INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

MODELING OF THE FAILURE PROPAGATION OF AN ADVANCED MECHATRONIC SYSTEM WITHIN THE SPECIFICATION OF ITS PRINCIPLE SOLUTION

R. Dorociak and J. Gausemeier

Keywords: conceptual design, mechatronics, failure propagation, reliability, dynamic fault trees

1. Introduction

The products of mechanical engineering and related industrial sectors, such as the automobile industry and railway technology, are often based on the close interaction of mechanics, electronics and software engineering, which is aptly expressed by the term mechatronics. Due to the involvement of different domains mechatronic systems are characterized by a high complexity; the design of such systems is very challenging. This especially holds true for the assurance of reliability. Indicators for this are the great number of product recalls and increasing warranty costs of the recent years. Most failures can be traced back to insufficient cooperation and communication of the involved domains. Usually these failures are first recognized very late in the product engineering process, as established reliability methods are typically conducted on detailed system designs. Yet: the later failures are discovered, the more does it cost to eliminate or at least mitigate them. Therefore, there is a need for reliability analysis methods and tools, which can be applied in the early engineering phase of conceptual design, so that failures can be detected at an early stage. Time consuming and costly iteration loops can then be avoided.

In our previous work we have developed a method for the description of failure propagation of an advanced mechatronic system. It allows modeling of failure propagation within the specification of the principle solution of the system; the principle solution is specified by means of the specification technique for the domain-spanning description of the principle solution of an advanced mechatronic system from the Collaborative Research Centre (CRC) 614 “Self-optimizing concepts and structures in mechanical engineering”. In this contribution, an extension of the method for the description of failure propagation of an advanced mechatronic system is presented. The extension is inspired by the state-of-the-research Dynamic Fault Tree framework. In particular, new gates for modelling of functional dependencies and sequence dependencies between failures as well as spare allocation are incorporated into the method. The method allows first principal statements with regard to the reliability of the system as well as to related aspects such as availability in the early development phase of conceptual design. Based on them counter measures are incorporated into the product conception at an early stage. Thus, a great number of time-consuming and costly iteration loops are avoided.

This paper begins with a brief overview over the specification technique for the domain-spanning description of the principle solution of an advanced mechatronic system. A short introduction of the case study – the innovative autonomous railway vehicle RailCab – follows. State of the art and previous work are then presented. Finally, the extended method for the description of failure

propagation of an advanced mechatronic system is introduced and its applicability is shown using the case study of the RailCab system.

2. Domain spanning specification of the principle solution of an advanced mechatronic system

The development of advanced mechatronic systems is divided into the discipline-spanning conceptual design and the discipline-specific concretization. During the conceptual design, experts from the disciplines of mechanical engineering, electrical engineering, control and software engineering work together and develop the principle solution. The principle solution defines the basic structure and operational mode of the system and its desired behaviour; it is the result of the conceptual design. During the following concretization, the involved disciplines develop their discipline specific aspects in parallel. Characteristic for this development phase is the high coordination and communication effort.

For the description of the domain-spanning principle solution of an advanced mechatronic system we use the specification technique CONSENS (CONceptual design Specification technique for the ENgineering of complex Systems), which has been developed within the CRC 614 [Gausemeier et al. 2009]. The description of the principle solution is divided into 8 aspects: environment, application scenarios, requirements, functions, active structure, shape, behavior and system of objectives (Figure). The aspects are computer-internally represented as partial models. The partial models are strongly interrelated and form a coherent system. In particular, the cross-references between the partial models are modeled. For example, it is modeled, which requirements are concretized by which functions. During the specification of the principle solution it is necessary to work alternately on the aspects although there is a certain order. A procedure model for the domain-spanning specification of an advanced mechatronic system has also been developed, which defines the constituent steps of the conceptual design, their result and their order and is fully aligned with the specification technique CONSENS [Gausemeier et al. 2009].

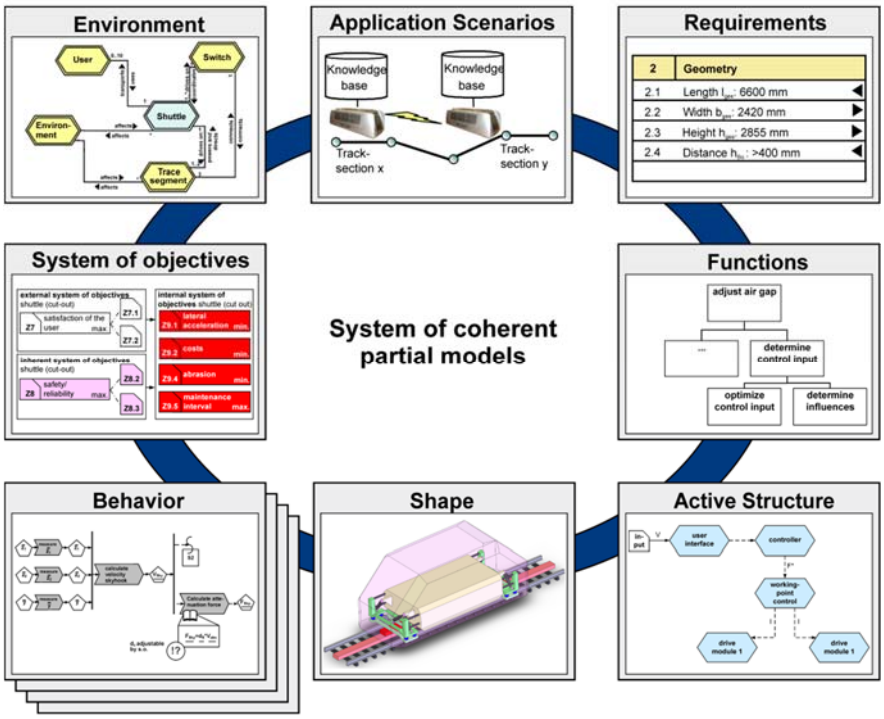

Figure 1. Aspects respectively partial models for the domain-spanning description of the principle solution of a mechatronic system

The specification of the principle solution provides all relevant information for the structuring of the system and forms the basis for the communication and cooperation of the developers from different domains. In contrast to other system modeling approaches such as UML or SysML [Friedenthal et al. 2008] the specification technique is strongly interconnected with the underlying procedure model and focuses strongly on mechatronic systems. For a detailed description of the specification technique, its partial models and the underlying procedure model, please refer to [Gausemeier et al. 2009].

3. Innovation field railway technology

The innovative railway system RailCab [RailCab 2011] is used as a demonstrator of the CRC 614 and serves as a case study throughout this contribution. A test facility on a scale of 1:2.5 has been built at the University of Paderborn (Figure). It is a highly modular system, which consists of a linear drive, an active guidance module, an active spring and tilt module and a hybrid energy storage system.

Figure 2. Prototype of the innovative autonomous railway system RailCab (scale 1:2.5)

The whole RailCab system has been specified using the specification technique for the domain-spanning description of the principle solution of an advanced mechatronic systems. In the following. some of the results for the spring and tilt module of the RailCab are shown. Each spring and tilt module has three servo-cylinders, which damp vibrations and tilt the vehicle body in curves. Each servo-cylinder consists of a hydraulic cylinder, a 4-4-way valve, a servo cylinder regulation and a hydraulik valve regulation [RailCab 2011]. Figure shows a cut-out of the principle solution model of the spring and tilt module.

4. Related work

The most established reliability engineering methods in the industry nowadays are Fault Tree Analysis (FTA) and Failure Modes and Effect Analysis (FMEA). FMEA [Birolini 2007] is a method, which describes possible failures of a system element[1] as well as its possible causes and consequences in a tabular form. For the identified causes and consequences the respective detection and counter-measures are determined and also recorded in the FMEA table. A substituent part of a FMEA is the risk priorisation by means of the risiko priority number (RPN).

FTA [Birolini 2007] is a deductive method. An undesirable failure, the top event, is first postulated. The possible failure which can lead to the occurrence of this top event are then determined. The analysis of fault trees is typically realised using the binary decision diagram (BDD) approach [Birolini 2007].

Component Fault Trees [Kaiser et al. 2003] extend traditional fault trees with a notion of components, which are connected using ports. The logical structure of a CFT is no longer a tree, but a directed acyclic graph. They go beyond traditional Fault Trees: 1) in CFT repeated events are represented only once and 2) several top events can be analysed. For the analysis of CFTs algorithms and data structures (e.g. BDDs) known from the traditional FTA can still be applied.

[1] System elements are used in the partial model active structure. A system element represents a part of the system, which has not been detailed yet. System elements are detailed in the course of the product development process and can be consolidated into modules, parts, assemblies and software components.

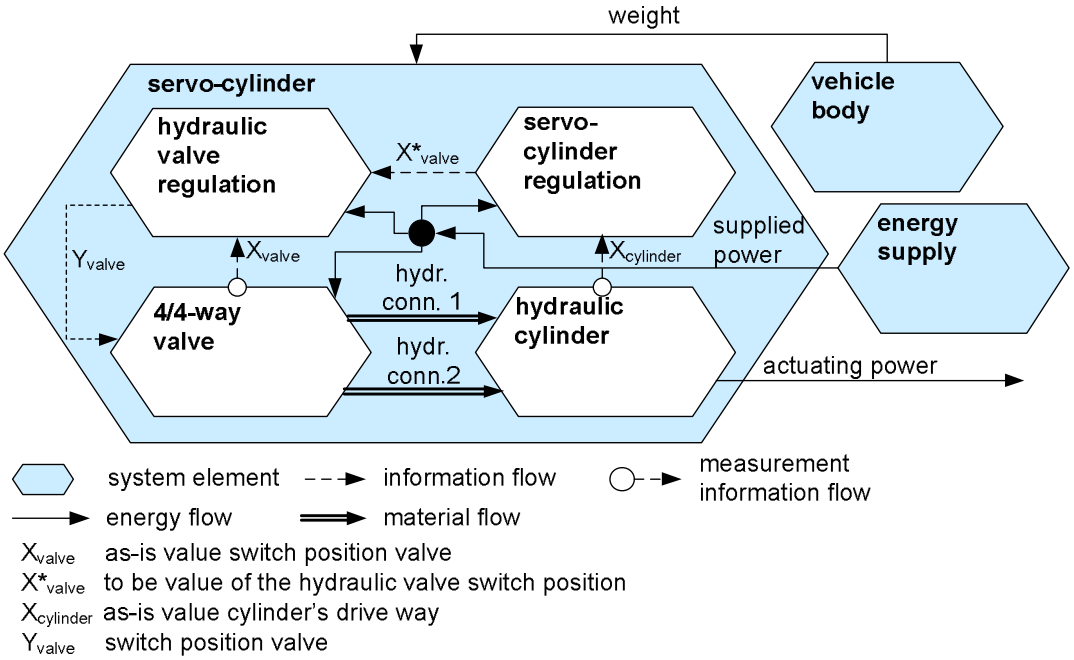

Figure 3. Active structure of the spring and tilt module (cut-out)

Another extension of standard Fault Trees are Dynamic Fault Trees (DFT) [Dugan et al. 1992, Birolini 2007]. They introduce new kind of gates, which can capture various additional dependencies between failures, e.g. dependencies concerning the occurrence order of failures. For analysis of DFTs Continuous Time Markov Chains (CTMC) [Birolini 2007] are used. Codetta-Raiteri [Codetta-Raiteri 2005] defined a transformation of DFT to Generalized Stochastic Petri Nets (GSPN) by means of graph transformation rules. A mapping of DFT onto Dynamic Bayesian Network (DBN) can be found in [Codetta-Raiteri et al. 2010]. Based on the resulting transformed representation reliability analyses are performed and counter-measures derived. The definition of a formal semantics for Dynamic Fault Trees can be found in [Boudali et al. 2010].

5. Previous work

Based on the specification of the principle solution, early reliability analyses can be performed. A number of approaches has been developed, which go in this direction.

In the CRC 614 an approach for an early FMEA (Failure Mode and Effects Analysis) based on the principle solution has been developed. System elements and functions of the system under development are transferred from the principle solution model into the FMEA table. The remaining steps of the FMEA failure analysis, risk assessment and optimization are performed then. The approach has been applied and validated on a dry vacuum pump of a notable German pump manufacturer in a transfer project of the CRC 614.

An method for the description of failure propagation of an advanced mechatronic system within its principle solution has been developed in the cooperative project "Early reliability analysis of mechatronic systems" (InZuMech) of the German Federal Ministry of Education and Research [Deyter et al. 2009]; it is based on the Fault Tree framework. For each system element its local and incoming failures as well as their impact on the outputs are modeled. Each input and output has a state tuple. The constituent failure states are usually called *ok* and *not(ok);* the state *not(ok)* represents the undesired behavior. For the description of the failure propagation the relationships *canImply*, *needs* and *canTolerate* and Boolean gates OR, AND, XOR etc. and voting gates (e.g. 2/3 voting gate) are used. The *canImply* relationship describes which combination of internal failures and faulty inputs can lead to an faulty output (i.e. *not(ok)* state). With the *needs* relationship it is modeled which state the inputs are ought to be in, for the output to be in state *ok*; this kind of modeling is very common in the field of functional security. The *canTolerate* relationship describes to which extent the outputs of a

system element can compensate internal failures and faulty inputs. Figure 4 shows an example of the specification of the failure propagation within an exemplary system element *SE*.

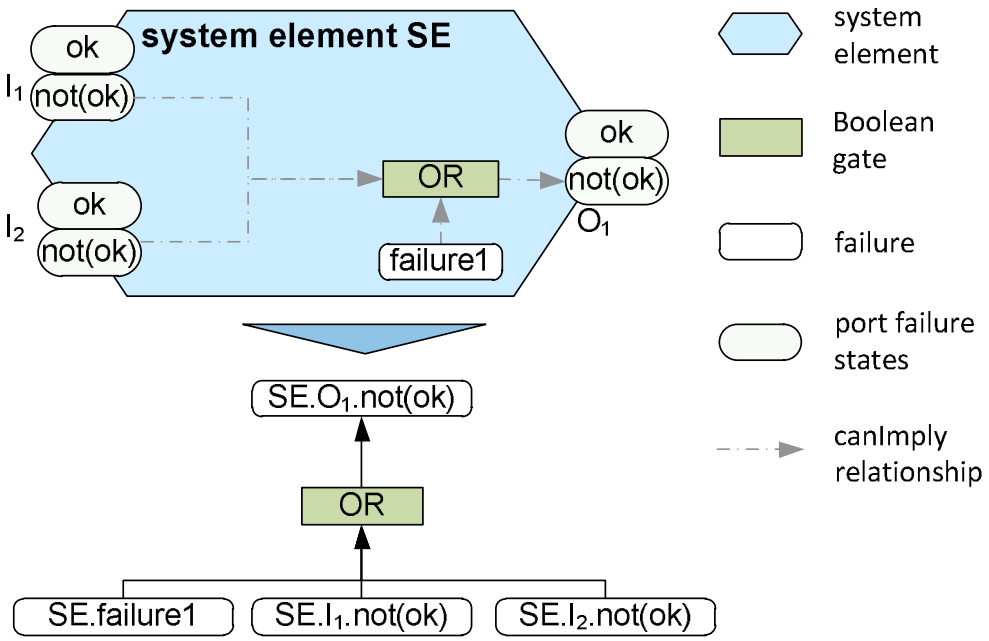

Figure 4. Specification of the failure propagation and the corresponding fault tree

The output O_1 of the system element *SE* exhibits an undesired system behavior, if the internal failure "failure1" occurs or one of the both inputs (I_1, I_2) is faulty. Based on such a description of the failure propagation a fault tree can be generated automatically [Deyter et al. 2009]. Finally established FTA analyses such as minimal cut sets, etc. [Birolini 2007] are conducted and the system reliability is improved.

6. Modeling of the failure propagation of an advanced mechatronic system within the specification of its principle solution

In this contribution the method for the description of failure propagation of an advanced mechatronic system within its principle solution has been extended. We incorporated the dynamic gates from the DFT framework (Section 4), which enable modelling of functional dependencies, sequence dependencies as well as spare allocation. In the following we present how our method has been extended and how the extensions are mapped on the Dynamic Fault Tree framework.

For the modelling of functional dependencies between failures the Functional Dependency gate (FDEP) is used. It consists of a triggering failure (*T*) and a set of dependent failures. When the triggering failure occurs, all dependent failures occur as well. Figure 5a) shows the adapted FDEP gate, specified within the system element *SE*. When the triggering failure T occurs, the outputs O_1, …, O_n switch to the state *not(ok)*. The corresponding DFT is also shown in Figure 5a). In particular, Common Cause Failures[2] can be modelled using the FDEP gate.

Sequence dependencies are modelled using the Priority AND (PAND) gate. An example is shown in Figure 5b). The PAND gate is connected to a number of incoming failures F_1, ..., F_n by ordered edges and to one outgoing failure *O1.not(ok)*. The outgoing failure occurs, when all of the incoming failures are true and they occurred in the order given by the ordered edges (here: $F_1 < \ldots < F_n$).

[2] Common cause failures (CCFs) are two or more system element failures occurring at the same time or within a relatively short time interval due to a common cause, e.g. a harsh environment influence such as vibration, that causes the failure of multiple system elements. Another example is the impact of cosmic radiation on circuit technology.

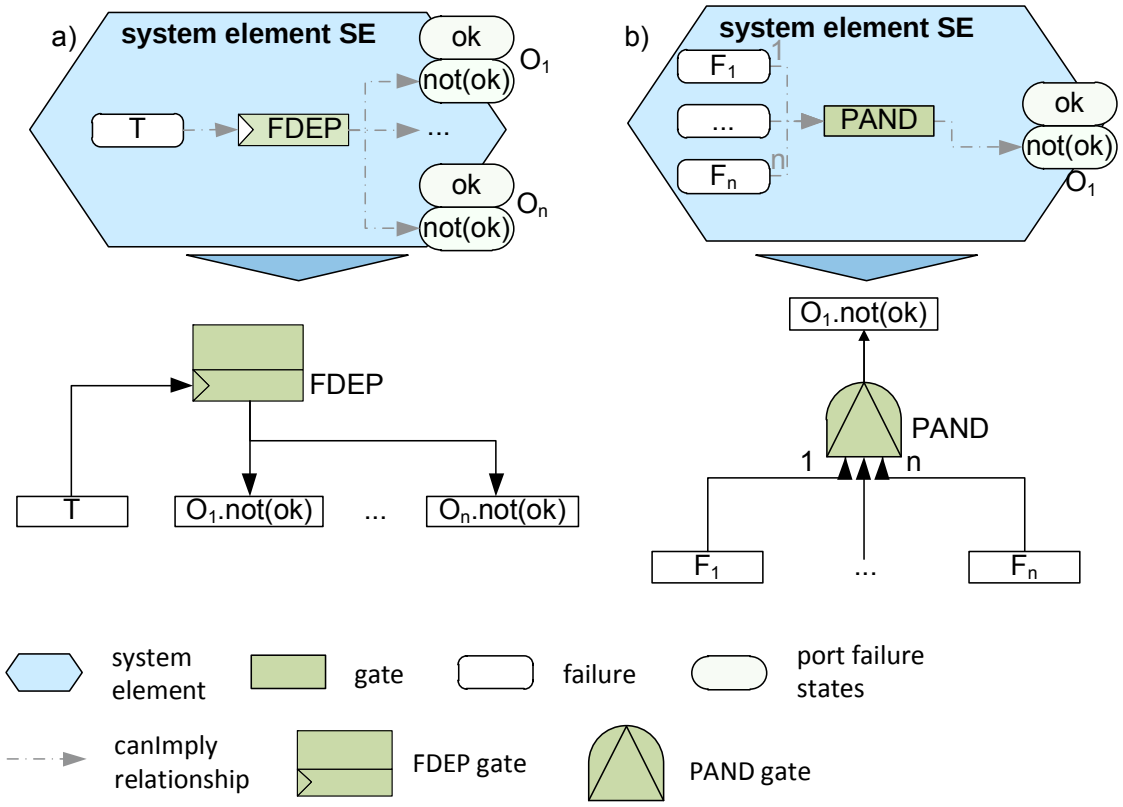

Figure 5. Specification of the failure propagation with a FDEP gate (a) and a PAND gate (b) as well as the corresponding dynamic fault trees

The existence and allocation of spare system elements is expressed by using the spare gate (SPARE). Figure 6 shows an example specification using the SPARE gate; the corresponding DFT is also depicted. For the main system element *SE_main* there exists a number of spare system elements *SE_spare_1*, …, *SE_spare_n* which are used, if the main system elements fails and replace it in its function and behaviour. The spare system elements can be in three operational states: standby (also called dormant), active or failed; initially each spare system element is in the standby state. Spares are connected to the main component using ordered edges, which define the order of replacement, e.g. if *SE_main* fails, it is replaced by *SE_spare_1*, if *SE_spare_1* fails, it is replaced by *SE_spare_2*, etc. The outgoing failure *F* occurs when the main system element *SE_main* and all of its spares had failed. The exchange of the main component by a spare may occur either automatically or manually (e.g. during system maintainance).

In our example (Figure 6) the main system element and its spares are modelled as instances of the system element template *SE_template*. The advantage of this approach is, that the subordinated structure and behaviour are modelled only once in the template and then reused (instantiated) several times within the specification of the partial model active structure.

One spare system element can be shared by several (main) system elements. Three kinds of spare system elements are distinquished: cold, warm and hot. The distinquishement is made upon the so-called dormancy factor. Let λ be the failure rate of an active spare. The failure rate of a spare system element in the dormant state is $\alpha\lambda$, where α is the dormancy factor and $0 \leq \alpha \leq 1$. If the dormancy factor is 0, the spare system element is called a cold spare and cannot fail while being in the dormant state. If the dormancy factor is 1, it is a hot spare; its failure rate is the same in active as well as in dormant state. If the dormancy factor is between 0 and 1, the spare system element is called a warm spare.

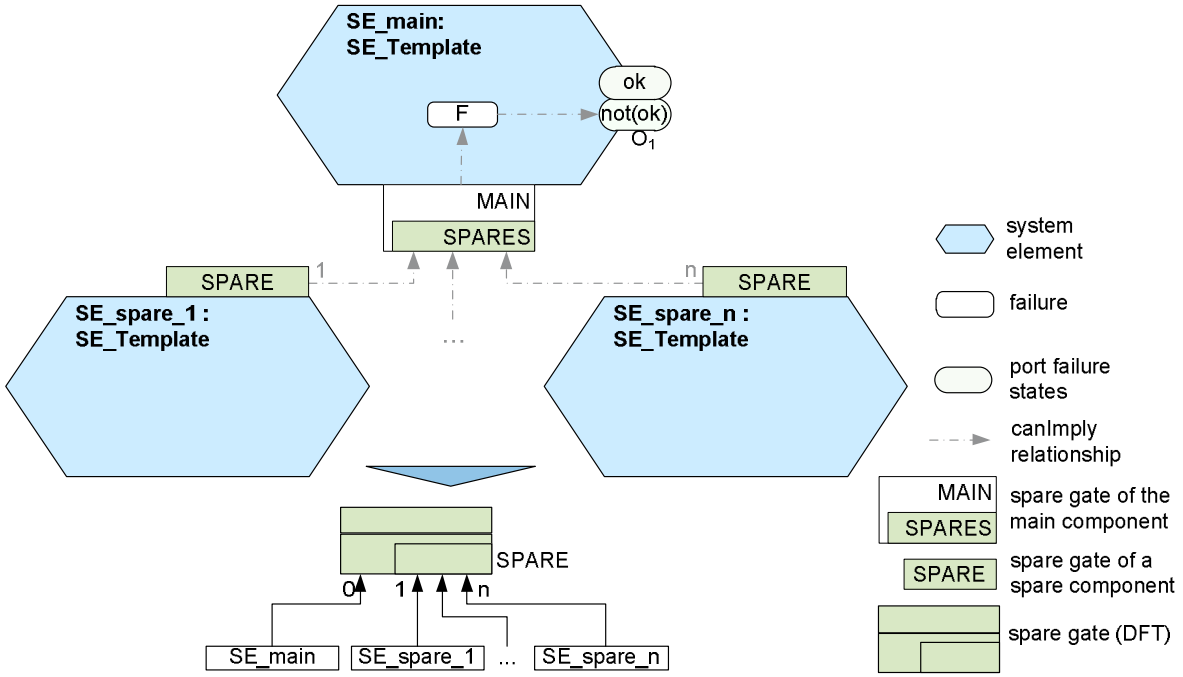

Figure 6. Specification of a SPARE gate within the principle solution (cut-out) and the corresponding dynamic fault tree

The sequence enforcing (SEQ) gate can also be modeled. It has a triggering failure T and a number of outgoing failures, which are connected to the gate by ordered edges. When the triggering failure T occurs, all outgoing failures are forced to occur in the particular order defined by the ordering of the edges. The graphical representation of the SEQ gate in the specification of the failure propagation resembles the graphical representation of the FDEP gate; the edges connecting outgoing failures are ordered. SEQ gate can be modeled as a special case of a cold spare gate. For newly incorporated gates rules have been defined, which allow to map the specification of the failure propagation onto a Dynamic Fault Tree; these rules are depicted in Figure 5 and Figure 6.

Another extension of the method for the description of failure propagation of an advanced mechatronic system is to allow the specification of multi-state inputs and outputs. Previously, each input and output was specified as a state tuple (usually called *ok* and *not(ok)*). A great number of applications however, requires more than two states to be modelled (e.g. *ok, fail-safe*, *fail-danger*). Therefore this limitation has been lifted.

7. Case study: The Spring and tilt module of the RailCab system

The specification of the principle solution of the spring and tilt module has been extended with the description of the failure propagation using the presented method (Figure 7). In particular, the FDEP, PAND and SPARE gates were used. The FDEP gate was used to model the Common Cause Failure, which occurs due to strong vibration caused by a highly harsh environment. This Common Cause Failure occurs within the *vehicle body* (F_1) and is propagated to other system elements including the *servo-cylinder* of the spring and tilt module. The propagation continues through the *servo-cylinder*. In particular, the occurrence of F_1 triggers the failure of *hydraulic valve regulation* (F_2) and *servo-cylinder* (F_3). Both lead to other failure effects within the respective system elements. In particular, the failure effect *4/4-way valve hydraulic valve regulation provides no switch position for the 4/4-way valve* may occur (state *not(ok)* of the input *WV1*) and is then propagated to the *4/4-way valve* system element. The PAND gate was used to model the dependency between the faulty input of the *4/4-way valve* system element *hydraulic valve regulation provides no switch position for the 4/4-way valve* (represented by the state *not(ok)* of the input *WV1*) and the internal failure F_5 (*valve slider stays in its current position*). It has been modelled, that if the faulty input occurs first and the internal failure F_5

follows, than the occurrence of the failure *valve does not change the pressure on the output anymore* is the consequence (faulty outputs *WV2* and *WV3* of the *4/4-way valve*).

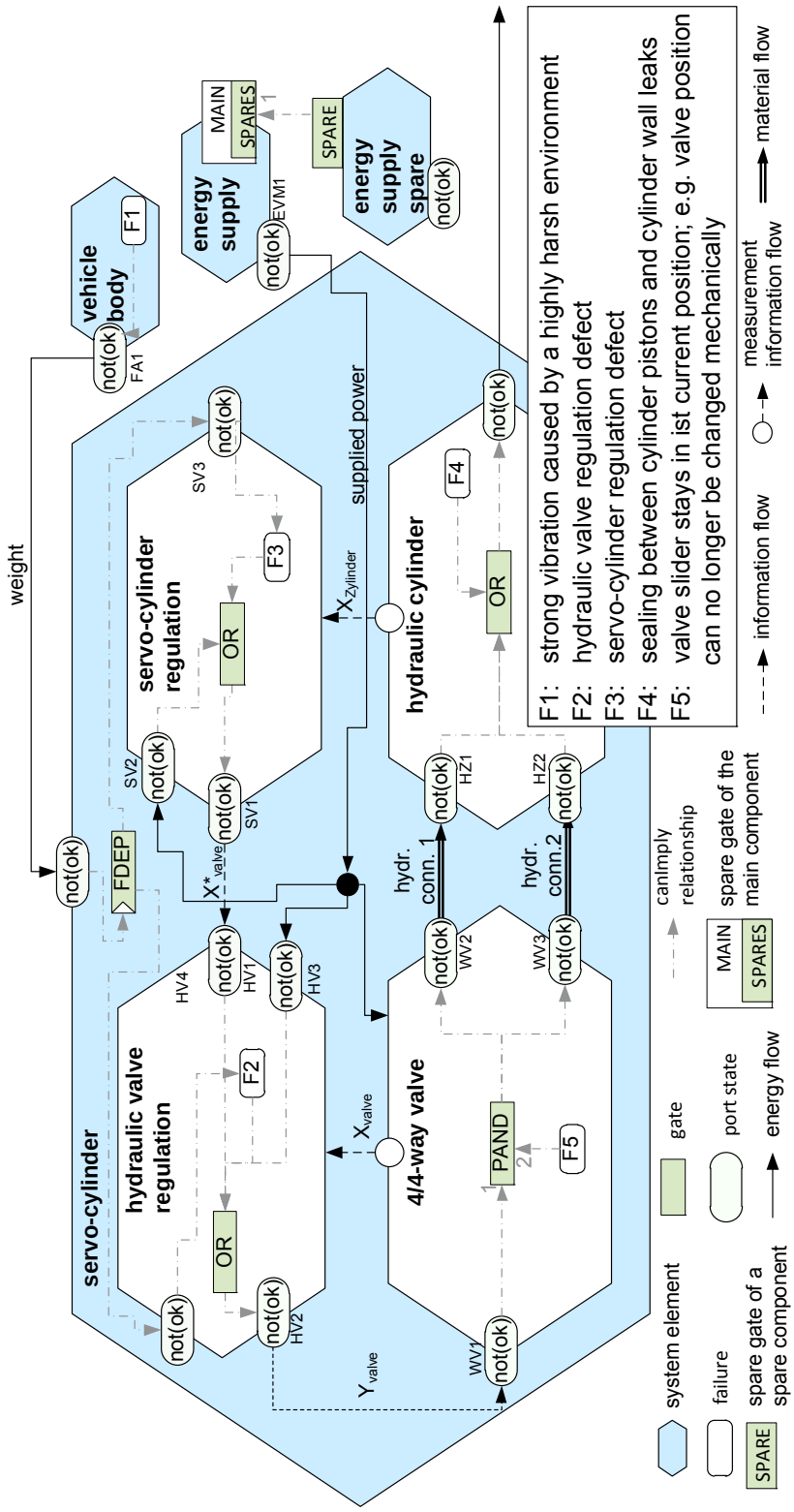

Figure 7. Specification of failure propagation within the partial model active structure (cut-out)

Furthermore the spare system element *energy supply spare* was modeled; the corresponding main system element is *energy supply*. When the main *energy supply* fails, it can be exchanged by the respective spare; the availability time of the system is increased.
With the presented methods failure propagation paths are modelled and analysed that cross domain boundaries, e.g. between system elements from hydraulic, electric engineering and mechanical engineering. For analysis purposes the description of the failure propagation is mapped onto a Dynamic Fault Tree. Established analyses from the DFT field are then used, which were introduced in Section 4 (e.g. [Codetta-Raiteri 2005], [Codetta-Raiteri et al. 2010]). Using the presented method first statements regarding reliability and availability (especially with regard to spare allocation) are made in the early engineering phase of conceptual design. Based on the analysis results, optimization measures can be planned and implemented. A great number of costly and time-intensive iteration loops is avoided, which would otherwise occur during the system integration.

8. Concluding remarks and outlook

In the contribution a method for the description of the failure propagation of an advanced mechatronic systems within it principle solution was introduced. The integration with the specification technique CONSENS enables the domain-spanning use of the method in the early development phase of conceptual design. The interdisciplinary dependencies and cross-domain failure propagation paths are taken into account. The state-of-the-research approach of Dynamic Fault Trees has been adapted and incorporated into our method. Modelling of functional dependencies, sequence dependencies as well as spare allocation is now possible. The method allows first principal statements with respect to reliability as well as to other aspects such as availability at an early development stage. Counter and detections measures are derived and implemented at an early development stage. Some costly and time-intensive iteration loops are avoided, which would otherweise occur during the integration of the contributions of the involved domains in the further development phases.
Future work will address Dynamic Bayesian Network driven analyses of the model (cp. [Codetta-Raiteri et al. 2010]) as well as the incorporation of the description of failure behaviour by means of failure rates and probability distribution functions. Moreover, an adequate software support is being developed.

Acknowledgement

This contribution was developed in the course of the Collaborative Research Centre 614 “Self-Optimizing Concepts and Structures in Mechanical Engineering” funded by the German Research Foundation (DFG).

References

Birolini, A., “Reliability Engineering. Theory and Practice”, 5th ed., Springer-Verlag Berlin Heidelberg, 2007.

Boudali, H., Crouzen, P. Stoelinga, M., “A Rigorous, Compositional, and Extensible Framework for Dynamic Fault Tree Analysis”, IEEE Transactions on Dependable and Secure Computing, Vol. 7, No. 2, 2010, pp. 128–143.

Codetta-Raiteri, D., “The Conversion of Dynamic Fault Trees to Stochastic Petri Nets, as a case of Graph Transformation”, Electronic Notes in Theoretical Computer Science, Vol. 127, No. 2, 2005, pp. 45–60.

Codetta-Raiteri, D., Bobbio, A., Montani, S., Portinale, L., “A dynamic Bayesian network based framework to evaluate cascading effects in a power grid”, Engineering Applications of Artificial Intelligence, 2010.

Deyter, S., Gausemeier, J., Kaiser, L., Poeschl, M., “Modeling and Analyzing Fault-Tolerant Mechatronic Systems”, Proceedings of the 17th International Conference on Engineering Design (ICED'09), 2009, pp. 55–66.

Dugan, J.B., Bavuso, S.J., Boyd, M.A., “Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems”, IEEE Transactions on Reliability, Vol. 41, No. 3, 1992, pp. 363-377.

Friedenthal, S., Steiner, R., Moore, A. C., “Practical Guide to SysML: The Systems Modeling Language”, Elsevier Science, 2008.

Gausemeier, J., Frank, U., Donoth, J., Kahl.S., “Specification technique for the description of self-optimizing mechatronic systems”, Research in Engineering Design, Vol. 20, No. 4, 2009, pp 201-223.

Kaiser, B., Liggesmeyer, P., Mäckel, O., "A new component concept for fault trees", Proceedings of the 8th Australian workshop on Safety critical systems and software, Vol. 33, Australian Computer Society, 2003, pp. 37–46.
RailCab – Neue Bahntechnik Paderborn: the project web site [online]. Available from: http://railcab.de/ [Accessed 11 October 2011].

M.Sc. Rafał Dorociak
Prof. Dr.-Ing. Juergen Gausemeier
Heinz Nixdorf Institute, University of Paderborn,
Fuerstenallee 11, 33102 Paderborn, Germany
Telephone: +49-5251-606261 | -6267
Telefax: +49-5251-606268
Email: Rafal.Dorociak@hni.upb.de | Juergen.Gausemeier@hni.upb.de
URL: http://wwwhni.upb.de/en/pe