

Failure and Change Impact Analysis for Safety-Critical Systems - Applied on a Medical Use Case

Philipp Lohmüller, Julia Rauscher, and Bernhard Bauer

Department of Computer Science, University of Augsburg, Germany
{philipp.lohmueLLer, julia.rauscher, bauer}@informatik.uni-augsburg.de

Abstract. Nowadays, safety-critical systems are used in various domains including Internet of Things of medical devices. However, such systems are usually very complex and fault-prone. This means, safety, security and real-time aspects are often only insufficiently considered. To mitigate or avoid safety-critical failures, it is mandatory to analyze effects by means of a failure and change impact analysis. In this paper, we propose an approach to analyze a hierarchical structured model to determine critical goals. Afterwards, the effects and impacts of failures are calculated and determined to identify components which have a need of counter measures. Furthermore, it is analyzed which kind of effects these counter measures will have within the hierarchical model. Finally, the developed approach is evaluated by means of a realistic medical use case.

Keywords: Impact Analysis · Risk Assessment · Safety-Critical System.

1 Introduction

The level of complexity in IT systems is rising constantly. Therefore, these systems are also getting more complicated and hardly manageable. However, if there is a safety-critical system, like in automotive or medical field, e.g., to guarantee the airbag triggering or the correct administration of drugs, humans or assets, like data, can be endangered if the systems was not controlled. As 50% of flaws happen in the design phase [18] an identification and elimination of safety and security vulnerabilities at an early stage is essential. However, it is not enough to identify potential failures as weak points can be complex and have impacts on other components or relations too. To remove or mitigate the weak points components with a need of counter measures (CMs) have to be identified as well. As existing approaches do not combine the identification of weak points, the impacts of failures and the subsequent determination of CMs, we developed a holistic process to increase the safety of safety-critical systems. To address the aforementioned issues our approach identifies critical elements in an architecture model by the usage and adaptation of architecture analysis approaches which enable not only to identify flaws, but also their impacts and needs of CMs.

Therefore, we developed an approach which uses a safety goal hierarchy (SGH) combined with FMEA to identify weak points and conduct a failure impact analysis (FIA) to determine failure effects. Following, architectural changes will be recognized through a change impact analysis (CIA). The paper is structured as follows: Section 2 contains related work for failure and change impacts as well as safety-critical systems. Afterwards needed basics of Failure Mode and Effects Analysis (FMEA), Bayesian Belief Networks (BBN) and dependencies between safety and security are presented. Section 4 describes the approach in 4 detailed steps, which are evaluated in Section 5 with aid of a medical smart home use case. A conclusion and outlook are given in Section 6.

2 Related Work

There are numerous projects and scientific publications with respect to FIA and CIA as well as safety-critical systems, which are presented hereinafter.

2.1 Failure and Change Impacts

The work of Langermeier et al. [10] provides a CIA approach based on Enterprise Architecture Models. By means of their approach, which is based on a data-flow analysis technique, it is analyzed which model elements are affected. The algorithm of the authors aims to apply a CIA in context of Enterprise Architecture Models. However, our paper proposes an approach how to apply FIA and CIA taken safety-critical concerns like, e.g., safety and security into account. [16] propose an Eclipse based tool named Chianti and analyzes change impacts of regression or unit tests. By means of the execution behavior a set of affected changes is determined for each affected test. Chianti does not analyze change impacts on models but on Java code. Such as [10], Ren et al. [16] don't consider safety-critical aspects in their approach in order to enable a maximum degree of safety and security by means of the CIA. The paper of Hanemann et al. [5] dealing with resource failures, which might endanger service level agreements by influencing services. Therefore, [5] presents an approach, which identifies the effect of resource failures with respect to the corresponding services and service level agreements. In addition to those effects, a technique is proposed in order to improve services and to provide them. Such as [10] and [16], Hanemann et. al [5] also do not take safety-critical issues into account. In summary, each of the presented scientific publications provide either a FIA or a CIA but there is no combination of them. Furthermore, all publications do not consider the topic of safety-critical systems.

2.2 Safety-Critical Systems

For instance, there is a project, which is called SESAMO (**S**ecurity and **S**afety **M**odelling) and focuses on safety and security requirements, aiming "to develop a component-oriented design methodology based upon model-driven technology,

jointly addressing safety and security aspects and their interrelation for networked embedded systems in multiple domains” [1]. This project focuses on identifying safety and security hazards in order to calculate a trade-off between contradicting safety and security issues. Furthermore, another project concerning safety is called SafeCer (**S**afety **C**ertification of Software-Intensive Systems with Reusable Components). The purpose of this project is to increase “[...] efficiency and reduce(d) time-to-market by composable safety certification of safety-relevant embedded systems.” [9] The main focus of this project is to provide a procedure of composing safety arguments for a system by reusing of already certificated arguments of subsystems. In this way, it enhances efficient safety assurance and certification. Furthermore, there is the work of Lohmüller et al. [12], which proposes an approach for calculating trade-offs between contradicting safety-critical concerns. These include safety, security and timing. However, it aims to guarantee an optimal solution, which is as safe as possible. When comparing these works, one will realize that all of them cover safety and security. The work of [1] and [12] even combines safety with security or security and timing. In contrast to this paper, no scientific work integrates a FIA or CIA in their approaches.

So far, it has not been scientifically evaluated how to combine the results of a FIA with a CIA in context of safety-critical systems. Therefore, the approach, which will be presented in this paper, is innovative.

3 Basics

Following, essential basics for the concept of this paper will be presented. These include the FMEA, BBN and the interplay between safety and security.

3.1 Failure Mode and Effects Analysis

Nowadays, the software development process in context of safety-critical systems requires risk assessment. The FMEA is a widely used and established technique and is applied in different domains like, e.g., medical information science, automotive, avionics and railway industry. It is purpose of the FMEA to mitigate risks as much as possible. This is done by detecting and preventing failures. For the failure prevention, it is essential to indicate and to prevent failures in early stages of product cycle. The later a failure will be indicated the more expensive the development costs. Accordingly, the costs will increase about 10 times for each posterior stage [2]. The failure detection has four essential goals:

1. Detection of possible fault sources, which can cause failures
2. All causes and consequences must be identified, mitigated or avoided
3. Faultless organization of process during the development cycle
4. Vulnerabilities of the system, products or processes must be identified in order that a constructive revision can be performed

To prevent and detect failures it is necessary to determine potential risks by means of the FMEA. whereas occurrence complies with the probability whether a hazard occurs. Severity corresponds to the severity of hazard. The detection complies with the probability that a hazard will be detected. Each of the three factors can range between 1 and 10, i.e., the RPN can range between 1 and 1000. In general, the lower the RPN the better the potential risk. Depending on the value of the RPN, the degree of risk and the necessity of CMs can be identified by means of Table 1. [2]

RPN	Risk of Error	Counter Measure
RPN = 1	none	no CMs required
$2 \leq \text{RPN} \leq 50$	acceptable	additional warning required
$50 < \text{RPN} \leq 250$	medium	additional protective CMs required
$250 < \text{RPN} \leq 1000$	high	constructive CMs absolutely required

Table 1. Interpretation of the RPN [3]

3.2 Bayesian Belief Networks

BBN are highly complex networks which represent the probabilities of conditional dependencies of variables. Many research approaches of plenty research fields used or described BBN, e.g., [8], [17] or [4]. As a detailed description is out of scope in this paper, the most important parts are described afterwards. The formulas and theorems are abstracted from [14]. BBN are probabilistic, graphical models which are used to represent and calculate the conditional probabilities of model elements. Important characteristics are directed and acyclic relations, random variables with discrete states and dependencies to ancestor and descendant nodes. These networks are not restricted to a specific field. The majority of use cases for BBN are networks with questions about dependent probabilities of nodes with different possible states, e.g., the determination of correctness of a disease test. To determine the corresponding conditional probability distribution of every node in the graph a Conditional Probability Table (CPT) has to be defined. This table contains all possible combinations of the diverse states of ancestor nodes to determine the probability of these combinations. Depending on the leading question a model can be analyzed with different approaches and formulas. However, these are the required formulas of our approach:

Bayesian Theorem: $P(B|A) = P(A|B) * P(B)/P(A)$

Conditional Independencies: $P(AB) = P(A) * P(B)$

Markov Assumption/Joint Probability: $P(X) = \prod_{i=1}^n P(X_i | \text{ancestor}(X_i))$

3.3 Influences of Security on Safety

Nowadays, it is possible that security violates safety aspects. The following example from the past demonstrates this scenario in more detail. Due to the terrorist

attacks of September 11th, 2001 on the World Trade Center in New York City it has been decided to perform stricter security measures. Given the fact that the airplanes from the terrorist attacks has been hijacked a security measure has been introduced that the access to the cockpit is denied during the flight. This security precaution became an obstacle for the Germanwings flight 4U9525 on March 24th, 2015, which crashed in the French Alps. The captain within the cockpit has full control over the door and can even inhibit emergency access. Based on voice records, the co-pilot is suspected of deliberately destroying the plane while preventing the captain from reentering the cockpit. This means, increasing security against hijackers intensified reliance on the pilot being left on his own and carrying full responsibility for flying the airplane [19]. As demonstrated by this example, security has decisive impacts on safety.

4 Concept

After presenting related work and required basics we introduce our impact approach. The concept is based on diverse architecture analysis approaches, which are mighty tools in the design phase of system development. These kinds of analyses can check different aspects depending on the application field and the leading question of analysis. One field is Enterprise Architecture Management (EAM), which uses the analyses to outline and check coherences of business components, relations and processes. [15] provides an overview and classification of these analysis types with their several aims and techniques.

The concept of this paper is subdivided into four essential steps, which are presented in Figure 1. First, it is necessary to define a SGH in order to model safety-critical matters and to take trade-offs into account. On the basis of this SGH, potential failures will be identified by applying the FMEA. Step 3 conducts a FIA to determine impacts and effects of these potential failures and aims in identifying components which requires CMs. However, applying new CMs involve modifying other goals within the SGH. These goals will be identified and the effect types are determined. The following subsections will cover the aforementioned steps in more detail.

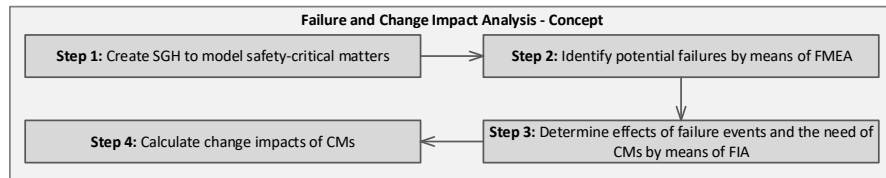


Fig. 1. Concept Picture of the Approach

4.1 Modeling SGH

The SGH (cf. Figure 2) is a hierarchical structure, i.e., there must be a root goal, which represents a safety aspect like, e.g., a system or a part of is acceptably safe. The root goal is refined by further safety-critical concerns, which influence the safety of this goal. These include, e.g., security or timing. Goals, which cannot be refined anymore are called Point of Vulnerabilities (POVs) and thus represent vulnerabilities of a system. Goals and POVs must accomplish quality attributes, i.e., they are annotated with attributes including a valid range of values. For instance, the goal *Airbag triggers in time* should be annotated with attribute `triggerTime` and range of values $0 < \text{triggerTime} \leq 100\text{ms}$. In safety-critical

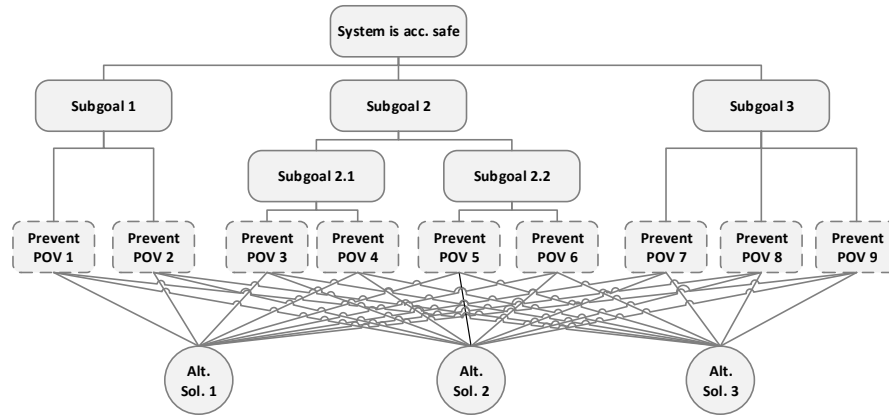


Fig. 2. Basic Structure of a SGH Supporting Three Alternative Solutions

systems there is a number of hardware components or software components installed, which have different safety requirements than other components, i.e., an individual component set must be defined for which the SGH should be applied. In this context, we speak about alternative solutions. Each of the alternative solutions accords with the aforementioned POV to a different extent.

4.2 Identifying Failures

The focus of this subsection is to assess risk of POVs in consideration of alternative solutions. Ideally, all POVs of a SGH should be prevented by the individual alternative solutions. However, in practice it is not compulsory possible, since the individual components of an alternative solution are not always in harmony with all POVs. To identify the failures, it is necessary to determine the RPN values (cf. Section 3.1) of the individual POVs depending on the corresponding

alternative solutions. This means in particular that we first need the probabilities of occurrence, severity and detection in order to finally determine the RPN assessments of them (cf. Section 3.1). Subsequently, it is necessary to classify the resulting RPN according to Table 1. If there is a risk of error classified with *medium* or *high*, i.e., $50 \leq RPN \leq 1000$ it is mandatory to identify the related elements with a need of corresponding CMs, which will be described in Section 4.3.

4.3 Failure Impact Analysis

After the identification of a potential failure we have to analyze the impacts on other elements. Since we already know the theoretical causal relation of a failure we do not have to analyze our system top-down. Therefore, we have to apply a bottom-up approach to monitor effects on elements which are logical dependent on the faulty element. Components which are highly negative affected by the failure have to be adapted by CMs. Architecture analyses are a possibility to identify these elements. As described above, EAM already uses these analyses successfully what yields us to conduct a concept transfer of a FIA approach by [6]. However, a few important aspects have to be changed to make the approach suitable for safety-critical systems. The most important change is the type of leading question for analysis as [6] analyzing their system top-down. Therefore, we need to shift metrics, variables, tools and use other BBN formulas.

Our FIA approach is divided in 5 **(A-E)** steps to clarify the execution of the analysis:

(A) A graphical representation of the whole system or of a specific process/service is required. To model our system we adapted on ArchiMate version 3.0.1 since it is an updated version with elements for Internet of Things (IoT) systems, which are kinds of safety-critical systems. Therefore, we distinguish between active/passive structure elements and behavioral elements for the representation of nodes. In addition, there are 11 relation types categorized in 4 classes which address diverse connections concerning structure, dependency and other aspects. Depending on the use case different layered approaches can be used. Exemplary, a layered architecture approach for IoT is presented in Figure 3. This approach differs strongly from EAM layers as IoT systems have other features like openness, flexibility, connection of autonomous devices with each other to measure and send data, etc. The presented layered approach is based on [13] and [7] and consists of 8 layers.

(B) Before mapping the model into a BBN model an analysis attribute, e.g., availability, has to be chosen. Depending on this attribute the BBN relations respectively dependencies can be determined.

(C) Translation of the model into a BBN structure to enable the modeling of probabilistic dependencies and the prediction capabilities. To transform the system model we conduct two sub-steps. First, we map every system element into a BBN node, i.e., variable. Thereby, all different kinds of element types transform and remain in the same layer. However, if an element is not involved into the

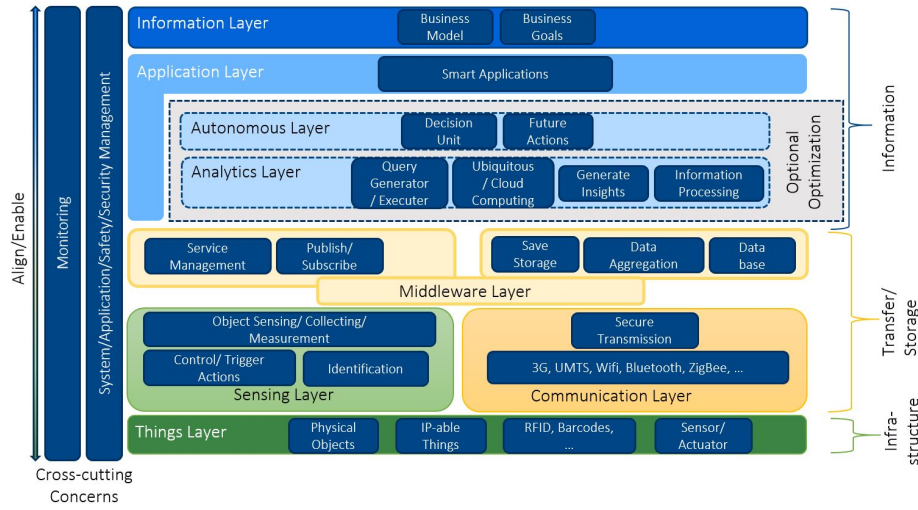


Fig. 3. Layered Architecture for IoT Systems

dependency structure it will be excluded. Secondly, the relation transformation has to be conducted depending on the attribute defined in (B). Every system relation is mapped to a causal BBN relation. Attention should be paid to the BBN characteristics presented in Section 3.2. Therefore, all acyclic or undirected relations of the system model have to be deleted or modified.

(D) Discretization of variables and determination of CPTs. As every BBN node represents a variable the variables' attributes get discretized and the corresponding probabilities are determined, e.g., a node *Weather* has the discrete attribute states with probabilities "Sunny=95%" and "Rainy=5%". Afterwards, the CPTs of every variable have to be defined including all possible combinations of the node's parents. This can be conducted by expert interviews, estimations or historical data analysis. To identify the impact of a failure we have to determine the nodes' probabilities before the occurrence of the failure. If causes of a failure have to be identified the Bayesian Theorem is used. However, if impacts of a failure are the focus of the analysis the Joint Probability formula is used with CPT values to identify the conditional probability of effects spreading out to the node.

(E) Calculation of impacts and determination of point for CM application. After the determination of the BBN model including probability states the failure is simulated and the probability of the faulty node is set accordingly. Consequently, the CPTs and probabilities of the BBN model have to be updated. The update can be conducted through new interviews or statistical calculations. Following this, a delta of probabilities of the nodes, before and after the failure, emerges which represents the impacts. This delta has to be evaluated by experts or a predefined scale, which divides the impacts into categories. Elements in a cat-

egory of highly negative impact triggered by failures need CMs. As a last step the most affected layer can be identified.

4.4 Change Impact Analysis

Assuming that experts have realized CMs in the last step, a CIA is needed to analyze which kind of effects necessary CMs will have. As a CM can require new nodes or the elimination of nodes, i.e., the SGH requires an update. Only an updated SGH can be used to evaluate the new safety-as-is status of a model in the future. First, it must be clarified which node types of the SGH are affected by the CIA and in which direction (top-down \downarrow or bottom-up \uparrow) the impacts will be propagandized:

1. Goal \rightarrow Goal: $\downarrow\uparrow$
2. Goal \rightarrow POV: \downarrow
3. POV \rightarrow Goal: \uparrow
4. POV \rightarrow Alternative Solution: \downarrow

This means, that an amendment of a goal can have both impacts on goals with higher abstraction level and goals with lower abstraction level. Moreover, POVs are involved as well. Modifications regarding POVs will affect goals on the next overlying layer. Furthermore, amendments of the POVs directly influence the alternative solutions. To perform the CIA step by step we need impact rules [10] with the following syntax:

$$A.X \rightarrow B.Y$$

In general, this statement expresses if source element **A** has the characteristic **X**, it follows that target element **B** has the characteristic **Y**. Concretely, this implies $A \in \{Goal, POV\}$ and $B \in A \cup \{AlternativeSolution\}$. The operations or effects, which are represented by **X** and **Y** are defined as follows: $X, Y \in \{noEffect, extend, modify, delete\}$. Extending a SGH element means to refine an element, e.g., by adding new elements. If an element is modified, the necessary information will be updated. Deleting a SGH element implies to remove it from the SGH. For instance, if the impact rule **G1.modify** \rightarrow **G2.extend** is applied, it means that *G1* will be modified and, in this regard *G2* must be extended as an impact. In this paper, we distinguish between Best-Case (BC) and Worst-Case (WC) CIA. The first one requires a minimum number of change impacts or lightweight change impacts and vice versa for the WC analysis. In the following, we define the change impact rules for the SGH, split into BC and WC (cf. Table 2). Hereinafter, the WC rules are explained in more detail. In case of deleting, modifying or extending a goal, the underlying goal or POV must be deleted, modified or extended as well. Furthermore, if a goal or POV is deleted, modified or extended the overlying goal must be modified in each case since information of the child nodes must be transmitted onto the corresponding parent node. The consequence of amendments on POVs is modifying all concerned alternative solutions. This can be justified by the fact that solutions directly and only depend on the POVs.

	BC	WC
Goal \rightarrow Goal \downarrow	A.delete \rightarrow B.extend	A.delete \rightarrow B.delete
Goal \rightarrow POV \downarrow	A.modify \rightarrow B.noEffect	A.modify \rightarrow B.modify
	A.extend \rightarrow B.modify	A.extend \rightarrow B.extend
Goal \rightarrow Goal \uparrow	A.delete \rightarrow B.extend	A.delete \rightarrow B.modify
POV \rightarrow Goal \uparrow	A.modify \rightarrow B.noEffect	A.modify \rightarrow B.modify
	A.extend \rightarrow B.modify	A.extend \rightarrow B.modify
POV \rightarrow Alternative Solution \downarrow	A.delete \rightarrow B.noEffect	A.delete \rightarrow B.modify
	A.modify \rightarrow B.noEffect	A.modify \rightarrow B.modify
	A.extend \rightarrow B.noEffect	A.extend \rightarrow B.modify

Table 2. Change Impact Rules**Algorithm 1** Change Impact Analysis

```

1: procedure CHANGEIMPACTANALYSIS(node, operation)
2:   if node isTypeOf Goal then
3:     if checkPrefAndApplyRule(node, getParentGoal(node), operation) then
4:       changeImpactAnalysis(g, getEffectType(node, operation))
5:     end if
6:     for all cg  $\in$  childGoals do
7:       if checkPrefAndApplyRule(node, cg, operation) then
8:         changeImpactAnalysis(g, getEffectType(node, operation))
9:       end if
10:    end for
11:    for all pov  $\in$  POVs do
12:      if checkPrefAndApplyRule(node, pov, operation) then
13:        changeImpactAnalysis(pov, getEffectType(node, operation))
14:      end if
15:    end for
16:  else
17:    if checkPrefAndApplyRule(node, getParentGoal(node), operation) then
18:      changeImpactAnalysis(g, getEffectType(node, operation))
19:    end if
20:    for all s  $\in$  solutions do
21:      ApplyRule(node, s, operation)
22:    end for
23:  end if
24: end procedure

```

So far, it was not specified to what extent the rules must be applied. Therefore, an algorithm (cf. Algorithm 1) is needed to consider this. The CIA is started initially by means of an impact rule according to Table 2. Subsequently, it is performed recursively until no more rules can be applied. Since the effects of some goals, POVs or solutions would be set multiple it is mandatory to define preferences of the effects types depending on BC or WC calculation. In case of BC the preferences are defined as $delete \succ_P modify \succ_P extend$ whereas for WC the preferences are defined as follows: $extend \succ_P modify \succ_P delete$. This

is due to the fact that it is more complex to extend a node more as to delete it. The resulting set of nodes including effect types, which must be enhanced by applying Algorithm 1 is defined as A_{CI} . So far, A_{CI} only consider the hierarchical structure of the SGH, but not any semantics within the SGH. Therefore, we also need the attribution within the SGH as proposed in Section 4.1. If the constraints of an attribute are violated because of invalid values, the SGH must be browsed for nodes with the same attributes and range of values. The set of all matches within the SGH is defined as B_{CI} . The final result of the CIA C_{CI} , i.e., the set of nodes which are affected by the violation of any constraints is defined as follows: $C_{CI} = A_{CI} \cap B_{CI}$.

5 Evaluation

After presenting the steps of our approach we conduct the evaluation with aid of a medical use case. As mentioned before IoT is a kind of safety-critical system if devices with safety goals are included, like IoT of medical devices or medical smart homes. Therefore, we use a system for Ambient Assisted Living (AAL) to evaluate our approach. Figure 4 depicts an exemplary AAL system with 4 medical or wellbeing devices delivering health support. As it exceeds the scope of our evaluation, just a small cutout of the system is shown and only 5 layers of the presented layered architecture in Section 4.3 are visible. The devices include sensors, actuators or RFID tags to measure and to trigger actions. For instance, the insulin pump measures data which are sent to the IoT-Hub which reviews the data and to trigger the SOS call if necessary.

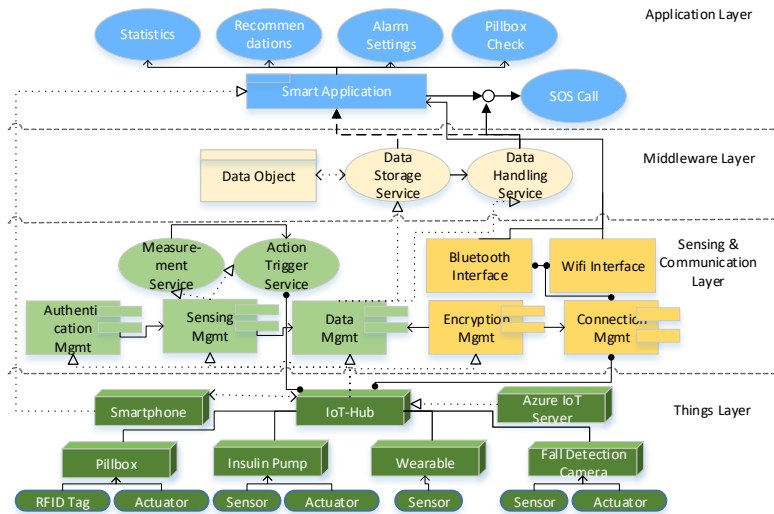


Fig. 4. AAL Use Case - Medical Smart Home

Step 1 and 2: First, we need to create a SGH (cf. Figure 5) representing an AAL, which is acceptably safe. For this purpose, the root node *AAL is acceptably safe* representing a safety goal is mandatory. The AAL SGH is further refined in consideration of the following context: For the success of an AAL system correctly functioning of the AAL sensors is a prerequisite, e.g., insulin pump sensors. Furthermore, AAL actuators must work correctly, e.g., activating pill-box. Moreover, the software running on an AAL system must work correctly. To ensure this, results of calculations must be correct and be performed in time without any delay. In addition, software must be acceptably secure against third party hacking attacks. The fourth aspect, which has been taken into account is the reliability of the AAL communication. For this purpose, the system must be secured against data theft and manipulation. Moreover, messages must be correctly transferred in time. As mentioned in Section 4.1 all nodes within the SGH must be annotated with an attribute. These attributes will be described in detail in step 4. The SGH of the AAL use case is extended by two alternative solutions.

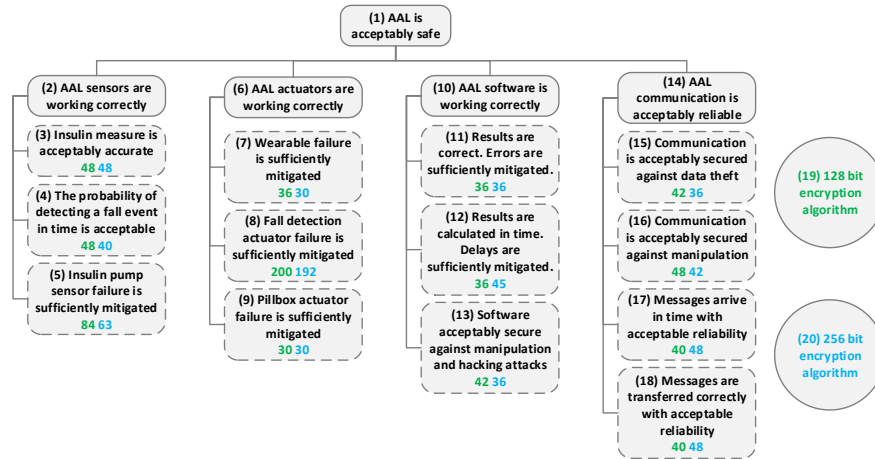


Fig. 5. AAL Use Case - SGH

In this use case, the SGH is applied for one of the following configurations:

1. 128 bit encryption algorithm
2. 256 bit encryption algorithm

Subsequently, the FMEA is performed as described in Section 4.2. In this context, it is essential to determine the corresponding RPNs of the POVs with regard to the individual alternative solutions. If there is a RPN classified with medium

or high risk of error (cf. Table 1) corresponding CMs are necessary. This affects two of the POVs: *Insulin pump sensor failure is sufficiently mitigated*, rated with a RPN of 84 or 63 and *Fall detection actuator failure is sufficiently mitigated*, which is rated with 200 or 192. An insulin pump sensor failure means that blood glucose level cannot be measured correctly. If the fall detection actuator fails, the SOS call cannot be performed.

Step 3: As the FMEA results yield potential failures in two nodes we have to check these nodes for impacts in case of a failure to be able to identify elements with a need of CMs. In the following case we merely inspect the failure event of node *Sensor* of the insulin pump. The FIA is conducted with the steps described in Section 4.3.

(A) The graphical representation of our use case is already shown in Figure 4 including different kinds of element and relation types of an AAL.

(B) As described, we need to choose an analysis attribute to be able to map the system into the BBN model accurately. As the diverse nodes can have multiple attributes we have to elect an attribute which is most suitable for the predictive leading analysis question. As we want to spot the impact of a failure of a data measuring node we decide to choose data quality as the analysis attribute. We define data quality as a combination of reliability, accurate amount of data measure sets and data intervals. To simplify the next step and the whole analysis we constrain data quality to correct data intervals as a irregular measurement can lead to faulty results.

(C) After having chosen the attribute we conduct the BBN mapping. At first, we map the system nodes into BBN nodes. In this use case we are able to transfer every element to a BBN node except the *Data Object*. This element has no dependencies for data intervals. Afterwards, we map the system relations into BBN relations. Hereby, we have to keep in mind the chosen analysis attribute as the conditional BBN relations can vary depending on the attribute. As this use case is highly dependent on data measurement we are able to transfer every directed relation. However, only some undirected relations could be modified and mapped. Figure 6 presents, i.a., the results of the mapping step.

(D) To determine the probabilities we have to discretize our analysis attribute data quality. As we consider the timing aspect of data quality we discretize the variables with the states "Up" for accomplishing the data interval and "Down" for contravening the interval. Afterwards, every node, i.e. variable, is assigned its own probability for both states. Once, the probabilities are set the CPTs are determined for every node depending on the combinations of their ancestor states. For instance, the probabilities of the *insulin pump* are "Up=60%" and "Down=40%" as well as "Up=45%" and "Down=55%" for the *wearable*. Table 3 shows a exemplary CPT of the IoT Hub. To be able to simulate the model random numbers were chosen for this use case. The scope of Figure 6 impeded the illustration of all probabilities and CPTs on the according nodes.

(E) As a last step we want to identify the impacts of the insulin pump sensor failure. As the probabilities before the failure event are already determined

Insulin Pump	Wearable	U	D
U	U	0.9	0.1
D	D	0,01	0,99
U	D	0,4	0,6
D	U	0,3	0,7

Table 3. CPT Example

we can simulate the failure now by setting the probability of the sensor on "Down=100%". Accordingly, all nodes which depend on the sensor have to update their probability and CPTs. Therefore, the Joint Probability formula by multiplying the CPTs is used. Afterwards, the delta of probabilities before and after the failure event can be identified. We present an exemplary impact identification on the nodes *Measurement Service* and *Connection Management*. For instance, Figure 6 displays the faulty sensor and the probabilities of both nodes for the attribute data quality before and after the failure. Since both nodes are negatively affected which leads to deterioration of more than 40%, both nodes need CMs to be prepared in case of a failure.

After the identification of effected nodes we have to transfer the knowledge to find the matching POV.

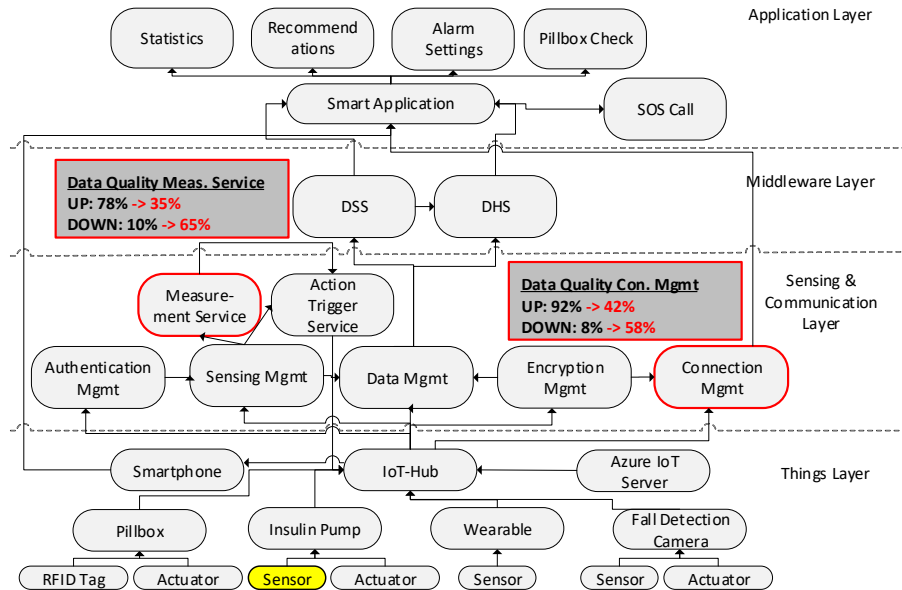


Fig. 6. BBN after Model Mapping with conducted analysis results

Step 4: Deductive to step 3, there is a CM necessary for the POV *Messages arrive in time with acceptable reliability*. However, it must be checked if there are further nodes within the SGH, which must be amended as well. Therefore, we need to apply the change impact rules according to Table 2. First, we have to check for the nodes' attributes which were annotated in step 1 and 2. Since we examine within this evaluation data quality, we annotated the corresponding nodes within the SGH with a suitable attribute. To match the analysis attribute data quality we use the node attribute `timeLimit` for the POVs *Results are calculated in time*, *Delays are sufficiently mitigated*, and *Messages arrive in time with acceptable reliability* as well as for the solutions *128 bit encryption algorithm* and *256 bit encryption algorithm*. Each of them are tagged with `timeLimit=200`, i.e., there is a maximum time limit of 200 ms in order to achieve the goals.

In this use case, we consider the WC scenario, i.e., $A_{CI} = \{1, \dots, 20\}$ and $B_{CI} = \{12, 17, 19, 20\}$. The entire sequence of change impact rules (necessary for A_{CI}) for this use case is listed in Table 4. According to Section 4.4 $C_{CI} = A_{CI} \cap B_{CI}$, i.e., $C_{CI} = \{12, 17, 19, 20\}$ since these nodes are also annotated with the corresponding `timeLimit` attribute. In summary, POV #17 from which the CIA has been started must be either deleted, modified or extended depending on CM action of #17. In addition, goal #12 and the solutions #19 and #20 must be modified. The CIA of this use case explicitly considers the specified time limit. When amending timing-critical goals or POVs, reliability is also fulfilled since messages arrive in time or calculations are performed in time. In this way, the desired number of messages or calculations can be done within a specified time limit.

delete	modify	extend
{17}.delete \rightarrow {14}.modify	{17}.modify \rightarrow {14}.modify	{17}.extend \rightarrow {14}.modify
	{14}.modify \rightarrow {1}.modify	
	{1}.modify \rightarrow {2,6,10,14}.modify	
	{2}.modify \rightarrow {3,4,5}.modify	
	{6}.modify \rightarrow {7,8,9}.modify	
	{10}.modify \rightarrow {11,12,13}.modify	
	{14}.modify \rightarrow {15,16,17,18}.modify	
	{3-5,7-9,11-13,15-18}.mod \rightarrow {19,20}.modify	

Table 4. AAL Use Case - Sequence of Change Impact Rules (WC)

In summary, we analyzed a medical AAL system for potential failures, their impacts and effects of realized CMs.

6 Conclusion and Outlook

In this paper, an approach has been presented in order to perform a FIA and a CIA in context of safety-critical systems, which has been demonstrated for

a medical use case. For this purpose some prerequisites must be met: An attributed SGH is required with subsequent FMEA analysis to identify potential failures. Afterwards, the impact of a failure event have been identified including the need of CMs by the usage of a FIA. Finally, we examine the effects of CMs through applying the change impact rules of the CIA. Due to the complexity of safety-critical systems, the utilization of impact analyses during design phase is becoming increasingly important. For future work, it might be useful to automatize some processes of this approach in order to renounce expert knowledge. These include, e.g., automation of CPT or a semantical analysis within the SGH. Furthermore, it might be useful to extend the approach of this paper by software product lines as used in [11].

Acknowledgment

This work has been partially supported by the German Federal Ministry of Economics and Technology (BMWi) in the framework of the Central Innovation Program SME (Zentrales Innovationsprogramm Mittelstand) within the project CBMD¹.

References

1. SESAMO project homepage. <http://sesamo-project.eu/> (2015), accessed January, 31st
2. Bertsche, B., Göhner, P., Jensen, U., Schinköthe, W., Wunderlich, H.J.: Zuverlässigkeit mechatronischer Systeme, Grundlagen und Bewertungen in frühen Entwicklungsphasen. Springer-Verlag Berlin-Heidelberg (2009)
3. Bundesverwaltungsamt: Fehlmöglichkeiten- und einflussanalyse (FMEA). Organisationshandbuch (2017)
4. Cai, B., Huang, L., Xie, M.: Bayesian networks in fault diagnosis. *IEEE Transactions on Industrial Informatics* **13**(5), 2227–2240 (Oct 2017). <https://doi.org/10.1109/TII.2017.2695583>
5. Hanemann, A., Schmitz, D., Sailer, M.: A framework for failure impact analysis and recovery with respect to service level agreements. In: 2005 IEEE International Conference on Services Computing (SCC'05) Vol-1 (2005)
6. Holschke, O., Närman, P., Flores, W.R., Eriksson, E., Schönherr, M.: Using enterprise architecture models and bayesian belief networks for failure impact analysis. In: International Conference on Service-Oriented Computing. Springer (2008)
7. Khan, R., Khan, S.U., Zaheer, R., Khan, S.: Future internet: the internet of things architecture, possible applications and key challenges. In: 2012 10th international conference on frontiers of information technology. IEEE (2012)
8. Koski, T., Noble, J.: Bayesian networks: an introduction, vol. 924. John Wiley & Sons (2011)
9. Kristen, E., Althammer, E.: Flexray robustness testing contributing to automated safety certification. In: Computer Safety, Reliability, and Security. pp. 201–211 (2015)

¹ <https://www.informatik.uni-augsburg.de/en/chairs/swt/ds/projects/mde/cbmd/>

10. Langermeier, M., Saad, C., Bauer, B.: Adaptive approach for impact analysis in enterprise architectures. In: Business Modeling and Software Design (2015)
11. Lohmüller, P., Bauer, B.: Software product line engineering for safety-critical systems. In: Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELWARD. Prague, Czech Republic (2019)
12. Lohmüller, P., Fendt, A., Bauer, B.: Multi-concerns engineering for safety-critical systems. In: Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development - Volume 1: MODELWARD. Funchal, Portugal (2018)
13. Microsoft: Azure iot reference architecture - version 2.0. Download Center (2018)
14. Neapolitan, R.E., et al.: Learning bayesian networks, vol. 38. Pearson Prentice Hall Upper Saddle River, NJ (2004)
15. Rauscher, J., Langermeier, M., Bauer, B.: Classification and definition of an enterprise architecture analyses language. Business Modeling and Software Design, 6th Int. Symposium 2016, Springer (2016)
16. Ren, X., Shah, F., Tip, F., Ryder, B.G., Chesley, O.: Chianti: A tool for change impact analysis of java programs. In: Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications. pp. 432–448 (2004)
17. Shin, J., Son, H., Heo, G., et al.: Development of a cyber security risk model using bayesian networks. Reliability Engineering & System Safety **134**, 208–217 (2015)
18. Viega, J., McGraw, G.R.: Building Secure Software: How to Avoid Security Problems the Right Way, Portable Documents. Pearson Education (2001)
19. Yu, Y.: Germanwings flight 4u9525: a victim of the deadlock between safety and security demands. <http://theconversation.com/germanwings-flight-4u9525-a-victim-of-the-deadlock-between-safety-and-security-demands-39386> (2015), accessed on May 3rd, 2019