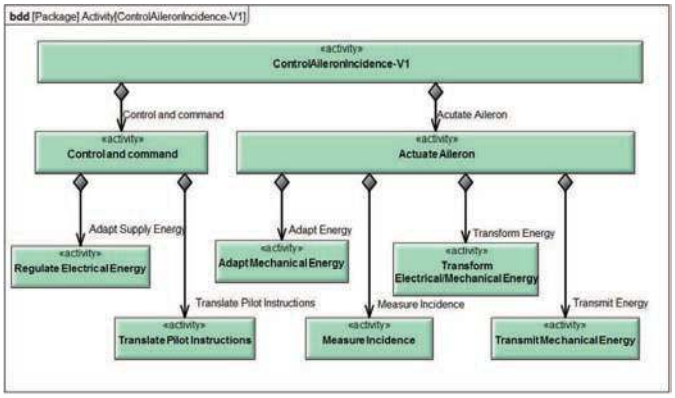
(EMA) is considered. The EMA aims at actuating the ailerons in an aircraft. The use of EMAs in flight control is increasing since they have many advantages [20]:

- Better environmental respect with suppression of hydraulic power and oil leak risks;
- Weight saving on aircraft;
- Maintenance cost reduction;
- Performance increase and speed accuracy due to electric actuators.

In **step 1**, the requirements of the system are defined. The main functional requirement of the EMA is to "Control the Aileron Incidence". In this step, a *black box* analysis is performed to identify the requirements by analyzing the context of the system as well as the functional scenarios. For more detail about the different steps of the *black box* analysis, please refer to [3]. Then, in **step 2**, the functional architecture of the system is determined based on the functional requirements identified in **step 1**. An example of functional architecture of the EMA is given in Fig. 3.



(a) Top Level Functional Breakdown



(b) Functions Hierarchy

Fig. 3. Functional Architecture

Thanks to the safety profile, any data about safety available at the design stage can be integrated in the system model. If some functions have specific failure modes, the designers can integrate them in the system model so that the safety expert could take them into account. For instance, Fig. 4 shows

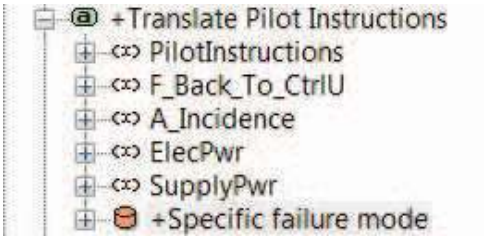


Fig. 4. Example of Failure Mode Added in the System Model

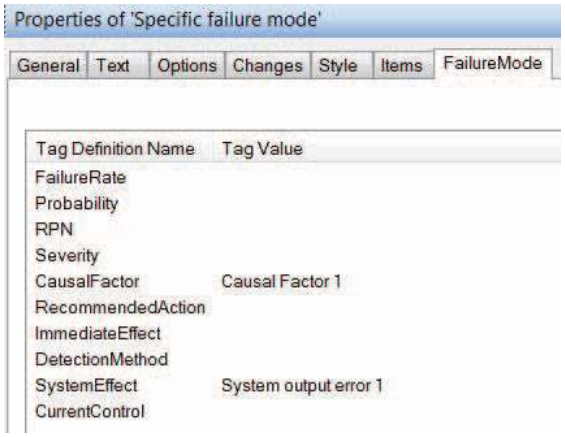


Fig. 5. Failure Mode Properties Added in the System Model

a failure mode "Specific failure mode" added in the system model for the function "Translate Pilot Instructions". This is done via the FailureMode stereotype which is an extension of the Parameter meta-class of the activity "Translate Pilot Instructions". This stereotype allows adding some properties to the failure mode in the system model which will be included in the generated FMEA as well. An extract of the generated functional FMEA is given in Fig. 6 including the "Specific failure mode" as well as the corresponding causal factor and system level effect entered in the system model (manually highlighted in red for the readers).

Function	Function failure mode	Causal factors	Immediate Effects	System Effects	Re
Regulate Electrical Energy	Fails to perform				
	Performs incorrectly				
	Operates inadvertently				
	Operates at incorrect time				
	Unable to stop operation				
	Receives erroneous data				
	Sends erroneous data				
Translate Pilot Instructions	Fails to perform				
	Performs incorrectly				
	Operates inadvertently				
	Operates at incorrect time				
	Unable to stop operation				
	Receives erroneous data				
	Sends erroneous data				
	Specific failure mode	Causal factor 1		System output error 1	
	Fails to perform				

Fig. 6. Generated Functional FMEA

In **step 3**, and based on the functional decomposition of **step 2**, an FMEA is generated with the list of the functions in the system model. For each function, generic failure modes are

added. Other important information such as potential causes and effects are also pre-filled. An extract of generated data-sheet for the EMA is given in Fig. 6. The safety expert then completes the data-sheet, identifies critical functions and tries to eliminate or reduce the effect of potential risks. Additional data completed by safety expert are then saved back to the SysML model via Safety Profile elements.

For this example, several single failures may lead to the system effect "Aileron locked" which may result into a loss of control of the aircraft. This risk shall be reduced by adding a new function "Internal Diagnosis". The system shall be able to detect failures and prevent the catastrophic effects they may have. New iteration of FMEA is performed to take into account the modified design. The added function "Internal Diagnosis" will be integrated in the FMEA and its failure modes and their impact on the system behavior will be analyzed. We consider that this is the only modification of the design and we will then move to the **step 4** where the system structure is defined by allocating components to functions. The resulting system structure is given in Fig. 8.

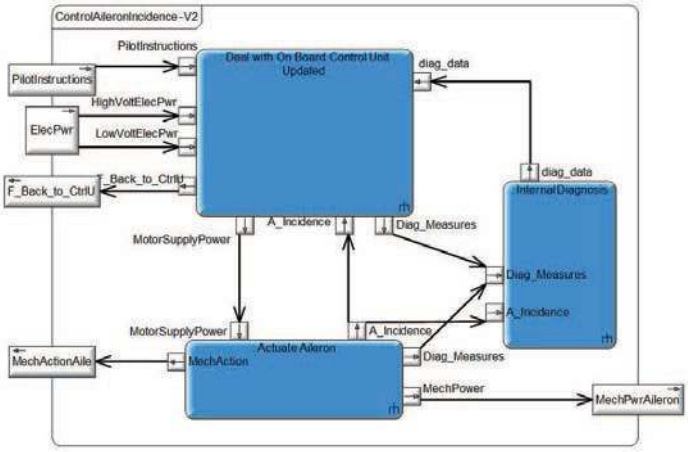


Fig. 7. Updated Functional Architecture

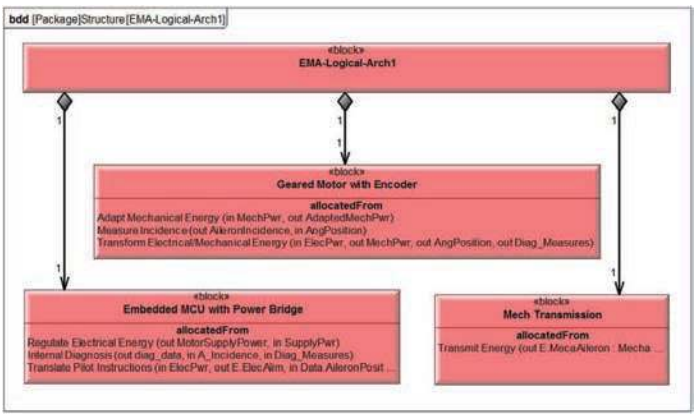


Fig. 8. System Structure

In the same way, a component FMEA is generated based on the system structure model (**step 5**). It contains the list of components, and for each component, the list of allocated functions (activities) to improve the consistency between the functional analysis and the component analysis. If several structural solutions are proposed, then several FMEAs are generated and the different solutions can be compared according to safety analysis. If any changes or improvements can be done to the system structure based on the results of safety analysis, then new iterations are performed to take into account the design changes and assess their impact on system safety. In the same way as for the functional FMEA, the safety profile allows systems engineers or designers to integrate some safety aspects into the system model and these will be automatically integrated into the generated component FMEA. Fig. 9 shows an example of component failure mode added in the system SysML model. The same properties as in Fig. 5 can be added in the system model for each failure mode.

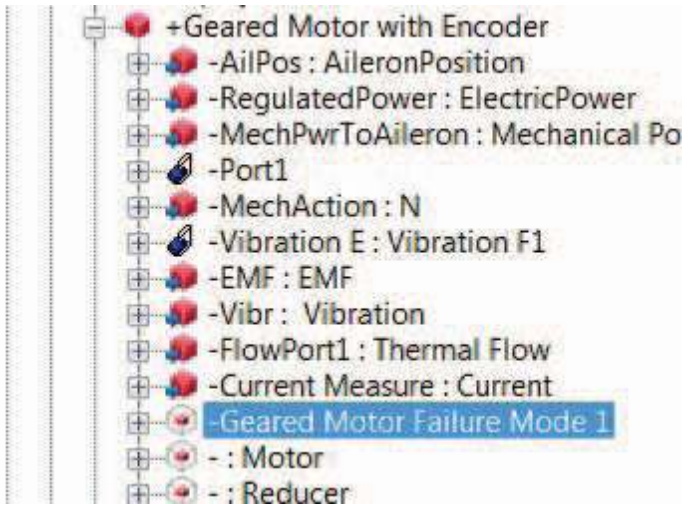


Fig. 9. Example of Failure Mode Added to a Component in the System Model

The last step is **step 6** relative to fault tree analysis. This step is based on the results of the previous design steps and safety analyses. More detail about fault tree generation can be found in [18].

VI. Conclusion

In this paper, we have proposed an implementation of the Safety Profile that extends some UML modeling elements to represent FMEA information namely failure modes, probability, severity, causes and effects. Combining this profile with SysML elements such as activities for functions and blocks for components, we are able to carry out the preliminary safety analysis within the mechatronic system design process by using the integrated methodology.

Automatic generation of FMEA data-sheets and of fault trees from the system structures can be a good support for safety experts. The consistency between the latest design modifications and the safety information is ensured by this SysML Safety Profile. In addition, our integrated framework can help systems engineers to keep traces between safety requirements and different functional and architectural solutions, which is very important for safety-critical mechatronic systems.

The further step of our work is to enrich the Safety Profile to take into account different fault tolerance mechanisms such as redundancy policies, as well as to integrate formal methods to validate system dynamic behaviors with respect to safety

requirements. Another extension of the profile concerning some mechatronic issues such as the connection components and the multi-physical interactions among components is also studied.

REFERENCES

[1] R. Cressent, P. David, V. Idasiak, and F. Kratz, "Dependability analysis activities merged with system engineering, a real case study feedback," in *Advances in Safety, Reliability and Risk Management: ESREL*, 2011.

[2] F. Mhenni, N. Nguyen, H. Kadima, and J.-Y. Choley, "Safety Analysis Integration in a SysML-Based Complex System Design," in *IEEE International Systems Conference SysCon, Orlando, USA.*, 2013.

[3] F. Mhenni, J.-Y. Choley, O. Penas, R. Plateaux, and M. Hammadi, "A SysML-based methodology for mechatronic systems architectural design," *Advanced Engineering Informatics*, vol. 28, no. 3, pp. 218 – 231, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474034614000342

[4] http://www.uml.org/.

[5] http://www.omgsysml.org/.

[6] W. Schäfer and H. Wehrheim, "Model-driven development with Mechatronic UML," in *Graph Transformations and Model-driven Engineering*, G. Engels, C. Lewerentz, W. Schäfer, A. Schürr, and B. Westfechtel, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 533–554. [Online]. Available: http://dl.acm.org/citation.cfm?id=1985522.1985549

[7] M. Faugere, *MARTE: Also a UML profile for AADL*, SAE AADL meeting Seattle ed., Thalès Research & Technology, 2009.

[8] M. Iqbal, S. Ali, T. Yue, and L. Briand, "Experiences of applying UML/MARTE on three industrial projects," in *Model Driven Engineering Languages and Systems*, ser. Lecture Notes in Computer Science, R. France, J. Kazmeier, R. Breu, and C. Atkinson, Eds. Springer Berlin Heidelberg, 2012, vol. 7590, pp. 642–658.

[9] S. Ali, L. Briand, and H. Hemmati, "Modeling robustness behavior using aspect-oriented modeling to support robustness testing of industrial systems," *Software & Systems Modeling*, vol. 11, no. 4, pp. 633–670, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10270-011-0206-z

[10] UDPM-Group and OMG, "Advancing MBSE via unified profile for DoDAF/MODAF (UPDM)."

[11] *Systems and software engineering - Architecture description*, ISO/IEC/IEEE Std., 2011.

[12] H. Dubois, "Gestion des exigences de sûreté de fonctionnement dans une approche IDM," in *Journées Neptune N 5, Paris, France*, 08 avril 2008.

[13] P. David, V. Idasiak, and F. Kratz, "Reliability study of complex physical systems using SysML," *Reliability Engineering and System Safety*, vol. 95, no. 4, pp. 431 – 450, 2010.

[14] R. Cressent, P. David, V. Idasiak, and F. Kratz, "Designing the database for a reliability aware model-based system engineering process," *Reliability Engineering & System Safety*, vol. 111, no. 0, pp. 171 – 182, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832012002177

[15] F. Thomas and F. Belmonte, "Performing safety analyses and SysML designs conjointly : a viewpoint matter," in *Complex Systems Design & Management*, 2011.

[16] R. Laleau, F. Semmak, A. Matoussi, D. Petit, A. Hammad, and B. Tatibouet, "A first attempt to combine SysML requirements diagrams and B," *Innovations in Systems and Software Engineering*, vol. 6, pp. 47–54, 2010.

[17] R. Guillerm, H. Demmou, and N. Sadou, "Global safety management method in complex system engineering," in *Fourth Inernational Conference on Complex Systems Design and Management, CSDM*, 2013.

[18] F. Mhenni, N. Nguyen, and J.-Y. Choley, "Automatic fault tree generation from SysML system models," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2014.

[19] *XML Metadata Interchange (XMI) Specification*, Object Management Group Std.

[20] D. Mami, "Définition, conception et expérimentation de structures dáctionneurs électromécaniques innovants incluant par conception des fonctionnalités de sûreté et de sécurité de fonctionnement," Ph.D. dissertation, Université de Toulouse, Institut National de Politechnique de Toulouse, 2010.