Integrating Failure Analysis into the Conceptual Design of Cognitive Products: Towards a New Paradigm

Thierry Sop Njindam[1], Torsten Metzler[2], Kristin Paetzold[1] and Kristina Shea[2]

[1] *Institute of Technical Product Development*
Universität der Bundeswehr München
[2] *Virtual Product Development Group*
Technische Universität München

Cognitive products integrate cognitive functionalities such as perceiving the environment, learning and reasoning from knowledge models that are created through the combination of a mechatronic system and advanced software algorithms. While the area of cognitive products is still in his infancy, we regard the safe and reliable performance fulfillment as one of the challenging tasks for the research community to develop cognitive products that meet customer expectations. This paper presents how failure analysis can be integrated in the functional modeling process of cognitive products to increase their safety. For this purpose, we explores if current state failure analysis methods are appropriate for analyzing cognitive products and where their weaknesses and strengths are. The cognitive coffee waiter is used as an illustrative example to concretely show the limitations of these methods.

1 Introduction

The effectiveness of failure analysis methods in engineering design is decreasing due to more complex products. By taking a look at modern engineering products, more than four decades have passed now since classical mechatronic products have been introduced for the first time. Harashima et al. defined in [8] the term mechatronics as "the synergetic integration of mechanical engineering with electronics and computer control in the design and manufacturing of industrial products and processes." Since then, the requirements of some of these products related to man-machine interaction and degree of autonomy have changed, thus increasing the amount of software and bringing other considerations to the foreground such as user interaction, environment considerations, product behavior, emergent properties etc. To date, products are called smart, adaptive, intelligent or cognitive. Cognitive products possess cognitive functions i.e. to learn, to perceive or to reason according to the integrated logic, flexible control loops and cognitive algorithms. This flexible behavior addresses users´ needs and desires better than mechatronic products do, but comes along with an increased susceptibility to failures and errors. Birolini stated in [6] that a "failure occurs when the item or the product stops to perform its required function". Performing failure analysis at early design stages refers then to detecting possible failures, to finding the cause (es) of the defects and to eliminating them as early as possible, thus, reducing iterations in the development process of safer and more reliable products.

Holding in view these considerations, it is our belief that cognitive technical systems or cognitive products, with regards to their self-sensing capabilities, high degree of autonomy, emerging behavior and increased complexity might be, to the contrary of what has been claimed by many researchers, rather less reliable and robust. Hence, failure analysis is more important today and in the future than it was in the past since cognitive products are supposed to closely interact with users and operate in a non-predefined environment, but despite that, must perform accurately their tasks at all times.

Failure analysis is not a novel concept at all. Several well-known failure analysis methods, e.g. Failure Modes and Effects Analysis (FMEA), Fault Tree Analysis (FTA), Hazard and Operability Studies (HAZOP) and Functional Hazard Assessment (FHA) were developed in 1960s by military or aerospace agencies mainly because of the strict safety and reliability requirements in these fields. Since then, they have been extended with good success to other fields including the automotive sector, the electronic sector and the mechanical engineering industry. Nevertheless, the German Department of Motor Vehicles recorded an increasing amount of callbacks in the automotive industry during the last years. This might probably be related to the increased integration of those advanced mechatronic systems such as adaptive cruise control (ACC), driving assistance systems, automated or semi-automated parking aids and so on, into today´s vehicles that expand the range of the classical vehicle behavior, thus pushing towards more autonomy. Even though the management of failures resulting from the increasing products complexity has been recognized as a problem by many research institutions, a solution has not been found yet. Till now, redundancy and highly robust components have been applied as universal solution to overcome these hurdles, thus leading complex systems, i.e. the NASA Rover to being very expensive [9]. Furthermore, this approach is not appropriate for mass production of consumer products as required for cognitive products. Redundancy, for example, comes often along with an increase in weight and highly robust components are generally more expensive than standard components. *But what has changed since the last decades in failure analysis methods? Are fundamentals and actual theory behind these methods suitable for these software-intensive products and systems such as cognitive products or do we have to adapt them to the actual context? This paper explores if current state failure analysis methods are appropriate for analyzing cognitive products and where their weaknesses and strengths are. This provides way for the development of new methods supporting the failure analysis of cognitive products that heavily rely on software***.** Due to the limited scope of this paper, we will limit ourselves to the methods for qualitative failure analysis we mentioned earlier. An extension to quantitative methodologies would go beyond the size of this paper.

2 Background

The following section introduces the relevant terms and methods of this paper, namely a basic introduction to cognitive products, their functional modeling and methods for qualitative failure analysis.

2.1 Cognitive Products

"Cognitive products are tangible and durable things with cognitive capabilities that consist of a physical carrier system with embodied mechatronics, electronics, microprocessors and software. The surplus value is created through cognitive capabilities, enabled by flexible control loops and cognitive algorithms" [3]. Cognitive capabilities basically describe the basic functions enabling cognition as a whole, e.g. to learn, to think, to understand and to reason [5]. Cognitive functions enable cognitive products to act more flexible because they do not necessarily obey rigid and pre-defined control algorithms but instead process the perceived data according to the context [5]. Customer needs are satisfied through the intelligent, flexible and robust behavior of cognitive products that meet and exceed the customers´ expectations. Cognitive products have all or a subset of capabilities of Cognitive Technical Systems (CTS) and the solid grounding of an everyday product that meets user needs and desires" [3].

Issues about their safety and reliability in unconstraint or partially known environments in which they interact with humans have to be clarified, especially at early stages of their design to avoid costs related to callbacks and later quality assurance measures, and represent one of the most challenging topics to be mastered before their market launch [18].

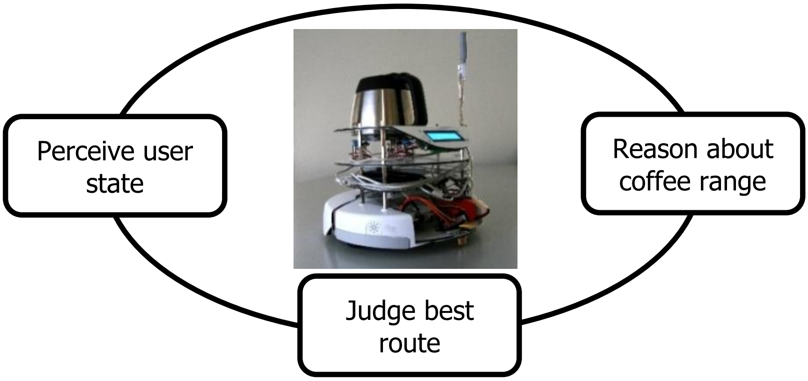

Figure 1: The cognitive coffee waiter "Starbugs"

"The coffee robot waiter (figure 1) is the most developed prototype and students continuously work on developing its cognitive capabilities. It is able to serve coffee based on an order placed on a website. This is enabled since the robot initially learned the environment, using a laser range scanner, and subsequently has a map of the environment. If more than one order is placed, it calculates the optimal route according to an online traveling salesman algorithm and can move autonomously to the target positions. Recently, the amount of coffee in the pot is taken into account for the calculation of a route as well as the previous average coffee consumption of the users. According to its estimation, it might refill coffee before serving customers if it anticipates running out of coffee. On its way, it can avoid obstacles by driving around them and gives the customers feedback about how long they have to wait. This information is also based on a learning algorithm that estimates the time to target based on parameters such as distance, past experience and dynamic obstacle volume" [13].

In the coffee robot waiter project, the software is structured as depicted in figure 2. The architecture is described from bottom to top. A Player server running on the robot hosts modules (called drivers in Player) for accessing the Create platform, laser range scanner and Phidgets. Another Player server runs on the computationally more potent laptop. It hosts proxy modules connecting to the Player server on the robot to forward data and commands, and modules processing the data: A localization module estimates the current robot position based on the robot's odometry data and by comparison of its laser range scans to a previously learnt map of the environment, a local navigation module provides short-range navigation and a global navigation module performs path-planning in the environment map. The actual robot controller is the client program connecting to the player server on the laptop. For the coffee robot waiter, Python has been chosen as the client programming language. It accesses a remote web server to fetch coffee, orders and then uses Player's global navigation module to drive the robot to its targets, as well as directly accesses the Phidgets module for displaying text and reading the coffee pot force sensors. The web server additionally provides the interface for users to place their coffee orders.

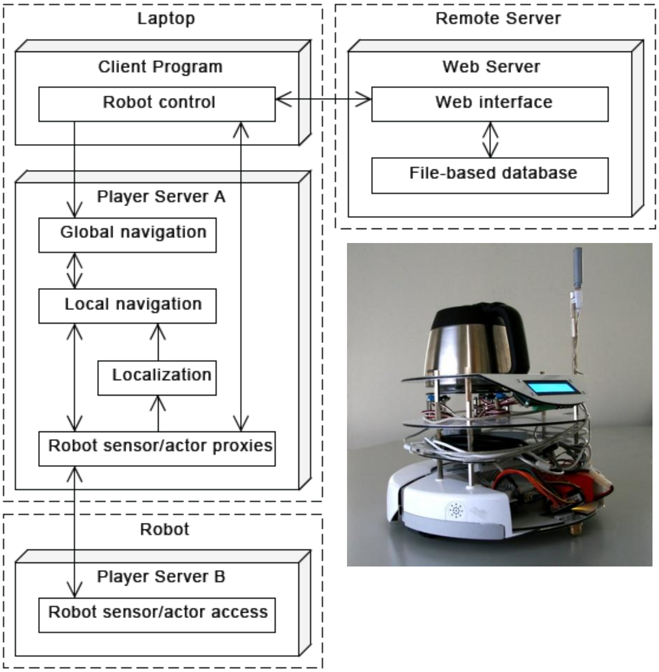

Figure 2: Starbugs software architecture: Programs, modules and information flow.

2.2 Functional Modeling with Cognitive Functions

"The process of describing the product function of a system or product in a model through sub-functions is called functional modeling [12, 13]. This usually takes place in conceptual design after identifying the system or product requirements and before searching for solutions. It is a key step in the product design process for original and redesign [14]. Functional modeling is an abstract but direct method for understanding and representing technical systems considering the product function and all sub-functions of the system or product while also representing their connectivity. It can help designers to better understand complex products [13, 15], e.g. cognitive products and CTSs. Design activities are eased through functional modeling by problem decomposition, physical modeling, product architecting, concept generation and team organization [12, 14, 15]. Flow-oriented function models are appropriate to describe systems or products with flows [12, 14]. Therefore, it is essential to define how different functions can be connected. This is usually done using energy, material and signal flows between functions in order to gain an in-depth understanding of the product´s functionality, especially when it comes to failure analysis. Generally speaking, every product can fail regardless of whether the focus is on hardware or software. Drawing upon functional models that illustrate signal, data and information flows during the product´s operation may help to reflect and gain an overall understanding of the possible failure scenarios at the conceptual stage of the design process.

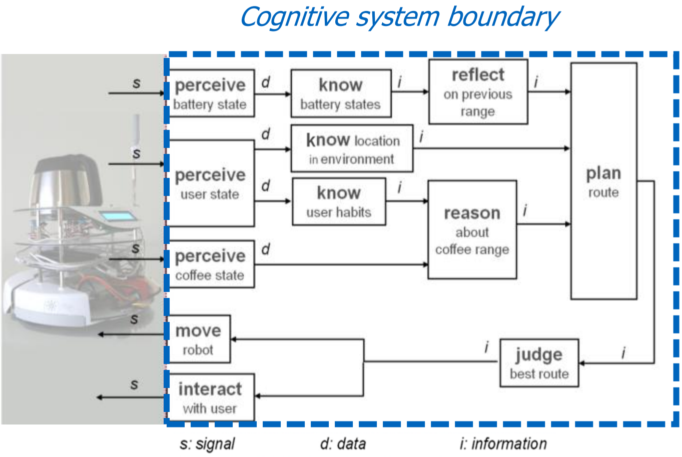

Figure 3: Application example of functional modeling with cognitive functions, according to [5].

Following, the functional modeling of a cognitive product using cognitive functions is explained. Due to the complexity of the whole system the coffee robot waiter is modeled only partially, taking into account functions that are related to the planning part of serving coffee. The result is a clearly arranged function structure, shown in figure 3. During service hours of the robot there is "interaction" between the users and the robot, more precisely the users can place orders on their computers that are transferred through electronic "user signals" to the robot. The robot "perceives user states" including who placed an order, expressed as "user data", and where to deliver coffee to, expressed as "location data". Because the robot has an internal map of its environment it "knows locations in the environment" and can transform the "location data" into "location information", meaning that it knows where to deliver the coffee in its environment. This is the first information necessary to "plan a route" for delivering coffee. Additionally, the robot is able to allocate certain user profiles to "user data" and assign user habits to the "user data". This is possible because every user has to register prior use of the service. The robot "knows user habits" of every user from past events. The result is "user information". Together, "coffee pot data" that comes from "perceiving coffee state" and "user information" enable the robot "to reason about coffee range" according previous coffee consumption of the users in the queue waiting for coffee and current filling level of the coffee pot. As a result "coffee information" is generated and integrated in the route planning. Since the start location for the route is necessary and given by the actual location of the robot, it needs "to perceive the environment", e.g. with a laser range scanner, and compare the "perceived environmental data" with an internal model of the environment. The robot "knows locations in the environment" and compiles "location information" about the current position. "Location information" of the robot itself and users is essential "to plan an optimal route" considering distance and "take account" of all waypoints. In our application example the cognitive function "plan route" is accomplished by applying an online traveling salesman algorithm. The result is "route information" [5].

2.3 Failure Analysis Methods

Failure analysis basically refers to finding the causes as well as the effects of failures in products. The focus is hereby neither to show the applications in the various disciplines or domains nor to differentiate how these methodologies can be used for quantitative of qualitative assessment, but to rely on the main principles of the aforementioned failure analysis methods to show their limitations for the analysis of cognitive products. Figure 4 visualizes the correlation of these failure analysis methods with system details for investigations in the conceptual design stage.

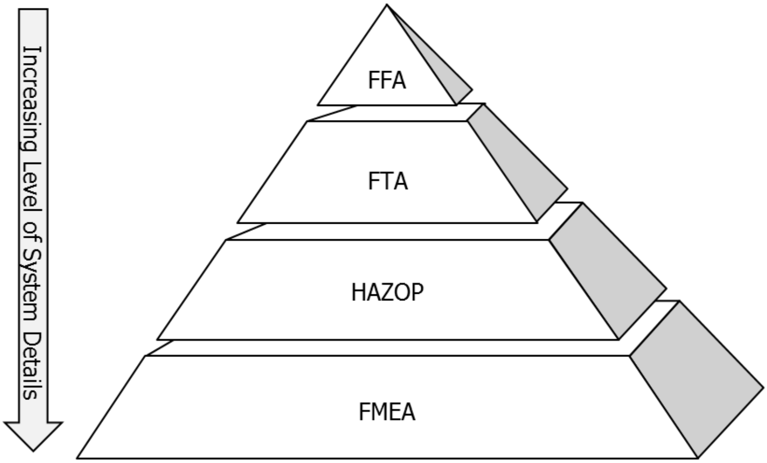

Figure 4: Common Failure Analysis Methods and their Level of Abstraction

While analyzing defects or systematic failures of mechanic or electronic artifacts, several failure analysis methods are available and can be applied. Their aim is mainly to identify and withdraw or fix weaknesses during the product development process. These weaknesses, which most of the time appear in form of failure modes, are not well identifiable for complex multidisciplinary systems, especially at early design stages when the product structure is not yet completely specified. Functional Hazard Assessment (FHA) is especially recommended to be carried out during these early design stages since it can be applied at this low abstraction level, and belongs therefore to the so-called predictive failure analysis methods. In this process (see figure 5), the effects and impacts of the failures of product functions on the system are analyzed. From there on, the estimated functional failures will be assigned to risk factors. Wilkinson et al. claim in [10] the ambiguous formulation of the product functions related to the predefined requirements, especially when it comes to the levels of abstraction. Staying at a too abstract level brings us far from reality and implementation details [10]. Furthermore, functional dependencies as well as interaction with the outside world, which are not unusual in technical artifacts, cannot be addressed with this method. Moreover, the effect of these functional failures on the system can be investigated only in the simplest cases. This method seems therefore not to be adequate for complex multidisciplinary products i.e. cognitive products.

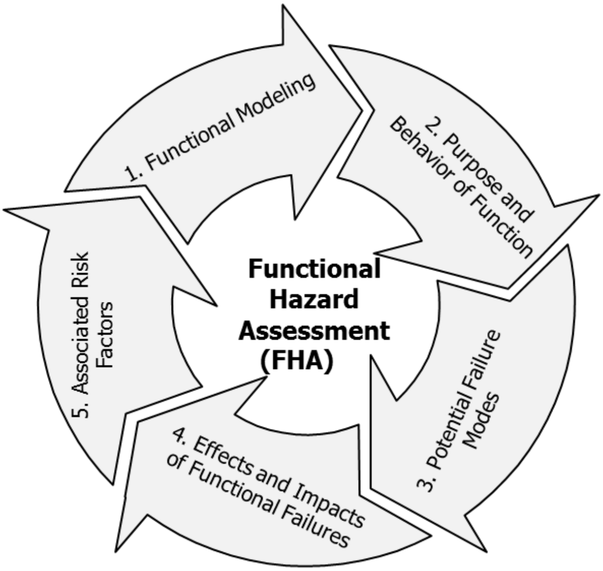
Functional Hazard Assessment (FHA)
1. Functional Modeling
2. Purpose and Behavior of Function
3. Potential Failure Modes
4. Effects and Impacts of Functional Failures
5. Associated Risk Factors

Figure 5: Analysis Steps of the Functional Hazard Assessment (FHA) [10]

Another method whose purpose is to identify potential failures and undertake corrective actions is the Hazard and Operability Study (HAZOP) which has also been developed throughout the 1960s. The constituent steps are defined in fig.6 and reflect the iterative process which is mostly viewed as teamwork. A team of experts is supposed to be led by a moderator and analyzes the system step by step starting with the description of the intended system behavior description or its intended functionality, then using simple keywords such as (more, less, both, different from) to analyze the impact of the system´s behavior deviation in order to undertake corrective actions.

Fault Tree Analysis (FTA) is another widely used method for safety and reliability investigations. FTA is a top-down approach to failure analysis and aims at translating the physical structure of a system into a structural logic diagram. A Top Event, which is considered as a product failure or a product undesirable event is considered as the starting point of this analysis procedure. Furthermore, causes or basic events can then be investigated in the context of the system operation as well as of its environment depending on whether they lead individually or in combination to this Top-Event, thus being connected through logic gates such as AND, OR, etc.

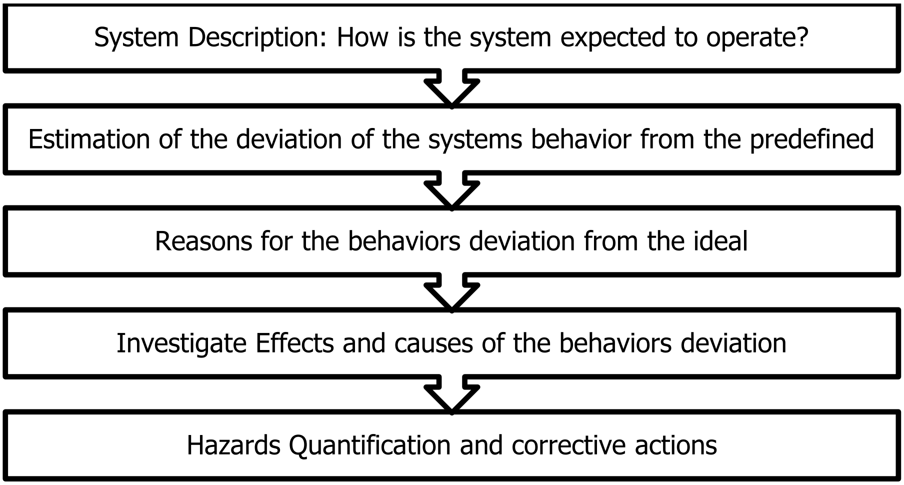
System Description: How is the system expected to operate?
Estimation of the deviation of the systems behavior from the predefined
Reasons for the behaviors deviation from the ideal
Investigate Effects and causes of the behaviors deviation
Hazards Quantification and corrective actions

Figure 6: Hazard and Operability Studies (HAZOP) [19]

.FTA has many advantages, inter alia, its easy to perform methodology and its application in a wide range of areas. With the FTA, it is possible to identify weaknesses of the system at early design stages, to predict some aspects of the system behavior in the sense of events and to undertake corrective actions, thus considering failures of the system as a whole including hardware failures, software errors and human failures. But it cannot consider all system failures, especially for cognitive products whose highly dynamic environment cannot be defined in advance. Moreover, failures, which occur but are not related to the considered Top-Event as well as failures resulting from the emerging behavior of the system, cannot be represented or assessed to be realistic by the specialist [21]. Fault Tree Analysis cannot be checked in most of cases for correctness or consistencies and rely too much on the experience or intuition of analysts, in other words: on their informal understanding on the system to be analyzed. Xiang et al. questioned fundamentally in [20] whether the system´s safety or reliability can be proven after having founded the basic fault events. However, Stamatelatos et al. point out in [21] the qualitative nature of Fault Tree assessments although they can also be used for quantitative purpose, in so far as quantitative data are available or be estimated through probabilistic analysis.

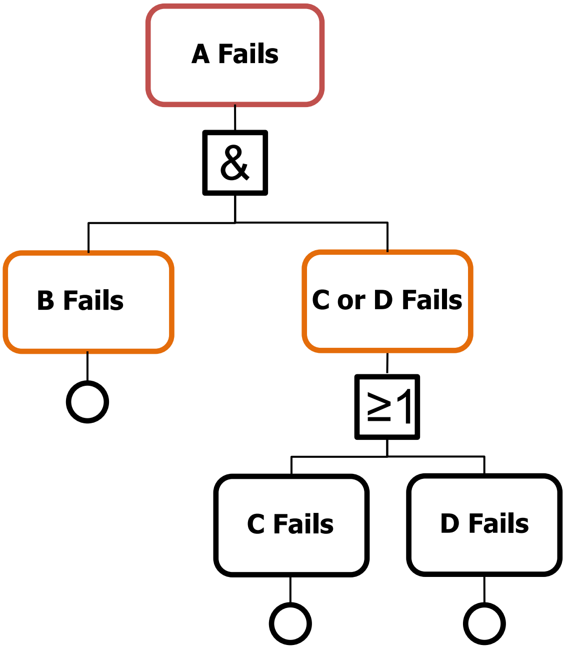

Figure 7: Fault Tree Analysis (FTA)

Figure 7 illustrates the fault-logic approach of the FTA, whereby the event "A Fails" represents the Top-Event. The Combination of the basic events "B Fails" and "C or D Fails" leads to the aforementioned Top-Event. The failure of either the components C or D, which leads to the basic events "C Fails" or "D Fails", leads to the event "C or D Fails".

The Failure Modes and Effects Analysis is probably the most popular and widely used method to systematically identify failure modes and their consequences within a system to conduct RAMS (Reliability, Availability, Maintainability, Safety)-Analysis. The procedure of the FMEA is mainly team-based. This means that its execution is mainly carried out by experts from different disciplines to detect failure modes, while considering different aspects and their views of the product to be investigated on. One of the main characteristics is the risk assessment and for optimization actions and countermeasures [17].The Risk Priority Number (RPN) quantifies the criticality of the detected failures with regard to their impact on the product, detective and preventive measures. Moreover, it is equal to the product of the Severity (S), Occurrence

(O) and Detection (D). A scale of numbers ranges from 10 to 1 for the Risk Assessment Value is used for the evaluation of the criticality level, whereby 10 will be assigned at high risk or poor evaluation and 1 for low risk or good evaluation. Priorities will then be specified after having identified assessments and the Risk Priority Number (RPN) to draw up optimization and improvement measures related to the design concept and eliminate weaknesses. The whole process is illustrated in figure 8.

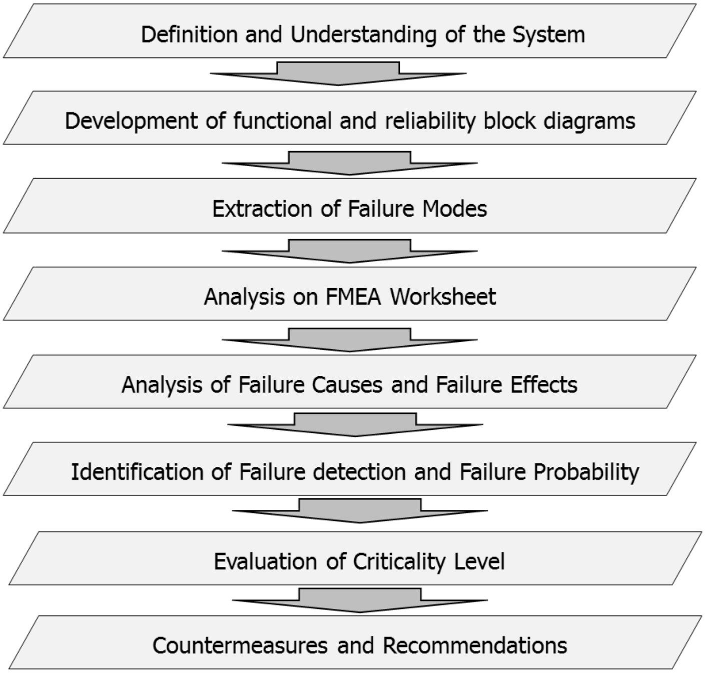
Definition and Understanding of the System
Development of functional and reliability block diagrams
Extraction of Failure Modes
Analysis on FMEA Worksheet
Analysis of Failure Causes and Failure Effects
Identification of Failure detection and Failure Probability
Evaluation of Criticality Level
Countermeasures and Recommendations

Figure 8: Failure Modes and Effects Analysis Procedure [2]

3 Limitations of the methods with relation to the cognitive coffee waiter

This section provides a set of limitations which might be encountered to the specific application of the abovementioned failure analysis methods to the design of cognitive products. Then we will consider the analysis and synthesis made by analyzing failures of cognitive products in specific cases to suggest enhancements and new considerations to be taken into account in future projects.

According to [10], Functional Hazard Assessment (FHA) is best required for early analysis of product failures when restricting our considerations to the conceptual design stage, in particular at the functional level. The identification of functions of the cognitive waiter as illustrated in figure 2 is helpful to consider related failure modes and safety integrity levels (SIL 1 -4: Negligible – Marginal – Critical - Catastrophic) as in most safety related recommendations, i.e. IEC 61508. An exemplary illustration of the assignment of functional failure modes to Safety Integrity Levels according to our considerations is showed in figure 9.

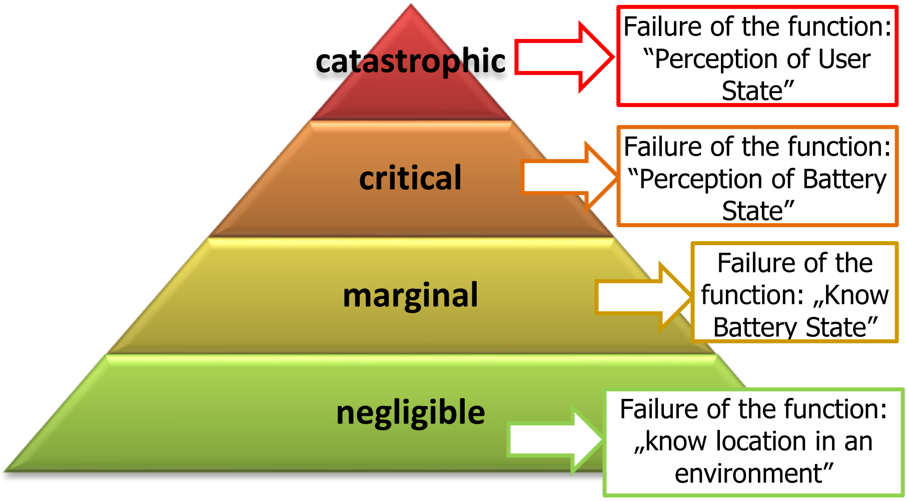
catastrophic
Failure of the function: "Perception of User State"
critical
Failure of the function: "Perception of Battery State"
marginal
Failure of the function: „Know Battery State"
negligible
Failure of the function: „know location in an environment"

Figure 9: Assignment of functional failure modes to safety integrity levels

Nevertheless, predicting the effects or consequences of these functional failures on the product or on the environment is a gigantic and non-trivial task. For this purpose, the users as well as the environment considerations need to be defined. In our view, environmental conditions are harsh, highly dynamic and difficult to predict at early design stages; therefore, specific use cases such as indoor use, charge at docking station, etc. shall be defined in order to analyze the effects of these functional failures in those specific use cases. FHA shows, as already mentioned, further limitations when it comes to consider functional dependencies. Referring to fig. 2, it is not obvious to clarify the dependency in terms of failure or misbehavior between the functions "perceive battery state" and "reflect on previous range". On the whole, traditional FHA cannot be applied to cognitive products due to their open-loop characteristics which allow them to strongly interact with environment as well as users and enforce these considerations with additional functional dependencies as part of the failure analysis even at early design stages.

Hazard and Operability Study (HAZOP) has been defined earlier on as another team-based technique to identify and analyze critical and catastrophic failures. The application of this method assumes that the system behavior must be as precisely clarified as possible to be able to estimate the causes or reasons as well as consequences of the system misbehavior. An Element-to-Element Analysis of the Functional Model coupled with the use of guide words such as "More, Less, Too Little, Too Early, No or Not" as stated in [19] might be according to the functional model of the coffee waiter (fig. 3) useful for safety analysis i.e. to clarify why the wrong perception of the battery state might have an influence on the route planning. However, the deviations from the intended system might be ambiguous at this abstract functional level. The use of this method is not appropriate for a more detailed model (fig. 2) but rather for functional models which exhibit flows (signal, data, information). Integrating environment uncertainties as well as users considerations, which might lead to system disturbances, need to be considered. In our considered view of the nondeterministic behavior of cognitive products, investigations on reliability and safety cannot be therefore undertaken on a deterministic way due to the unpredictability of the environment. Failure analysis of cognitive products at the functional level must be enhanced with behavioral and structural considerations. As HAZOP is not suitable for detailed models of cognitive products at tangible level where components are defined, it cannot be sufficient alone for safety or reliability validation.

Höfler defined in [2] the FMEA as a generally recognized method to systematically analyze failures systems by means that external influences can hardly be taken into consideration. Moreover, the system dynamic behavior as well as the successive sequences cannot be considered from the functional model (fig. 3) are not considered. The assumption that failures are independent from each other or cannot occur simultaneously cannot be reasonably considered for cognitive products. This means that traditional FMEA investigates the effect of a single failure while the rest of the system is considered to function properly. By looking at figure 10 that i.e. the misestimation of the battery states and the misjudgment of the optimal route cannot happen at the same time. But this assumption cannot be scientifically approved in this example. In all, we cannot leave aside the dynamic behavior of the system even at this conceptual design stage because it is in the essence of cognitive products to operate in the real world environment, thus interacting with users or other products. Therefore, only a dynamic system as well as failure behavior can be considered.

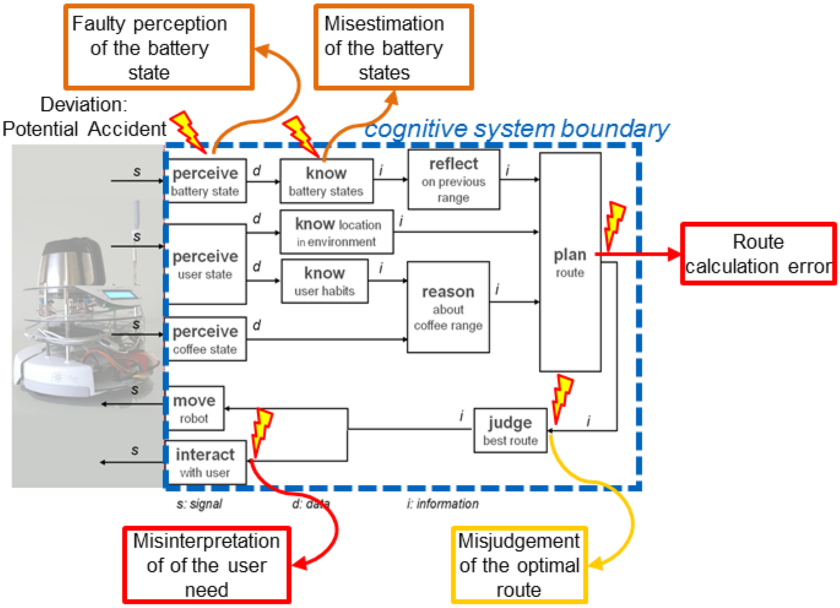
Faulty perception of the battery state
Misestimation of the battery states
Deviation: Potential Accident
cognitive system boundary
perceive battery state
know battery states
reflect on previous range
know location in environment
perceive user state
know user habits
reason about coffee range
perceive coffee state
plan route
Route calculation error
move robot
interact with user
judge best route
s
d
i
s: signal
d: data
i: information
Misinterpretation of of the user need
Misjudgement of the optimal route

Figure 10: Cognitive coffee waiter: deviation from the system behavior

The first step by the evaluation of the robustness of cognitive products with the Fault Tree Analysis (FTA) is to find in a deductive manner events or causes that lead to failures of the system, which are referred to in this context as Top-Events. One strength of the FTA is its Top-down approach to search for the failure causes regardless of whether they are in the software or in the hardware. The difficulty lies in the modeling of system failures. FTA can certainly not model all system failures. As already mentioned, it is necessary to consider use cases, in which a reduced amount of reduced can be considered. One additional weakness of the FTA is that only the failures that lead to the considered Top-Event can be considered. Even some basic events which might lead to failures can be overlooked. Furthermore, it is not clear how environmental conditions and user behavior, which might also lead to the failure of the system, can be integrated since they cannot be considered as static.

4 Discussion and Conclusion

In this contribution, we tackled the integration of the failure analysis methods in the conceptual design stage of cognitive products. After the definition and the functional modeling of cognitive products in the first step, we briefly covered the basic concepts of the Functional Hazard Assessment (FHA), Hazard and Operability Study (HAZOP), Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA) for the qualitative failure analysis. Further, these methods have been analyzed if and in what extend they are suitable for the failure analysis of cognitive products. Deriving and integrating real world conditions as well as user interactions, which must be considered and might lead to system failures, represents a challenge for the failure analysis of cognitive products. It seems realistic to argue and analyze the failure behavior of cognitive products on the base of predefined usage scenarios in order to limit the amount of failures to be analyzed. But despite that, these traditional failure analysis methods have several weaknesses which need to be compensated. FMEA does not consider the dynamic system behavior as well as common cause failures; FTA can only illustrate failures that lead to the top event system failure; FHA and HAZOP do not take the functional dependencies into consideration and remain to abstract for later design stages and implementation.

However, we are aware of the difficulties related to the integration of these aspects. We can only argue on the base of these usage scenarios otherwise the amount of failures which may occur during the operation and which are to be considered will grow explosively, though harshly dynamic environmental considerations as well user integration will contribute to the better understanding of the failure behavior analysis of cognitive products.

5 Acknowledgement

This research is part of the Innovation@CoTeSys project within the Cluster of Excellence, Cognition for Technical Systems – CoTeSys (www.cotesys.org), funded by the Deutsche Forschungsgemeinschaft (DFG).

Literature

[1] Graham, J .H. et al.: "Research Issues in Robot Safety", Carl Hanser Verlag, München, 2002.

[2] Höfler, A.:" Mechatronik-FMEA: Erweiterung der Methode für die Analyse von mechatronischen Systemen", VDM Verlag Dr. Müller, 2008

[3] Metzler, T., Shea, K.: "Cognitive Products: Definition and Framework". In: Proceedings of the 11th International Design Conference – DESIGN 10, Dubrovnik, pp. 865-874.

[4] Sop Njindam, T., Paetzold, K.: "An investigation of the reliability of fault-tolerant cognitive technical systems". In 8th European Conference on Computing and Philosophy, ECAP10

[5] Metzler, T., Shea, K.: "Taxonomy of Cognitive Functions". In: Proceedings of the International Conference on Engineering Design, ICED11, Copenhagen, Denmark, 2011, pp. 330-341.

[6] Birolini, A.: "Reliability Engineering: Theory and Practice", 6th Edition, Springer Verlag, Zurich, September 2010

[7] Alter W. & Logan J.: "NASA goes to ground - National Aeronautics and Space Administration", Whole Earth Review, 1991

[8] Harashima et. al.:" What is it? Why, and How" An Editorial, IEEE/ASME Transactions on Mechatronics, Vol. 1, No 1, 1996, S. ¼

[9] Stancliff, S. B.: "Planning to Fail: Incorporating Reliability Analysis into Design and Mission Planning for Mobile Robots", PhD-Thesis, Carnegie Mellon University, Pittsburgh, 2009.

[10] Wilkinson, P. J. et. Al. "Functional Hazard Analysis for Highly Integrat ed Aerospace Systems", Certification of Ground/Air Systems Seminar, Ref (No. 1998/255), IEEE, 2002

[11] Metzler, T., Shea, K.: "Lessons Learned from a Project-Based Learning Approach for Teaching New Cognitive Product Development to Multi-Disciplinary Student Teams." In: Proceedings of the ASME 2011 IDETC/CIE 2011, Washington, DC, USA.

[12] Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Berlin 2008.

[13] Erden, M.S.; Komoto, H.; van Beek, T.J.; D'Amelio, V.; Echavarria, E.; Tomiyama, T.: A review of functional modeling: Approaches and applications. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 22, 2008, pp.147-169.

[14] Stone, R.B., Wood, K.L.: Development of a Functional Basis for Design. In: Journal of Mechanical Design, Vol. 122, December 2000, pp. 359-370.

[15] Strube, G.: Modelling Motivation and Action Control in Cognitive Systems. In: Schmid, U.; Krems, J.; Wysocki, F.: Mind Modelling. Berlin, Pabst, 1998, pp. 89-108

[16] Beetz et. Al.: „Cognitive Technical Systems – What is the Role of Artificial Intelligence?"

[17] Bertsche, B.:"Reliability in Automotive and Mechanical Engineering", Springer Verlag, Stuttgart, 2007

[18] Sop Njindam, T., Paetzold, K.: "Design for Reliability: An Event and Function-Based Framework for the Failure Behavior Analysis of Cognitive Products in the Conceptual Design Stage", In: Proceedings of the International Conference on Engineering Design, ICED11, Copenhagen, Denmark, 2011

[19] Earthy, J. V.:" Hazard and Operability Study as an Approach to Software Safety Assessment", Colloquium on Hazard Analysis, IEEE, August 2002

[20] XIANG, J. et. al.:" Fault Tree and Formal Methods in System Safety Analysis", 4th International Conference on Computer and Information Technology, 2004

[21] Stamatelatos et. Al. "Fault Tree Handbook with Aerospace Applications", NASA Office of Safety and Mission Assurance, 2002