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

DESIGN 2012

A NEW INTEGRATED APPROACH TO IMPROVE THE ROBUSTNESS OF COMPLEX MULTI-DISCIPLINARY SYSTEMS

T. Sop Njindam and K. Paetzold

Keywords: systems engineering, multidisciplinary systems, failure analysis

1. Introduction

As technology evolves, the integration of information processing and other software components into our daily products is becoming common place, turning them into multidisciplinary systems. We define multidisciplinary systems as systems that consist of physical components and exhibit autonomous behaviour by sensing their environment, being equipped with sensors, and by affecting or acting without human guidance in their environment. These technological advances probably impact the susceptibility of these products to fail - the majority of users claiming that product robustness has decreased over the years - which might be justified by the several recent product recalls. Although we can observe, ever since the 1950s, an increased awareness towards the perception of the importance of safety and reliability for aerospace and nuclear applications, the same can hardly be said about multidisciplinary systems. Till now, redundancy, that comes along with an increased weight and highly robust components, has been applied as universal solution to robustness issues, thus shifting the focus to physical or hardware components. "This assumption that system failures are only caused by components failures is no longer true for today´s systems" [Leveson 2011]. They do not fail solely due to mechanical or electronic failures but also due to the software controlling their behaviour. However, to come back to the robustness issues, the fundamental question relates to how to design multidisciplinary products and adjust hardware and software components, so that the system operates satisfactorily with a minimal susceptibility to degrade and stop to perform its required tasks. Thus, we simply limit ourselves to the robustness linked with the system ability to achieve its required tasks, that is to say a minimal susceptibility not to fail, despite the occurrence of potential harmful external and internal factors. Improving the robustness of complex multidisciplinary systems means, in other words, to reduce the impacts and effects of these external and internal influences on the system performance as whole.

This paper presents a new approach on how we can improve the robustness of complex multidisciplinary software-driven systems by using system engineering concepts throughout the design procedure. It is structured as follows: section 2 will introduce the variety of failures to occur in multidisciplinary systems. Section 3 will present the state of the art in failure analysis of complex multidisciplinary systems in terms of the current gaps to fill. Then after having reviewed it, we will introduce our integrated approach (section 4), based on specific failure criteria to be defined and which cover system viewpoints such as requirements, structure, functions, behaviour. In section 5, an indoor mobile robot will be used to demonstrate the way in which our integrated method can efficiently fix these failures to improve product robustness. The final section will give a summary of the proposed approach and highlight proposals for future work.

2. Multidisciplinary systems failures

The introduction of new technology in product design might bring with it many unknowns and involves as a result increased risks to fail which must be taken into account from the earliest design stages. Systems fail generally either due to weak functional description not complying with the initial specifications, because of deviation from the correct service caused by bugs in the software yielding their behaviour, or because of physical failures related to hardware components. An attempt to briefly determine the possible causes or consequences of failures software-hardware systems has been made in [Avizienis 2000] with a differentiation between the often confused terms faults, errors and failures. According to [Birolini 2010] and in line with accepted standards, " a failure occurs when a product or an item stops to perform its required function". A fault is defined in IEC 50(191) as " the state of an item charecterized by inability to perform a required function". The term error is referred to as the deviation from an expected theretically correct value.

However, for complex multidisciplinary systems, the causes of faults, errors and failures are so many, complex, often interdependent and vary with products so that listing or describing them would go beyond the scope of this paper. We can instead consider specific scenarios, which may lead to the system inability to perform its required tasks, at their level of occurrence in the system lifecycle (see figure 1) with the aim of fixing them or to reducing their impact on the system performance.

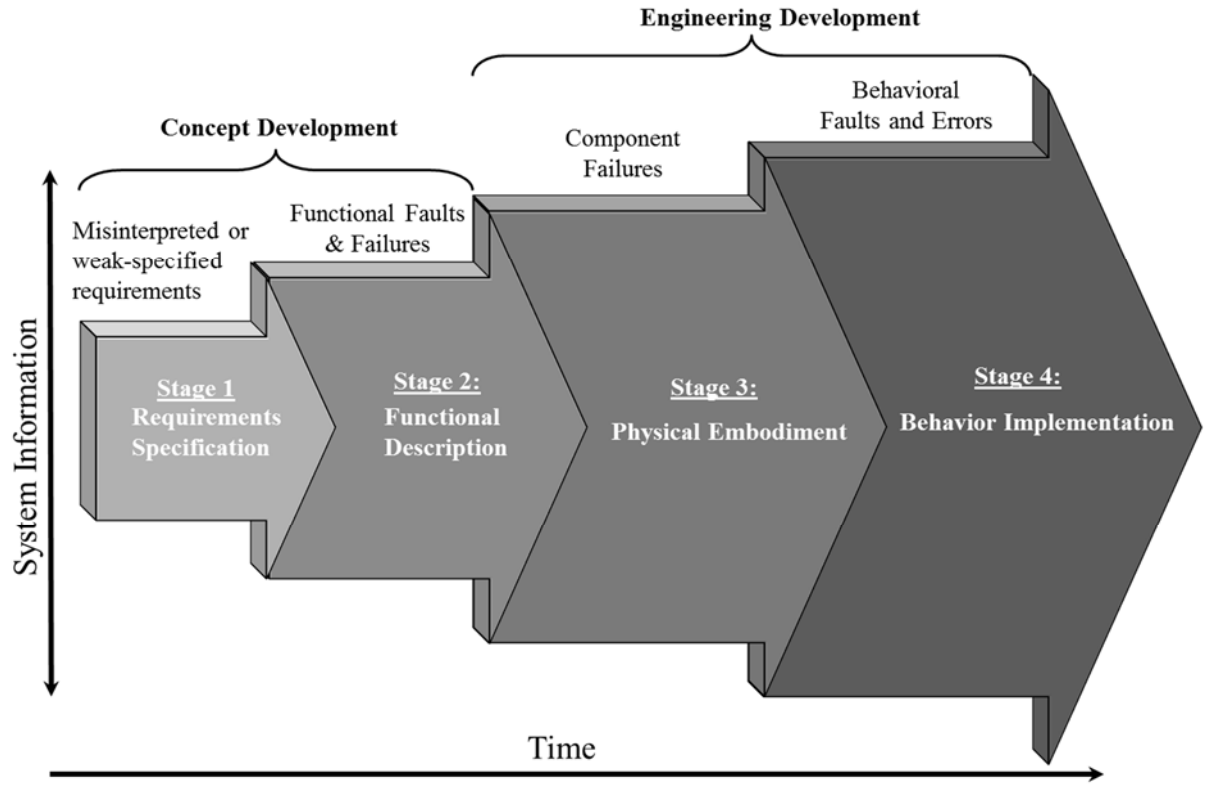

Figure 1. Development stages of the system life cycle and levels of occurrence system failures

The key to develop robust products and reduce their susceptibility to fail as whole is the identification of the factors which might lead to a faulty state. In our opinion, causes of faults, errors and failures can occur in every activity within the system lifecycle from the requirements specification and functional description to the system operational stage. It is therefore important to consider, right from the beginning, the system in its entirety rather than just focussing on specific properties.

At the earliest stages of the design process, typically in the concept definition phase, tasks the system is supposed to perform are usually specified by answering questions like how well and under what conditions the system must perform which task. Further classification and weighting of requirements proceeds with the categorization in functional, technical, customer, non-functional requirements. Dozen of system failures, up to 40%, arise due to weak requirements specification at this stage.

According to [Pahl 2006], the functions the system is supposed to perform are typically represented within the design process as the relationship between the system input and output and, generally speaking, seen as a sequence of actions to be performed to achieve the overall function. Causes that may lead to the termination of the system function are associated at this level of consideration with the functional elements (signal, data , material and energy elements) and with the product specific logical sequence of actions to perform a required task.
At the next stage, a major step which occurs towards the implementation of the derived product functions within the design procedure is the embodiment into structural components. Faulty states at the structural level are the most common and broadly linked to mechanical, electrical and electronic components. It is necessary from a system engineering point of view to early visualize this physical embodiement process of the functional configuration, even though it occurs at the latest design stages, in order to minimize the occurrence of system failure due to physical failures.
Behavioral faults and errors are generally misassigned to physical failures. This may not be true for multidisciplinary systems since their behaviour is yielded not only by the structural but also by the software components. In concrete terms, faulty states occurring at the behavioural level are to be related with the deviation from the system variables affecting its dynamics, its internal state and self-monitoring over time.

3. Problem definition

Complex multidisciplinary systems are usually handled during their development through the application of many basic activities by large teams of engineers or departments according to their expertise [Kossiakoff 2003]. The successive application of these activities should go in line with systems engineering concepts, as illustrated in Figure 2, and includes:

- Requirements Specification
- Functional Analysis
- Physical Analysis
- Behavioral Analysis

Moreover, failure analysis methods are traditionally specifically conducted at each of these design, each design stage building upon available data and gained insights to serve as a basis for the next stage. Thereafter, an overall integration testing, which consists of the testing of the software components on the hardware plattform, is often performed at the last stages to validate the system as whole. Accordingly, many weak spots, not revealed during the single stages, are not revealed until the late validation stage is done for multiple reasons that are considered, in our view, as weaknesses in current failure analysis applications in multidisciplinary systems:

- System details as well as subsystem dependencies, as referred in various publications in systems engineering, might be overlooked.
- A similar reason concerns the lack of continuity and consistency of failure analysis activities throughout the whole design procedure, which actually focus explicitely on physical components and downplay other behavioural aspects that impact the system as well (see Figure 2).

Since tools, data, modeling paradigmas, and system viewpoints usually differ from stages to stages, it often remains unclear how the result of the failure analysis from a previous stage impacts the subsequent analysis in the following stage. As an example, a wrong understanding of what the system should do and how it must do it, would probably lead not only to inappropriate functional models of the system but would also fail to address reliability issues right from the start. In a similar manner, the link between functions and components for the physical embodiement would be helpful to set out which component failure leads to which functional failure. The same can be said with the linking of physical and functional failures to behavioural faults and errors with the objective to clearly and concisely determine potential causes of the termination of the system function.

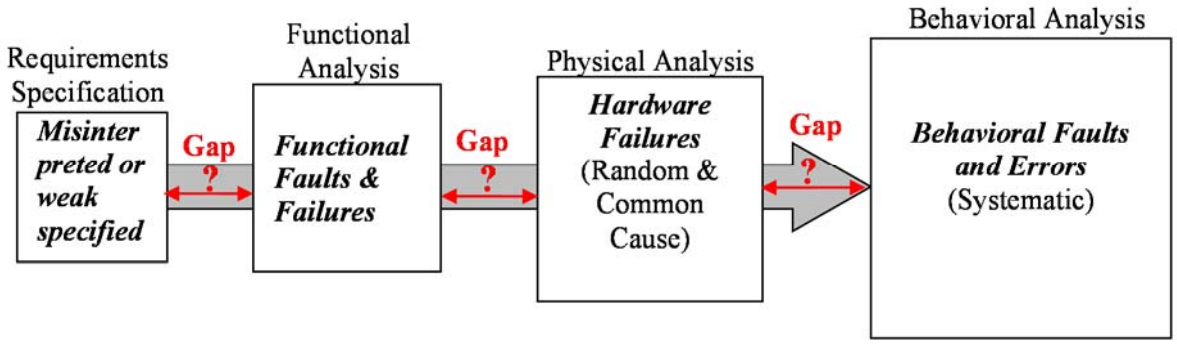

Figure 2. Gaps in state-of-the-art approach for failure analysis of multidisciplinary systems

4. Integrated approach

Until very recently, the tendency since the proliferation of multidisciplinary software-driven systems such as mechatronic systems has been to use system subdivision in hardware and software parts as a means to undergo separate failure analysis and validation once the system architecture is done [Smith 2008]. However, the need to analyze system failures as a whole and right from the start by means of analysing their robustness and estimating system reliability has been recognized in recent works. The hurdle was to bring hardware and software reliability data together, which is to the best of our knowledge, still not realized. In essence, hardware and software parts are different in nature. Whilst hardware parts are seen as predictable and quantifiable in terms of their failure rate, software faults are quite the contrary neither predictable nor quantifiable, at least in the conceptual design stage. In addition, several software reliability growth models have been developed to cope with the complex issue of early assessment of the quality of software, but still remain not standardized and often require data from the late testing stages such as lines of code, testing time and operating conditions [Lyu 2007].

Our approach does not reinvent the wheel by formulating new reliability models for software or for multidisciplinary software-driven systems. A number of techniques has already been proposed in the literature to attack this problem even if they still give unsatisfactory predictions [Lyu 2007]. It is rather our intention to limit ourselves within the scope of this paper to the detection and fixation of factors affecting the system ability to perform its required tasks and focus more closely on its overall behavior. Hence, we propose an approach that complies with system engineering concepts and the close connection between system aspects such as its requirements, functions, structure, behaviour and the information related to these viewpoints so that the coherence and the functionality of the system can be maintained as a whole.

The basic hypothesis of the approach (see Figure 3) we present to analyze the failure behaviour of complex multidisciplinary systems is based on the integrated modeling architecture the research community is claiming for the realization of next generation of complex systems, since no single approach can capture and describe all system aspects. The objective is to detect and fix design flaws, that is to say, functional, structure and behavioural faults, errors and failures according to previously defined stakeholders and technical constraints. However, one of the difficulties with multidisciplinary systems, as illustrated in Figure 2, is the integration and forwarding of system information related to its core aspects throughout the design procedure. This obstacle is overcome by a first separate failure analysis at each of the levels of occurrence and then further linking between them (see Figure 3).

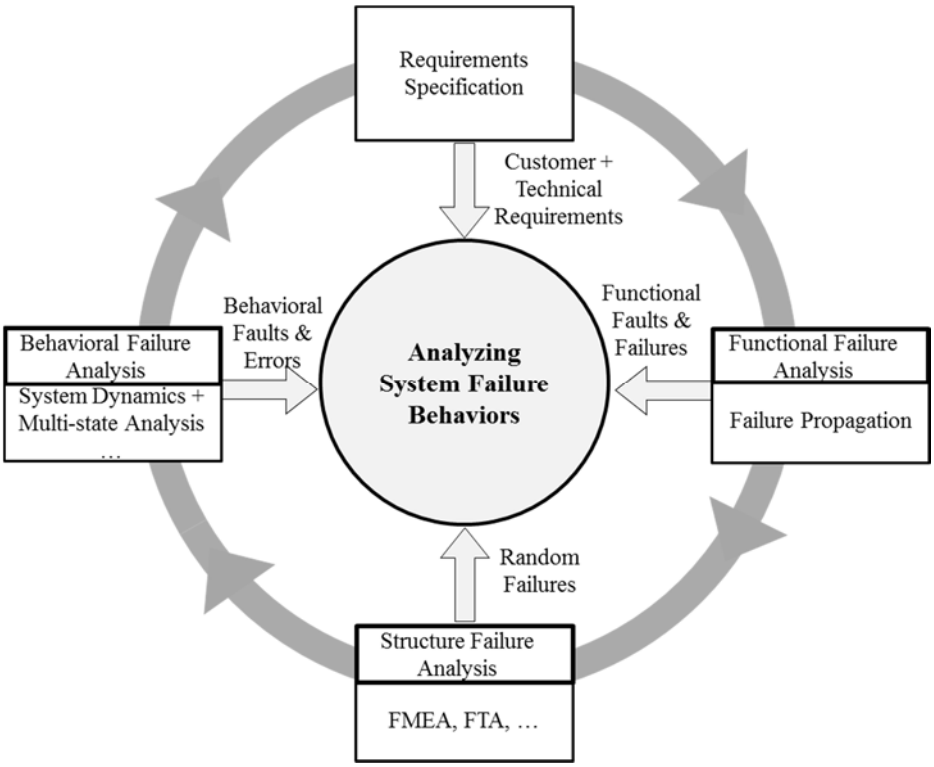

Figure 3. Integrated systems-based approach to the robusteness of complex multidisciplinary systems

The abstraction of functional models makes it possible to leave first inter-domain barriers out of consideration and to gain concrete insights into the system functionality by describing functional principles of the system. Required functions in various application cases can then be sketched and potential interactions can be investigated by means of the impact of the failure of one function on the others, of how it propagates, and of the impact of the failure of one of the system information flows on the system. Furthermore, first considerations about the insertion of safety functions can be undertaken to ensure a proper user-device interaction.

As the system architecture is usually generated once the functional model of the system is completed, traditional methods such as System-FMEA (Failure Modes and Effects Analysis) or FTA (Fault Tree Analysis) make use of the system architecture to to address robustness issues. At this stage, the system is decomposed into manageable subsystems, which are subsequently decomposed in a hierarchical way into further elements or components. By restricting ourselves to the functional-structural look at this stage, we can find out which component failure is likely to lead to which functional fault or failure and decide depending on the requirements whether redundancies should be introduced.

Upon further reflection, the system architecture and its behaviour are tied coupled. Extracting the system behaviour requires additionally, as afore mentioned, an analysis of its internal states in terms of what factors may lead to the system degradation as well as of the acceptable limits of its dynamic performance based upon its analytical models. At this stage, the task is to set the decisive performance parameters governing the system behavior within specific bounds and to define critical-feared states in relation with the failure of safety critical components. Critical feared states are hereby defined as states, in which the probability of system failure is higher than in other states. We might then define error margin, that might lead to performance degradation and then generate discrete-time Markov models whose transition will be coupled with probabilities.

5. Application example

We demonstrate the usefulness and applicability of our method with a classical household autonomous robot. We assume the robot to be a classical indoor robot, capable of making tight turns in its environment, while moving objects from initial random positions to goal positions. On the most basic

level, robots typically sense their surrounding environment and receive as result information about it. Based on this information, they plan their actions according the control laws yielding their behaviour and move towards the target while avoiding possible encountered obstacles to act on the environment.
As we already mentioned in the previous section, causes leading to faulty states of such complex multidisciplinary systems should be addresses right from the requirements specification. Thus, after having taken in consideration the stakeholders needs as well as the task the robot is supposed to perform, we were able to gain some insight into the system functionality in terms of the basic abstract representation of the system functions. As illustrated in Figure 4, the robot has four main functions:

- to sense the world and other physical objects
- plan actions
- to move in an indoor environment
- to move objects.

The functional model provides also valuable information on interaction between the system functions by means of energy, signal and material flows on what basis first investigations on how the system fail can be undertaken. An effective method to reveal sources of failures is to negate system functions [Bertsche 2008]. Several other methods such as Functional Hazard Assessment (FHA) or Hazard of Operability Studies (HAZOP) are also suitable for this purpose. Our aim is nevertheless to link functional faults and failures with component failures in accordance with the functional-structure-mapping process. Further, Fig. 4 shows the failure of the function *move in an indoor environment* and the potential affected components namely the electric motor, the drive transmission and wheels. An outcome of this early linking between functional and components failures may consider the insertion of redundant components. However, it remains very ineffective to confine our attention to this level, as the system behaviour has still not be taken into consideration. Moreover, as the system prototype has not yet been built, information about the system physical struture can realistically only be available in functions-components catalogues, unless the design is a variant of an existing one.

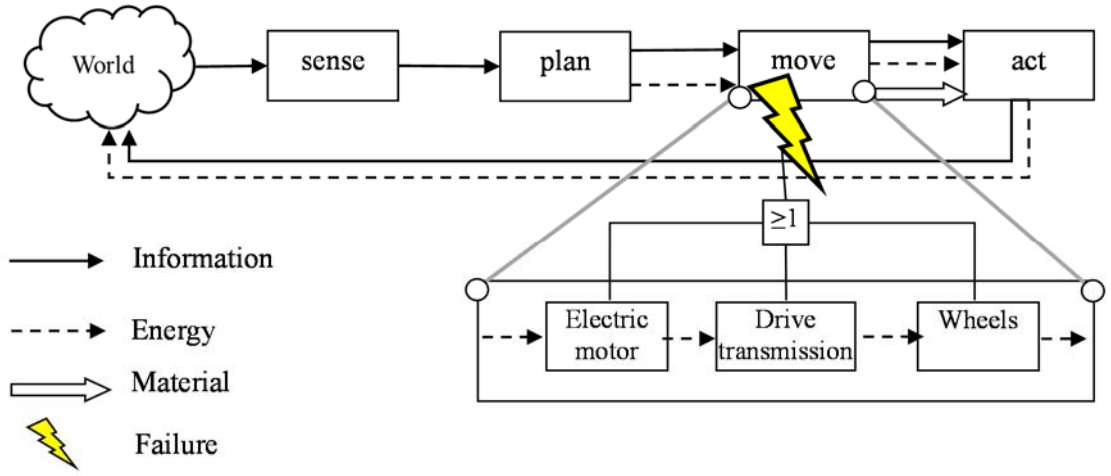

Figure 4. Link between functional and component failures

The final issue of this paper is to link behavioural faults and errors with component failures. The system behaviour is herein defined as the set of actions, assigned with inputs and outputs, the robot shall perform to fulfil its tasks (see Figure 5). Within this framework, action outputs can be analysed with dynamic system models based on equations such as Newton´s law of motion, or fundamental motor formulas. Two possibilities lead to behavioural faults and errors in this case: either the system does not perform the actions as predefined or the output values deviate by far from the expected ones. We will focus on the later case within the bounds of this paper.
Figure 5 illustrates an example of a robot behavioural failure which occurred while moving forward and which is due to the significant difference between the expected and the actual motion speed. A faulty state during the forward moving action is logically very likely to be linked with the movable physical structure of the robot that includes the electric motor, the drive transmission and the wheels.

We could then vary design parameters characterizing the dynamic model of the motion speed to obtain threshold values so that the speed motion should not fall below the minimum value of the expected speed motion margin. This error margin specification might be of practical importance with view to an early analysis of the impact of behavioural errors on the system behaviour.
Another attempt to combine the system behaviour with components, based on our previous work in [Sop Njindam 2011], is to make an analysis based on the multi-state model by assigning the failure of specific components with system internal states, that is to say, as also shown in Figure 5:

- an initial system state with operational components Z0
- the failure of one out of two speed sensors that leads to the state Z1
- the failure of the transmission drive that leads to the state Z2
- the failure of the electrical motor that corresponds to a critical state Z3.
- the transitions between the states that correspond to failure and repair rates of the corresponding failed components.

From this point on,we can then randomly vary the failure and repair rates with the aim of investigating when critical feared states can be reached.

6. Conclusion and future work

A major hurdle in assessing failure analysis issues on complex multidisciplinary systems is on the one hand the strict system subdivision in hardware and software parts, each of them undergoing a separate failure analysis and on the other hand the late integration of hardware and software parts during which unexpected weak spots can occur. The approach we take follows the simple assumption that detecting and fixing factors having a negative influence on the system ability to perform its required tasks would increase its robustness. It relies on systems engineering standards by focusing the system as a whole and by linking the system viewpoints requirements, function, structure and behaviour to each other. As faults, errors and failures sources can be traced back at anytime during the design procedure, we believe that it is our concern to identify their causes and to fix them by adapting the system design in order to meet the stakeholders requirements. Using the integrated approach we described in this paper, failure analysis investigations can be first conducted at the functional and at the component level with a view to the relevance of system functions, components and to the system whole functionality. One way of improving the product design after such an analysis would be to insert redundant components at the right place. Additionally, the system behavioural characteristics such as its internal states, the set of actions it must perform to fulfill its tasks and its dynamics can be explored after having defined failure criteria such as the reaching of a critical state or the exceeding of an error margin of one of its performance characteristics. The system design can then be improved and result either in a tradeoff between design parameters by setting an appropriate values range for the system functionality being striven for or in the insertion of fail-safe states as necessary.
There is much exciting work to be done as we believe that the design process should be enhanced by an emphasis on how the system might fail as a whole. As afore mentioned, the validation procedures can be implemented at the different stages: functional, structural, behavioural and consequently related to each other. In assessing behavioural failure analysis, we should anyway be aware that failure criteria and scenarios should have been previously and properly defined. Although it is nearly impossible to define all failure scenarios and system internal states in a dynamic environment, we can still improve the quality of our products by detecting and fixing design flaws that may cause the system fail with the objective of minimizing the number of product recalls.

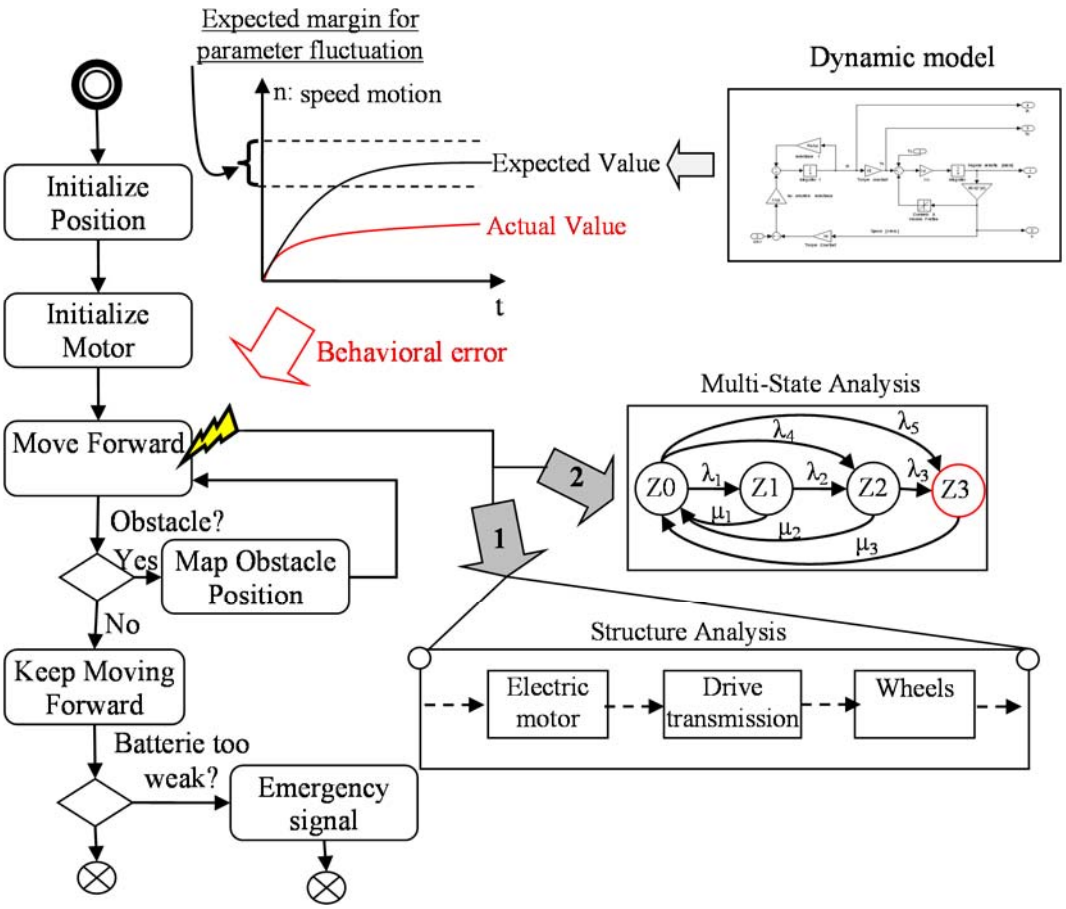

Figure 5. Link between behavioural faults, errors and component failures

References

Avizienis, A., Laprie, J. C., Randell, B., "Fundamental Concepts of Dependability", In Proceedings of the 3rd IEEE Information Survivability Workshop, Boston, Massaschusetts, USA, October 22-24, 2000, pp24-26

Bertsche, B. et. al., " Zuverlässigkeit mechatronischer Systeme: Grundlagen und Bewertung in frühen Entwicklungsphasen", Springer, 2008

Birolini, A. "Reliability Engineering: Theory and Practice", 6th Edition, Springer Verlag, Zürich, 2010

IEC 50(191), International Electrotechnical Vocabulary (IEV), Chapter 191 – Dependability and quality of service, International Electrotechnical Commission, Geneva, 1990

Kossiakoff, A., Sweet,W. N., "Systems Engineering: Principles and Practice", John Wiley and Sons, Inc., 2003

Leveson, N."Engineering a Safer World: Systems Thinking Applied to Safety", 2011

Lyu, M.," Software Reliability Engineering: A Roadmap" International Conference on Software Engineering 2007, pp. 153-170.

Pahl, G. et. al., " Konstruktionslehre", 7th Edition, Springer, 2006

Sop Njindam, T., Paetzold, K., "Design for Reliability: An Event- and Function-Based Framework for Failure Behavior Analysis in the Conceptual Design of Cognitive Products", Auf: International Conference on Engineering Design (ICED11), Kopenhagen, 2011

Dipl.Ing.Thierry Sop Njindam,
Universität der Bundeswehr München, Institute of Technical Product Developement
Werner-Heisenberg-Weg 39, D-85577 Neubiberg, Germany
Telephone: +49-(0)89-6004-2821 or -2814
Telefax: +49-(0)89-6004-2815
Email: thierry.sop@unibw.de
URL: http://www.unibw.de/lrt3