IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 December 2023, accepted 19 December 2023, date of publication 26 December 2023,
date of current version 4 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347553

RESEARCH ARTICLE

A Framework for Evaluating the Safety of Health Monitoring Systems in the Home Environment

ZHENGGUO YANG
School of Information Technology, Shangqiu Normal University, Shangqiu, Henan 476000, China
e-mail: yangzhengguo@sqnu.edu.cn

This work was supported by Shangqiu Normal University under Grant 7001-700226.

ABSTRACT Health monitoring systems in the home environment are safety-critical. It is necessary to evaluate the safety of health monitoring systems. Conventionally, system safety is guaranteed through conformance to related safety standards or requirements. This conformance is qualitatively demonstrated by the safety case. However, the result of the qualitative evaluation is not straightforward to review. Moreover, two problems must be considered from the viewpoint of systematic evaluation. First, if the result of the evaluation of a system is unsafe, the traceability of the result to the system safety requirements must be considered to improve system safety. Second, re-evaluation may occur in case of changes to system safety requirements or related system design. Thus, this paper proposes a framework for quantitative evaluating system safety for health monitoring systems. The framework focuses on three models, i.e., system safety requirements structure, safety case, and Bayesian network, with five steps to solve the above concerns. Finally, the framework is applied to an example health monitoring system for demonstration.

INDEX TERMS Bayesian network, health monitoring systems, safety case, system safety evaluation framework, system safety requirements, traceability.

I. INTRODUCTION

Health monitoring systems (HMSs) in the home environment adopt information and communication technologies to sense health information from occupants and transmit it to the cloud for processing in order to provide appropriate and timely health services [1], [2]. Thus, HMSs are safety-critical as their failure or malfunction would cause harm or even death to occupants, especially the elderly [3], [4]. For example, an HMS system that failed to provide correct service to hypertensive patients based on collected health data would increase the risk of developing cardiovascular disease [5]. Therefore, it requires evaluating the safety of HMS systems.

Conventionally, the safety of a system is guaranteed by conforming to related safety standards or requirements; for example, a vehicle mounted system must comply with ISO 26262 for vehicle safety [6]. This conformance is generally qualitatively demonstrated by safety cases [7], [8], [9]. A safety case is a reasoned and compelling evidence-supported argument that a system is safe for a defined application in a given environment. Some works in the literature [10], [11], [12], and [13] transform safety cases into Bayesian networks, which enable the quantitative safety evaluation of a system. Bayesian networks (BNs) [14], [15] are probabilistic graphical models that express causal relationships of a set of variables on which to infer uncertain knowledge. The reasons to adopt BNs for quantitative system safety evaluation are triple. First, the knowledge the safety case presented should be retained since the knowledge is the theoretical foundation based on which a system can be concluded safe. This knowledge can be represented by probabilistic variables and their causal relationships of BNs. Second, safety cases are characterized by epistemic uncertainty, since it provides an inductive reasoning process with uncertainties in the evidence and arguments [16], [17]. BNs can deal with epistemic uncertainty by reasoning about probabilistic variables with uncertain states [14]. Third, safety cases are dynamic [18]. The states of some node variables could not be determined when transforming into BN. BN can handle this, as it can readily process incomplete data sets [19].

However, there are limitations to only transforming safety cases into BNs without explicitly considering system safety requirements from the perspective of systematic system safety evaluation. First, if a system is evaluated as unsafe,

The associate editor coordinating the review of this manuscript and approving it for publication was Mouquan Shen.

it cannot retrieve related system safety requirements based on the evaluation result to further improve the system safety. By retrieving related system safety requirements, system design and results of engineering activities, such as testing, that can prove that the system design satisfies the requirements can be scrutinized. Second, the safety requirements of the system would change since the HMS systems are self-adaptive and dynamic [20]. Thus, a re-evaluation of the system safety is required. In this case, the changes cannot be propagated to the BN through safety cases to enable re-evaluation.

To overcome the above two limitations, the safety requirements of the system must explicitly be linked to the safety cases that connect to the BNs in the first place. Thus, system safety requirements retrieval and change propagation can be enabled. To this end, requirements traceability [21], [22], [23] can be employed. It is used to link artifacts and to trace the link between artifacts. Artifacts are traceable elements like a requirement. The traceability in this paper is different from the requirements traceability. First, the traceability in this paper is to trace an evaluation result, if it is not satisfactory, to related system safety requirements for system safety improvement. Second, the artifacts are not only system safety requirements but also elements of safety cases and random variables of BNs. Third, the links are between the system safety requirements, the safety cases, and the BN, rather than between the requirements themselves. So, the link types are different.

Then, on the basis of the traceability, the two limitations discussed above can be solved. To retrieve related system safety requirements, highly influential random variables in the BN to the evaluation result must be identified. The random variables are then traced through safety cases to related system safety requirements. To this end, the sensitivity analysis [24] can be applied. Sensitivity analysis investigates the importance of model input in determining its output. The work of this paper takes the variable representing the safe state of an HMS system as the target while the other variables in the BN as input to investigate highly influential variables. For change propagation, changes to system safety requirements have to be captured first. The changes are propagated to the BN through traceability links. In other words, the safety case and the BN need to be revised in accordance with the changes.

Therefore, this paper proposes a framework for quantitatively evaluating the safety of HMS systems in the home environment. To the best of the author's knowledge, there has been no similar framework proposed in the literature. Thus, compared to the conventional way of evaluating system safety, i.e., safety cases, the contributions of this paper are summarized as follows.

- A framework is proposed to quantitatively evaluate the safety of HMS systems.
- The rules proposed to enable the transformation of the system safety requirements structure to the safety case and the BN.
- A traceability metamodel is proposed to build traceability models between the system safety requirements, the safety cases, and the BN.
- The framework enables requirements retrieval and change propagation based on the traceability.
- An application example is utilized to demonstrate that the limitations discussed have been overcome by the proposed framework.

The paper is organized as follows. Section II discusses related work. Section III introduces some preliminary knowledge to better understand the proposed framework. Then Section IV elaborates the framework. Next an application example is adopted to apply the proposed framework in Section V. Section VI discusses some issues based on the application example. Finally, Section VII concludes this paper.

II. RELATED WORK

This section introduces related work on the application of safety cases to evaluate system safety, the transformation of safety cases into BN, and the traceability of requirements.

A. APPLICATION OF SAFETY CASES

Safety case is the pivotal technique of the proposed framework. It has been applied to qualitatively evaluate the safety of various safety-critical systems. In safety management requirements for defense systems, Def Stan 00-56 [9] introduced a generalization of the safety case concept and prescribed the necessitate of safety cases in all phases of system development. [25] focuses on the usability of safety cases. The author integrates the Bowtie methodology into the process production to ensure usable safety in the field of nuclear power plants. The System-Theoretic Process and Analysis (STPA) [26] is recommended to assist in producing a safety case in the present paper rather than the Bowtie methodology. The reason is that STPA is a relatively new technique. The safety case evolves along with the system development process. Therefore, [18] proposed a dynamic safety case for aviation systems. The dynamic safety case framework consists of four steps, that is, identify, monitor, analyze, and response. The rationale is to monitor and analyze the identified uncertainties in the safety case and response appropriately. The dynamicity of the safety case in the present paper is due to changes to safety requirements. Changes are propagated to the safety case through traceability links. Safety case in the automotive domain is also a research interest. The paper [27] presents the content of the safety case in compliance with ISO 26262 and embeds it into the existing hierarchy of automotive safety. However, if an automotive system cannot be demonstrated safe by the safety case, improvement to system safety is not explicitly given.

B. QUANTITATIVE EVALUATION

There are a few techniques to ensure quantitative evaluation of system safety. A survey in [11] suggested that almost half of the techniques surveyed for the quantification of safety cases are based on BN. In [10] the safety case is represented in the Goal Structuring Notation (GSN) that will be introduced in Section III-B and is encoded in the BN by following some predefined rules. This paper proposes the transformation rules for transforming a safety case represented by GSN into BN based on [10]. There are some other related works. Each of these works was to solve some aspect of applying BN to obtain quantitative confidence in the safety case. The paper [28] discussed the suitability of employing BN to represent the safety case. Reference [29] adopted BN for quantitative confidence propagation in the safety case network. The work in [13] applied BN to obtain quantified confidence of each claim.

C. REQUIREMENTS TRACEABILITY

Requirements traceability defined in the literature is "the ability to describe and follow the life of a requirement in both a forwards and backwards direction" [23], [30]. As discussed in Sections I and IV-D, the differences between traceability in this paper and requirements traceability are artifacts and trace links. However, requirements traceability tools could be used to implement the traceability in this paper. For example, matrix, cross-referencing, hierarchy, and database, which the introduction of these tools can be found in [31] and [23]. Cross-referencing is adopted in the application example in Section V. This does not mean that the cross-referencing is more outstanding than others. One could select one based on the application needs and the features of the tool.

A traceability model varies since the application scenarios are different. It is impossible to provide a traceability model for all. Thus, a metamodel that is used to define a model is required. There are two types of languages to describe a metamodel, i.e., Ecore and UML (Unified Modeling Language) class diagram [32]. Ecore [33] is included in the core of the Eclipse Modeling Framework and can be used to describe models. The UML class diagram can be found in the Object Management Group standard [34]. The class diagram is adopted in this paper since it is widely used and is more familiar to the author.

III. PRELIMINARIES

This section briefly introduces four concepts, i.e., health monitoring systems, safety case, system safety requirements, and Bayesian network, to better understand the proposed framework.

A. HEALTH MONITORING SYSTEMS

Health monitoring systems (HMSs) in the home environment adopt information and communication technologies to sense vital signs of occupants (e.g., heart rate), home environment (e.g., room temperature), occupants' behavior (e.g., fall), etc. and preprocessed locally and transmitted them to the cloud for further processing in order to provide appropriate and timely health services [1], [2]. The home environment refers to the smart home whose definitions are classified into two categories [35]. One focuses on the home and the user. Another focuses on the building and the system. Sensing data is preprocessed locally for several reasons. For example, to obtain preliminary results in case the cloud service is unavailable due to network failure and to compress the data for better network bandwidth utilization. The sensed data is processed on the cloud by professionals such as the hospital to obtain more accurate results.

The components of an HMS system can consist of sensing, processing, actuating, communication and storage [1], [36] based on cloud-based smart home [35]. The sensing component is to collect various sensor data, for example, vital signs. The processing component includes two functions, namely perception and reasoning. Perception means extracting features through various properties. For example, to extract high- or low-blood pressure features from a person's blood pressure data. The reasoning is to predict and detect more complex situations or illnesses using rules or algorithms. Actuating refers to informing caregivers and medical professionals of health deterioration and the like. Each component should communicate with others through wired or wireless techniques. Sensor data and processed ones need to be stored locally or remotely in the cloud.

B. SAFETY CASE

In the safety-critical domains, a system is safe when it complies with related safety standards or requirements. This is demonstrated by the safety case. A safety case is a reasoned and compelling evidence-supported argument that a system is safe for a defined application in a given environment [7], [9]. The argument of a safety case includes claims and evidence. Claims are about the safety of a system, while evidence is used to support claims.

There are many ways to demonstrate the safety case, among which the goal structuring notation (GSN) [8] is predominantly adopted in the literature [37]. The GSN notation is a directed acyclic graphical argument notation that is capable of explicitly documenting the elements and structure of an argument and the relationship of an argument to related evidence. The claims are goals in the GSN notation, and their satisfaction is achieved through evidence or subgoals. The evidence can be engineering activities, such as testing, showing that a goal is satisfied [38]. Thus, the GSN notation provides a layered structure of arguments. An argument, except the bottom one, consists of a goal that is supported by one or more subgoals. A bottom-layer argument includes a goal that is supported by related evidence.

The core elements of the GSN notation are listed in Table 1. The core relationships between the core elements are described in Table 2.

C. SYSTEM SAFETY REQUIREMENTS

System safety requirements are prescriptive statements to be enforced in terms of the safety of a system in a given environment [9], [30], [31]. They prescribe to mitigate, eliminate, or monitor the identified hazards. The initiation of them can be from requests from stakeholders or safety standards. For example, stakeholders may require an HMS system to provide

TABLE 1. The core elements of GSN notation.

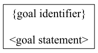
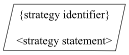
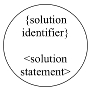
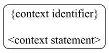
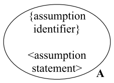
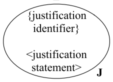
Element	Rendering	Explanation
Goal		A requirements statement which expressed as a claim concerning some aspect of the system design, implementation, operation, or maintenance.
Strategy		It provides the rationale of the child goals supporting their parent goal. Thus, it is the interface between a goal and its supporting goals.
Solution		It presents a reference to an evidence item. It provides the backing to state that a goal has been achieved.
Context		It depicts the boundary over which a goal or strategy is made. It applies to a goal structure, and any other goal structure that has a connection to the goal structure should be within the context boundary.
Assumption		An unsubstantiated statement believed to be true about the system, its operating environment, and the technologies employed. It can be connected to a goal or a strategy and applies to the entirety of the goal structure.
Justification	 {justification identifier} <justification statement> J	A statement that provides a rationale behind the adoption of a strategy or the depiction of a goal. It applies to the element to which it is linked, not the entirety of the goal structure.

TABLE 2. The core relationships of GSN notation.

Relationship	Rendering	Explanation
SupportedBy	→	It allows support relationships between a goal to another goal, a strategy, and a solution.
InContextOf	⇾	A contextual relationship between a goal to a context, a justification, and an assumption; and a strategy to a context, an assumption, and a justification.

the correct service to prevent a hypertension patient from developing cardiovascular disease.

System safety requirements are system requirements, so they can be documented in ways that treat system requirements. On the one hand, the system requirements are documented hierarchically. A high-level requirement is satisfied by satisfying all its connecting bottom-level requirements through intermediate-level requirements. On the other hand, system safety is considered a control problem [39]. System safety requirements are elicited along with the decomposition of system-level hazards into factors related to controllers of the system. The controllers must constrain the behavior of the system. Therefore, system safety requirements can also be documented hierarchically.

In addition to system requirements, the general building blocks of the requirements hierarchy also include the requirement context, etc. [30]. A brief introduction of them is in Table 3. There are tools that can be used to build a hierarchy of safety requirements of the system. A prominent one is the requirement diagram of the system modeling language (SysML) [40]. SysML is a general-purpose system modeling language. It extends the UML to address the requirements [34]. The SysML requirements diagram is capable of illustrating the requirements hierarchy and their relationships with other building blocks.

D. BAYESIAN NETWORK

Bayesian networks (BNs) [14], [15] are a probabilistic graphical model that represents the joint probability distributions of a set of variables in a given domain. It is formally defined as a two-tuple, that is, $BN = (\mathbb{G}, P)$, where $\mathbb{G}$ represents the directed acyclic graph and P is the joint probability distribution. $\mathbb{G}$ is also a two-tuple and $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ denotes a set of nodes that represents random variables and $\mathbb{E}$ is a set of directed edges between nodes. P is defined as the factorization of the joint probability into the product of conditional probabilities, that is, $P(V_1, V_2, \ldots, V_n) = \prod_{i=1}^{n} P(V_i|pa(V_i))$, where $V_i \in \mathbb{V}$ $(i = 1 \ldots n)$, and $pa(V_i)$ is the parent nodes of V_i.

The construction of BNs has two parts based on the BN definition above. One is to build the directed acyclic graph $\mathbb{G}$. Another is the elicitation of conditional probabilities. The former is to parameterize domain knowledge and clarify their dependencies. The latter is to determine the probability of a node variable at some states dependent on other node variables. Both can be learned from data or elicited by domain experts to follow some formal procedures [19], [41], [42].

One critical application of BN is to reason about the probability of a node variable at a state, given some evidence. Evidence is that the dependent node variables are in some known states. One can refer to the sprinkler example in [19] for better understanding. The present paper employs BN to evaluate the probability that an HMS system is in a safe state, given some evidence that influences the safety of the system. The variable node of BN that represents the safe state of an HMS system is the target node.

IV. SYSTEM SAFETY EVALUATION FRAMEWORK

The safety of an HMS system is determined by the compliance of the system with related safety requirements.

TABLE 3. General elements to build a system requirements hierarchy.

Element	Explanation
Requirement justification	It explains the necessity of a safety requirement and demonstrates the rationale and sufficiency of child requirements in supporting their parent requirement.
Requirement context	The facts or assumptions about a system and/or its environment in which the requirement applies.
Scheme	A solution to implement a safety requirement and related verification/validation activities. For example, it could be testing, etc. A scheme connects to a bottom safety requirement of the hierarchy.

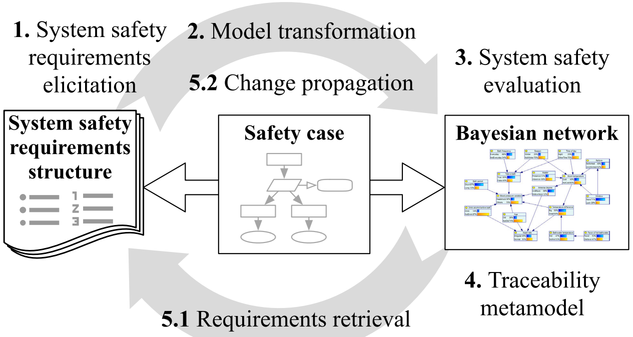

1. System safety requirements elicitation
2. Model transformation
5.2 Change propagation
3. System safety evaluation
System safety requirements structure
Safety case
Bayesian network
4. Traceability metamodel
5.1 Requirements retrieval

FIGURE 1. The proposed framework for evaluating system safety.

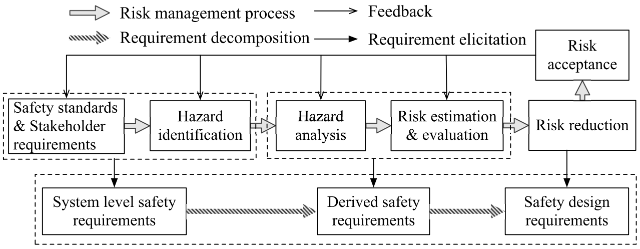

Risk management process
Feedback
Requirement decomposition
Requirement elicitation
Risk acceptance
Safety standards & Stakeholder requirements
Hazard identification
Hazard analysis
Risk estimation & evaluation
Risk reduction
System level safety requirements
Derived safety requirements
Safety design requirements

FIGURE 2. System safety requirements elicitation along with the risk management process based on [9].

The system safety evaluation consists of calculating the degree of confidence in the conformance and determining the acceptance of the result. To this end, the proposed framework consists of three models, i.e., system safety requirements structure, safety case, and the BN, as shown in Figure 1. The safety case is the conventional tool for qualitative safety evaluation. It is transformed into BN for quantitative safety evaluation. System safety requirements are required due to two reasons. First, it is used when retrieving related safety requirements to improve system safety. Second, changes to the system safety requirements must be propagated to the BN through the safety case for re-evaluation. Then five steps are proposed with respect to the three models. Step five includes 5.1 and 5.2 in the figure, which means that requirement retrieval and change propagation can be done in parallel with no preference.

There are two situations about the existence of the system safety requirements structure and safety case, i.e., preexist and not exist. This paper deals with the latter case. For the former case, step one and two would be *linking elements of the safety case to the system safety requirements structure*, and *transforming the safety case into BN*, respectively. In this case, step one is to set up the traceability model between the safety case and the requirements structure. This can be achieved by using the metamodel introduced in Section IV-D. Step two can follow the transformation rules in Section IV-B2.

A. STEP 1: SYSTEM SAFETY REQUIREMENTS ELICITATION

Since system safety requirements are prescribed to mitigate, eliminate or monitor identified hazards, as discussed in Section III-C, their elicitation can occur along with the risk management process, as shown in Figure 2. Risk management includes hazard identification and analysis, risk estimation and evaluation, risk reduction, and risk acceptance [9]. It can start with an undesired event, for example, an accident, to identify system-level hazards. Then the identified system-level hazards can be analyzed, and related risks are estimated and evaluated. Finally, risks are reduced through countermeasures and demonstrated to be acceptable.

For a better understanding of the process of eliciting system safety requirements along the risk management process, system safety requirements are classified into three categories, i.e., system-level safety requirements, derived safety requirements, and safety design requirements. System-level safety requirements correspond to system-level hazards that can be elicited based on experts' knowledge. System-level hazards are system states that, under some worst-case environmental conditions, will result in undesired events [39]. Then, hazard analysis is adopted to identify the causes of system-level hazards. The causes are unsafe control actions of system components that can be taken as controllers. Hazard analysis can be achieved using the System-Theoretic Process and Analysis (STPA) [26], [39]. The reasons are two-fold. First, STPA is a new hazard analysis technique. Second, STPA complies with the philosophy that system safety is a control problem. Therefore, it provides a way to identify unsafe control actions that cause system-level hazards. The estimation and evaluation of the risk can determine whether the risk has been reduced to an acceptable level by considering the causes. Thus, the derived safety requirements correspond to unsafe control actions. Risk reduction is the systematic process of reducing risks [9]. It can be a design of sub-systems or a redesign of the original system to eliminate, mitigate, or monitor the identified hazards. The input to risk reduction can be the final result of the STPA analysis, that is, the causes of unsafe control actions. The safety design requirements are the design constraints that relate to the causes of unsafe control actions.

Other elements of the requirements hierarchy, shown in Table 3, can also be generated during the risk management process. First, the risk management process results, i.e., system-level hazards, unsafe control actions, and their causes, also have context and the necessity of existence. These can be transformed into requirement context and justification of safety requirements. Second, one or more unsafe control actions can cause a system-level hazard, and several causal factors will cause an unsafe control action. These relationships can be satisfaction relationships. Third, the causes of unsafe control actions can be utilized to propose design solutions. The solution can be the scheme in the requirements hierarchy.

A map between the system safety requirements and a system with its components is required. First, the system and its components are supposed to satisfy these safety requirements. Second, the system and its components could be the reason for a system to be unsafe. In step IV-E, only related safety requirements must be retrieved based on traceability for further analysis. Third, if the system and its components change, which compromises system safety, the change must be represented in the requirements structure. The system-level safety requirements correspond to the entire HMS system. The derived safety requirements map to the sensing, processing, actuating, communication, and storage components as introduced in Section III-A. The safety design requirements are related to the detailed design of these components.

B. STEP 2: MODEL TRANSFORMATION

This step introduces the model transformation rules to build BN from a safety case that is transformed from the system safety requirements structure. Model transformation is the creation of a new model based on another model manually or facilitated by some tools [43]. The transformations are between two mappings. One is between the system safety requirements structure and the safety case. Another is between the safety case and the BN.

1) TRANSFORMING FROM THE REQUIREMENTS STRUCTURE INTO A SAFETY CASE

As discussed in Section III-B, the safety case in this paper is represented by the GSN. Therefore, this section will introduce the mapping between the system safety requirements structure and the GSN, with their transformation rules. The mapping is based on the semantic meanings of related terms. The mapping and explanation are shown in Table 4.

The transformation rules are the following steps. It builds the safety case in a top-down fashion from the requirements structure. The goals and requirements form the skeletons of the structures of GSN and the requirements, respectively. So, other elements can be easily transformed if the skeleton is transformed. The transformation rules applied to a system safety requirements structure will produce a safety case.

1) Transform the system-level safety requirement and its connecting nodes into the top-level goal and the nodes that connect to it, respectively.
   a) Transform the system-level safety requirement into the top-level goal.
   b) Transform the nodes connecting to the system safety requirement into the nodes connecting to the top-level goal based on the mapping in Table 4. Justification and context may need to be transformed here. In this case, the connections between these nodes and the system safety requirement need to be transformed into the InContextOf relationship of the GSN structure.
   c) Transform the satisfy connection between the system-level safety requirement and its adjacent node below into the SupportedBy relationship.
2) Transform the derived safety requirements and their connecting nodes into intermediate-level goals and nodes that connect them.
   a) Transform a derived safety requirement into a corresponding intermediate-level goal.
   b) Transform the nodes connecting to the derived safety requirement into nodes that connect to the intermediate-level goal based on the mapping in Table 4. Justification and context may need to be transformed in this step. The connections between these nodes and the derived safety requirements should be transformed into the InContextOf relationship.
   c) Transform the satisfy connection between the derived safety requirement and its adjacent nodes below into a SupportedBy relationship.
   d) If there are derived safety requirements that are not transformed, then go to step 2a.
3) Transform the scheme nodes into solution nodes.
   a) Transform a scheme node into a solution node.
   b) If there are scheme nodes that are not transformed, go to step 3a.

2) TRANSFORMING FROM THE SAFETY CASE INTO BN

The causal relationships of the nodes in a safety case enable the transformation into BN. In the GSN structure, every goal is supported by sub-goals or solutions. The goal holds only if its supporting sub-goals or solutions, together with the rationale of the supportiveness, i.e., the strategy hold. Context, assumption, and justification can be taken as premises in order for a goal or strategy to hold. The transformation introduced in this section is based on the mapping approach discussed in [10]. The main work in this step consists of transforming the safety case structure into a BN structure and probability elicitation.

TABLE 4. The mapping between system safety requirements and GSN elements.

Mapping elements		Explanation
GSN elements	System safety requirements elements	
Goal	Requirement	Different levels of goals correspond to the different levels of requirements. A goal is a requirement based on the definition of the goal.
Strategy	Requirement justification	Both represent the rationale for items at the child level to support items at the parent level.
Assumption	Requirement context	The assumed portion in the requirement context.
Justification	Requirement justification	Justification of goals and strategies both correspond to the requirement justification.
Context	Requirement context	Goal and strategy context corresponds to the context of the requirement.
Solution	Scheme	Solution means to provide evidence for a goal. The scheme is to provide evidence for a requirement.

To enable structural transformation, the nodes of the safety case must be parameterized. Parameters can be used as the node variables in the BN. The mapping is illustrated in Table 5. The state space of the variables includes *hold* and *notHold*. *hold* means that the corresponding GSN element is satisfied. The transformation follows a top-down fashion, i.e., from the top goal down to the evidence along the SupportedBy connections. The rules below can be used to achieve the transformation.

TABLE 5. The mapping between GSN elements and variables of BN.

GSN elements	Variables of BN	State space
Goal	V_g	hold, notHold
Strategy	V_s	hold, notHold
Context	V_c	hold, notHold
Assumption	V_a	hold, notHold
Justification	V_j	hold, notHold
Solution	V_e	hold, notHold

1) Transform the top-level goal into V_g.
   a) Transform the supporting elements of the top-level goal into causal factors of V_g, that is, transform the sub-goals, strategy, context, and justification that connect to the top-level goal into $\{V_g^{s1}, V_g^{s2}, \dots\}$, V_s, V_c, and V_j, respectively. Then connect $\{V_g^{s1}, V_g^{s2}, \dots\}$, V_s, V_c, and V_j to V_g. If the strategy has context and justification that connect to it, then transform them into V_c' and V_j', respectively. And connect V_c' and V_j' to V_s as causal factors.
2) Transform an intermediate-level goal into V_g^i.
   a) Transform the supporting elements of the intermediate level goal into causal factors of V_g^i, that is, its sub-goals, strategy, context, and justification that connect to the intermediate level goal into $\{V_g^{si1}, V_g^{si2}, \dots\}$, V_s^i, V_c^i, and V_j^i, respectively. Then connect $\{V_g^{si1}, V_g^{si2}, \dots\}$, V_s^i, V_c^i, and V_j^i to V_g^i. If the strategy has context and justification that connect to it, then transform them into $V_c^{i'}$ and $V_j^{i'}$, respectively. And connect $V_c^{i'}$ and $V_j^{i'}$ to V_s^i as causal factors.
   b) If there exist intermediate-level goals that have not been transformed, then go to step 2.
3) Transform a bottom-level goal into V_g^b.
   a) Transform the supporting elements of the bottom-level goal into causal factors of V_g^b, that is, the solution, strategy, context, and justification that connect to the bottom-level goal into V_e^b, V_s^b, V_c^b, and V_j^b, respectively. Then connect V_e^b, V_s^b, V_c^b, and V_j^b to V_g^b. If the strategy has context and justification that connect to it, then transform them into $V_c^{b'}$ and $V_j^{b'}$, respectively. And connect $V_c^{b'}$ and $V_j^{b'}$ to V_s^b as causal factors.
   b) If there are bottom goals that have not been transformed, go to step 3.

There are two categories for probability elicitation [41], [42], [44]. The first category is the judgment of experts assisted by, for example, probability-scale methods, gamble-like methods, and probability-wheel methods. This category heavily depends on expert knowledge and empirical experience to elicit related probabilities by marking them on like scales. The second category is learning from related data sets. The quality of the data set, for example, whether missing values exist and the size of the data set, is critical.

The first category may be more feasible for the probability elicitation of this paper. On the one hand, the BN built in this paper is transformed from a safety case that is transformed from the system safety requirements structure. Experts who developed the system safety requirements structure may be capable of eliciting the probabilities assisted by tools or BN experts. On the other hand, the data set used to learn the probabilities is difficult to obtain. To the best of the author's knowledge, there are no such and similar data sets in the literature. Even if there is a small amount of data, it can be a non-trivial work to apply the data in this paper due to different systems having different design goals and working environments.

According to the characteristics of the safety case structure and the transformed BN, the node variables in the BN corresponding to the context, assumption, justification, and solution are the leaf nodes. Their probabilities in the *hold* state may depend on the conformity of their description to related facts. The node variable in the BN corresponding to a strategy in the *hold* state is conditioned on variables related to the context and assumption. The node variable in the BN corresponding to a goal in the *hold* state is conditioned on the variables that correspond to its sub-goals, solutions, context, justification, assumption, and strategy.

C. STEP 3: SYSTEM SAFETY EVALUATION

The system safety can be quantitatively evaluated on the basis of the BN obtained in the previous step. To this end, evidence must be gathered first to evaluate system safety. Then assess to determine whether the evaluation result is acceptable or not.

The evidence in the BN is the concrete state of a node variable. For example, a Boolean variable V denotes all relevant hazards that have been identified. The evidence can be $V = true$ or $V = false$. To determine the state of V, the answer to the question, for example, *are identified hazards relevant and comprehensive?* must be provided with proof. Not all variables' states can be determined. The more evidence gathered, the more reliable the evaluation result will be.

Generally, safety can be concluded when related risk has been reduced to an acceptable or tolerable level. Risk is the combination of two parts [9], [45], that is, the probability of the occurrence of an unsafe event and the severity of the consequence of the event. Therefore, safety levels can be defined in terms of the above two factors. A prominent and widely used benchmark is the safety integrity level (SIL), as introduced in IEC 61508 [45]. The safety integrity is ''the probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time''. The rationale for evaluating safety can also be applied to assess the BN result. The reason is that safety is obtained when all safety requirements are met. System safety requirements are defined where hazards must be eliminated, mitigated, or monitored considering the risk the hazards raised, as introduced in step one (Section IV-A). Risk can be reduced to an acceptable or tolerable level when the safety requirements of the system are met. So, the SIL can be applied in this paper. However, the SIL may need to be customized to fit the context of HMS systems. Methods to determine the SIL are not introduced, as they are beyond the scope of this paper.

D. STEP 4: TRACEABILITY METAMODEL

Traceability is required to solve the problems discussed in steps 5.1 and 5.2, that is, to retrieve related system safety requirements to improve system safety; and to propagate the changes to system safety requirements and then to the BN for re-evaluation. Two things have to be done to enable traceability, that is, artifacts can be traced and links between artifacts. Since the instances of the system safety requirements structure, the safety case, and the BN vary. In this paper, a traceability metamodel is proposed. A metamodel is generally defined as a model used to build a model [46], [47]. Therefore, no matter what instances of the system safety requirements structure, the safety case, and the BN are, the traceability between them must comply with the rules defined in the metamodel.

The artifacts to be traced include three categories, i.e., elements of system safety requirements and GSN notation, as well as node variables of BN. For the traceability of this paper, the link is to connect artifacts that have mapping relations. The metamodel is depicted in Figure 3. There are three columns in the figure. The left column corresponds to the elements of the system safety requirements. The middle column relates to elements of the GSN notation. The right column corresponds to the node variables of the BN. The connection between two artifacts is represented by the association relation of the class of the unified modeling language. The association is to connect classes that have relationships. For simplicity, the details of the classes are omitted. The relationships between classes of each column are not shown in the figure, since they are irrelevant to traceability.

E. STEP 5.1: REQUIREMENTS RETRIEVAL

Once the safety level of the HMS system is evaluated as unsafe by the BN, the system safety needs to be improved. To this end, it first identifies the variable nodes that are in *notHold* state and have a high influence on the target node of the BN. In other words, a unit variation in these variables causes more variation in the target node variable than others. Then trace these variable nodes to related system safety requirements based on the traceability. Finally, analyze the reason for the safety requirements in *notHold* state to improve the safety of the system.

To identify the variables that have a high influence on the target node variable, sensitivity analysis [24] can be applied. In the field of BN, it studies the uncertainties of random variables in their possible states to affect the probability of the target variable in a specific state [48], [49], [50]. Most BN tools can achieve sensitivity analysis by, for example, setting the target node and clicking on some buttons. For example, the GeNIe modeler provides ways of automatically analyzing sensitivity [51]. It is a graphical tool to assist in building and analyzing based on BNs.

After identifying highly influential variables, trace them to the system safety requirements through the safety case along the traceability links. Then analyze to improve system safety.

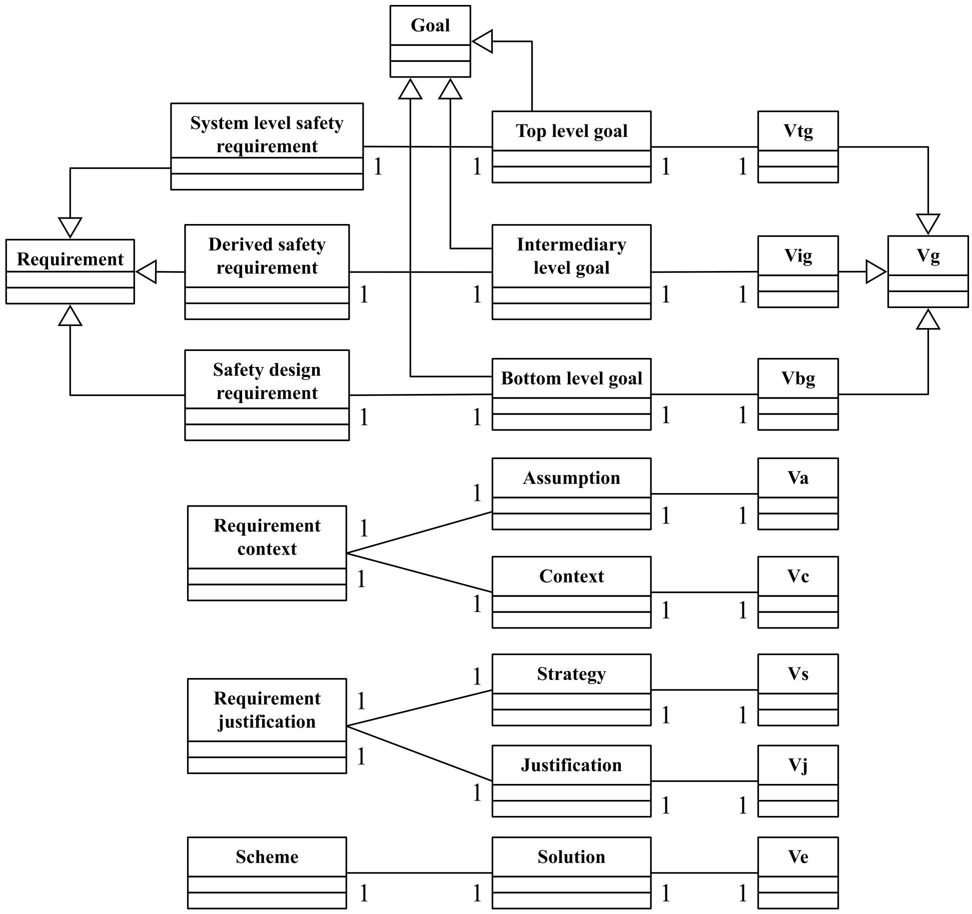


FIGURE 3. The traceability metamodel.

The analysis should be conducted by experienced experts. The following example questions can be used to guide the analysis.

1) Are hazards identified comprehensively?
2) Are hazards correctly analyzed?
3) Are system safety requirements correctly and reasonably elicited based on the hazards?
4) Are system components correctly designed to meet the safety requirements?
5) Have the tools used in hazard identification/analysis, safety requirements elicitation, and system components design compromised system safety?
6) Can the proposed solutions improve system safety?

F. STEP 5.2: CHANGE PROPAGATION

HMS systems are self-adaptive systems that are characterized by the dynamicity of system requirements [20]. Therefore, the system safety requirements can be changed during the system life cycle. The changes have to be evaluated to determine if they affect system safety. If they do, the changes must be propagated to the BN through traceability links for re-evaluation of the system safety.

In requirement engineering, requirements can be maintained, for example, via the version control or evolution link [30], [31], [52]. Thus, changes to the system safety requirements can be captured by comparing two versions of the system safety requirements or by tracing the requirements changes through evolution links. To evaluate the changes, one can always check whether the system-level hazards, unsafe control actions, and the causes of unsafe control actions are still taken care of by the changed safety requirements, since a system is safe when all hazards are under control.

After the changes in the safety requirements are determined, they are about to be propagated to the safety case and then to the BN. To this end, the mappings and transformation rules introduced in Section IV-B can be used. It has to transform the changes to the corresponding elements of the safety case first. Then transform the changes in the safety case to the related part of BN. Be aware that the conditional probabilities of the changed node variables in different states may also need to be reelicited. Finally, a re-evaluation of the system safety can be performed.

V. APPLICATION EXAMPLE

This section adopts an example HMS to demonstrate the application of the proposed framework so as to discuss the contribution of this paper. To this end, this section first introduces an example HMS system. Then apply the

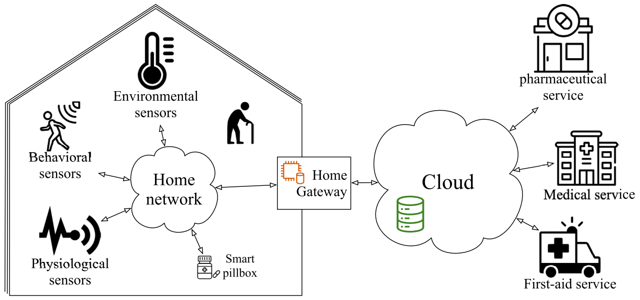


FIGURE 4. Architecture of the example HMS system.

proposed framework to quantitatively evaluate the safety of the example system. Finally, to demonstrate requirements retrieval and change propagation.

A. AN EXAMPLE HMS SYSTEM

The example HMS system introduced in this paper is revised based on the one in [53]. It gathers environmental, behavioral, and physiological data through various sensors. Then analyze them to determine if occupants that are assumed to have chronic disease need to change their medication or call emergency service. The architecture of the example system is illustrated in Figure 4. In this example, two scenarios are considered. One is to change the current or to deliver a new medicine. Another is to call first-aid service. The data is acquired in the home environment and stored locally in the home gateway and in the cloud. The home gateway can do preliminary analysis, and the medical service is responsible for comprehensively analyzing the data. If there is a need to change the current or to deliver new medication, the gateway and the medical service can send commands to the pharmaceutical service. The pharmaceutical service can inform the information about the medicines through the smart pillbox to occupants. If an emergency is necessary, they can command the first-aid service to send an ambulance.

B. QUANTITATIVE EVALUATION OF THE EXAMPLE SYSTEM

This section introduces the application of the proposed framework to evaluate the safety of the example HMS system. First, the system safety requirements structure is built based on the risk management process. Then the safety case represented by GSN notation is transformed from the requirements structure. Third, transform the safety case into BN for quantitative safety evaluation.

To build the system safety requirements structure, hazards and their causal factors are required to be identified. This is achieved by applying the STPA technique. The first step is to define the undesired safety event and system-level hazards and requirements, as shown in Table 6. Then to induce unsafe control actions based on the control structure illustrated in Figure 5. Unsafe control actions and their causal factors are shown in Table 7. Be aware that the example is to demonstrate the application of the proposed framework and not to elicit comprehensive unsafe control actions and their causal factors.

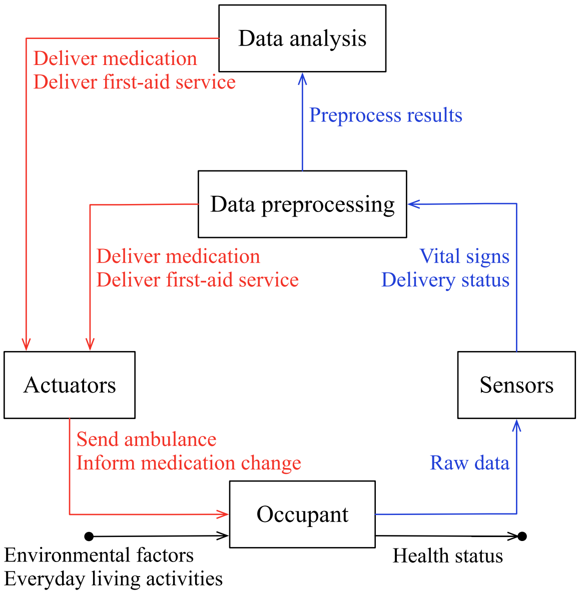


FIGURE 5. Control structure of the HMS system example for eliciting unsafe control actions. The red arrows represent the control directions, and the text attached to them is the control actions. The blue arrows denote the feedback.

System safety requirements are then derived from system-level hazards, unsafe control actions, and their causal factors. The requirements structure is illustrated by the requirement diagram of SysML as shown in Figure 6. For compliance with the relationship between system safety requirements introduced in section III-C, the satisfy relationship of SysML is adopted to represent the relationship between different levels of requirements. Though the meaning of satisfy relationship may not be exactly the same as in SysML.

Then transform the system safety requirements structure into the safety case that is represented by the GSN notation as shown in Figure 7. The transformation takes the requirements structure and the transformation rules introduced in Section IV-B as input. Finally, the safety case is transformed into the BN based on the transformation rules introduced in Section IV-B. The transformed BN is illustrated in Figure 8. The author elicited the probabilities of the node variables of the BN based on the probability-scale method [41].

After obtaining the BN, it can be used to evaluate the system safety. This paper utilizes the GeNIe modeler for building and evaluating based on the BN. GeNIe is an interactive tool for model building and learning. The node $V_g\_G\_SR$ in Figure 8 represents the safety state of the HMS system. For demonstration purposes, the state of node $V_g\_G1\_1$ is set to *notHold* and the probability of node $V_g\_G\_SR$ in state *hold* is 78.24%. This means that the example HMS system is evaluated as 78.24% safe.

To determine whether 78.24% represents safe or unsafe, the SIL should be set up. IEC 61508 [45] introduced steps and suggested techniques to determine the SIL. The SIL, for example, can have four levels, e.g., safe, tolerable, tolerable after improvement, and intolerable. Each level corresponds to

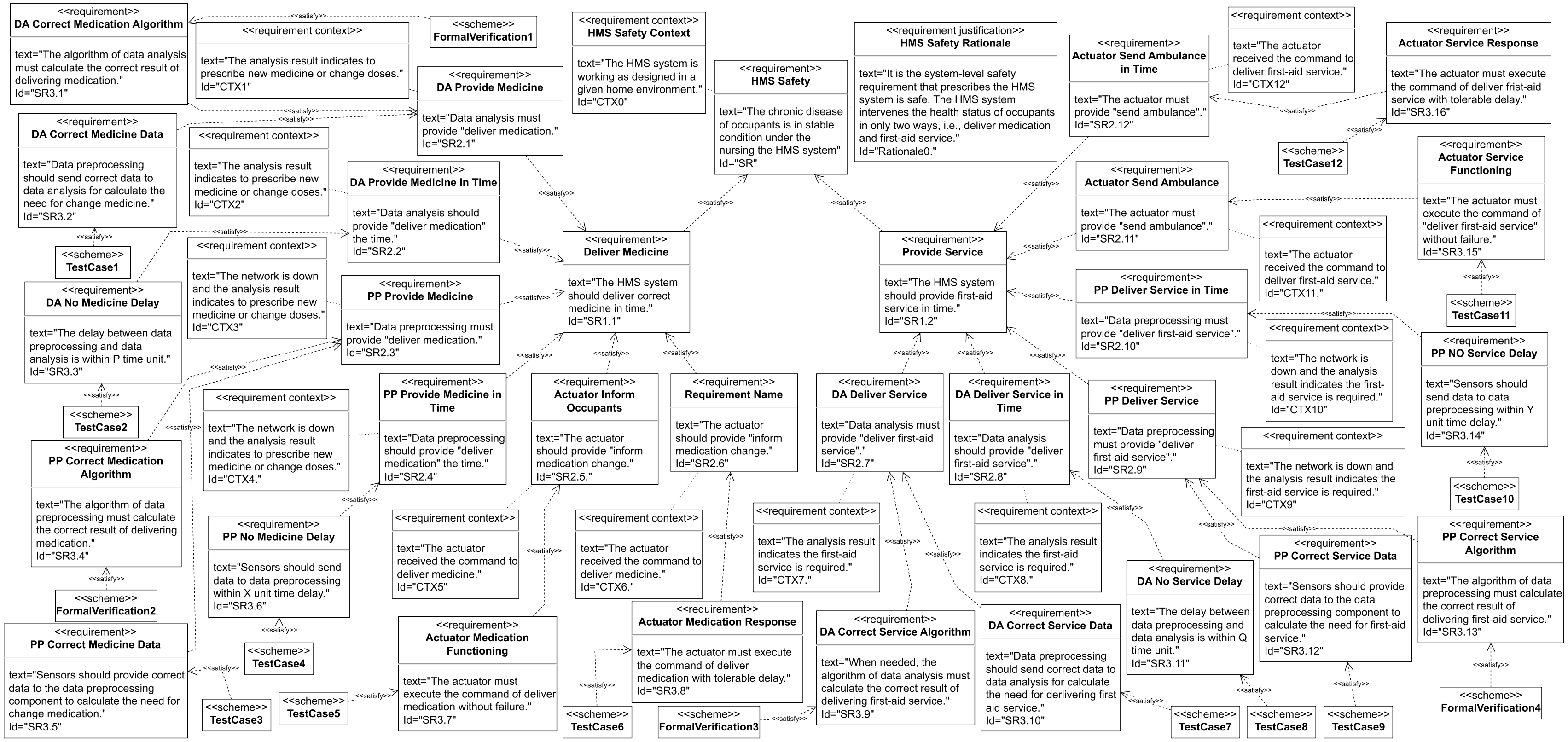

<<requirement>> DA Correct Medication Algorithm text="The algorithm of data analysis must calculate the correct result of delivering medication." Id="SR3.1"
<<requirement context>> text="The analysis result indicates to prescribe new medicine or change doses." Id="CTX1"
<<scheme>> FormalVerification1
<<requirement context>> HMS Safety Context text="The HMS system is working as designed in a given home environment." Id="CTX0"
<<requirement>> HMS Safety text="The chronic disease of occupants is in stable condition under the nursing the HMS system" Id="SR"
<<requirement justification>> HMS Safety Rationale text="It is the system-level safety requirement that prescribes the HMS system is safe. The HMS system intervenes the health status of occupants in only two ways, i.e., deliver medication and first-aid service." Id="Rationale0."
<<requirement>> Actuator Send Ambulance in Time text="The actuator must provide "send ambulance"." Id="SR2.12"
<<requirement context>> text="The actuator received the command to deliver first-aid service." Id="CTX12"
<<requirement>> Actuator Service Response text="The actuator must execute the command of deliver first-aid service with tolerable delay." Id="SR3.16"
<<requirement>> DA Provide Medicine text="Data analysis must provide "deliver medication." Id="SR2.1"
<<requirement>> DA Correct Medicine Data text="Data preprocessing should send correct data to data analysis for calculate the need for change medicine." Id="SR3.2"
<<requirement context>> text="The analysis result indicates to prescribe new medicine or change doses." Id="CTX2"
<<requirement>> DA Provide Medicine in TIme text="Data analysis should provide "deliver medication" the time." Id="SR2.2"
<<scheme>> TestCase12
<<requirement>> Actuator Send Ambulance text="The actuator must provide "send ambulance"." Id="SR2.11"
<<requirement>> Actuator Service Functioning text="The actuator must execute the command of "deliver first-aid service" without failure." Id="SR3.15"
<<scheme>> TestCase1
<<requirement>> Deliver Medicine text="The HMS system should deliver correct medicine in time." Id="SR1.1"
<<requirement>> Provide Service text="The HMS system should provide first-aid service in time." Id="SR1.2"
<<requirement context>> text="The actuator received the command to deliver first-aid service." Id="CTX11."
<<requirement>> DA No Medicine Delay text="The delay between data preprocessing and data analysis is within P time unit." Id="SR3.3"
<<requirement context>> text="The network is down and the analysis result indicates to prescribe new medicine or change doses." Id="CTX3"
<<requirement>> PP Provide Medicine text="Data preprocessing must provide "deliver medication." Id="SR2.3"
<<requirement>> PP Deliver Service in Time text="Data preprocessing must provide "deliver first-aid service"." Id="SR2.10"
<<scheme>> TestCase11
<<requirement context>> text="The network is down and the analysis result indicates the first-aid service is required." Id="CTX10"
<<requirement>> PP NO Service Delay text="Sensors should send data to data preprocessing within Y unit time delay." Id="SR3.14"
<<scheme>> TestCase2
<<requirement context>> text="The network is down and the analysis result indicates to prescribe new medicine or change doses." Id="CTX4."
<<requirement>> PP Provide Medicine in Time text="Data preprocessing should provide "deliver medication" the time." Id="SR2.4"
<<requirement>> Actuator Inform Occupants text="The actuator should provide "inform medication change." Id="SR2.5."
<<requirement>> Requirement Name text="The actuator should provide "inform medication change." Id="SR2.6"
<<requirement>> DA Deliver Service text="Data analysis must provide "deliver first-aid service"." Id="SR2.7"
<<requirement>> DA Deliver Service in Time text="Data analysis should provide "deliver first-aid service"." Id="SR2.8"
<<requirement>> PP Deliver Service text="Data preprocessing must provide "deliver first-aid service"." Id="SR2.9"
<<requirement context>> text="The network is down and the analysis result indicates the first-aid service is required." Id="CTX9"
<<scheme>> TestCase10
<<requirement>> PP Correct Medication Algorithm text="The algorithm of data preprocessing must calculate the correct result of delivering medication." Id="SR3.4"
<<requirement>> PP No Medicine Delay text="Sensors should send data to data preprocessing within X unit time delay." Id="SR3.6"
<<requirement context>> text="The actuator received the command to deliver medicine." Id="CTX5"
<<requirement context>> text="The actuator received the command to deliver medicine." Id="CTX6."
<<requirement context>> text="The analysis result indicates the first-aid service is required." Id="CTX7."
<<requirement context>> text="The analysis result indicates the first-aid service is required." Id="CTX8."
<<requirement>> PP Correct Service Data text="Sensors should provide correct data to the data preprocessing component to calculate the need for first-aid service." Id="SR3.12"
<<requirement>> PP Correct Service Algorithm text="The algorithm of data preprocessing must calculate the correct result of delivering first-aid service." Id="SR3.13"
<<requirement>> DA No Service Delay text="The delay between data preprocessing and data analysis is within Q time unit." Id="SR3.11"
<<scheme>> FormalVerification2
<<requirement>> Actuator Medication Functioning text="The actuator must execute the command of deliver medication without failure." Id="SR3.7"
<<requirement>> Actuator Medication Response text="The actuator must execute the command of deliver medication with tolerable delay." Id="SR3.8"
<<requirement>> DA Correct Service Algorithm text="When needed, the algorithm of data analysis must calculate the correct result of delivering first-aid service." Id="SR3.9"
<<requirement>> DA Correct Service Data text="Data preprocessing should send correct data to data analysis for calculate the need for derlivering first aid service." Id="SR3.10"
<<requirement>> PP Correct Medicine Data text="Sensors should provide correct data to the data preprocessing component to calculate the need for change medication." Id="SR3.5"
<<scheme>> TestCase4
<<scheme>> TestCase3
<<scheme>> TestCase5
<<scheme>> TestCase6
<<scheme>> FormalVerification3
<<scheme>> TestCase7
<<scheme>> TestCase8
<<scheme>> TestCase9
<<scheme>> FormalVerification4
<<satisfy>>

FIGURE 6. System safety requirements structure documented by the requirement diagram of SysML.

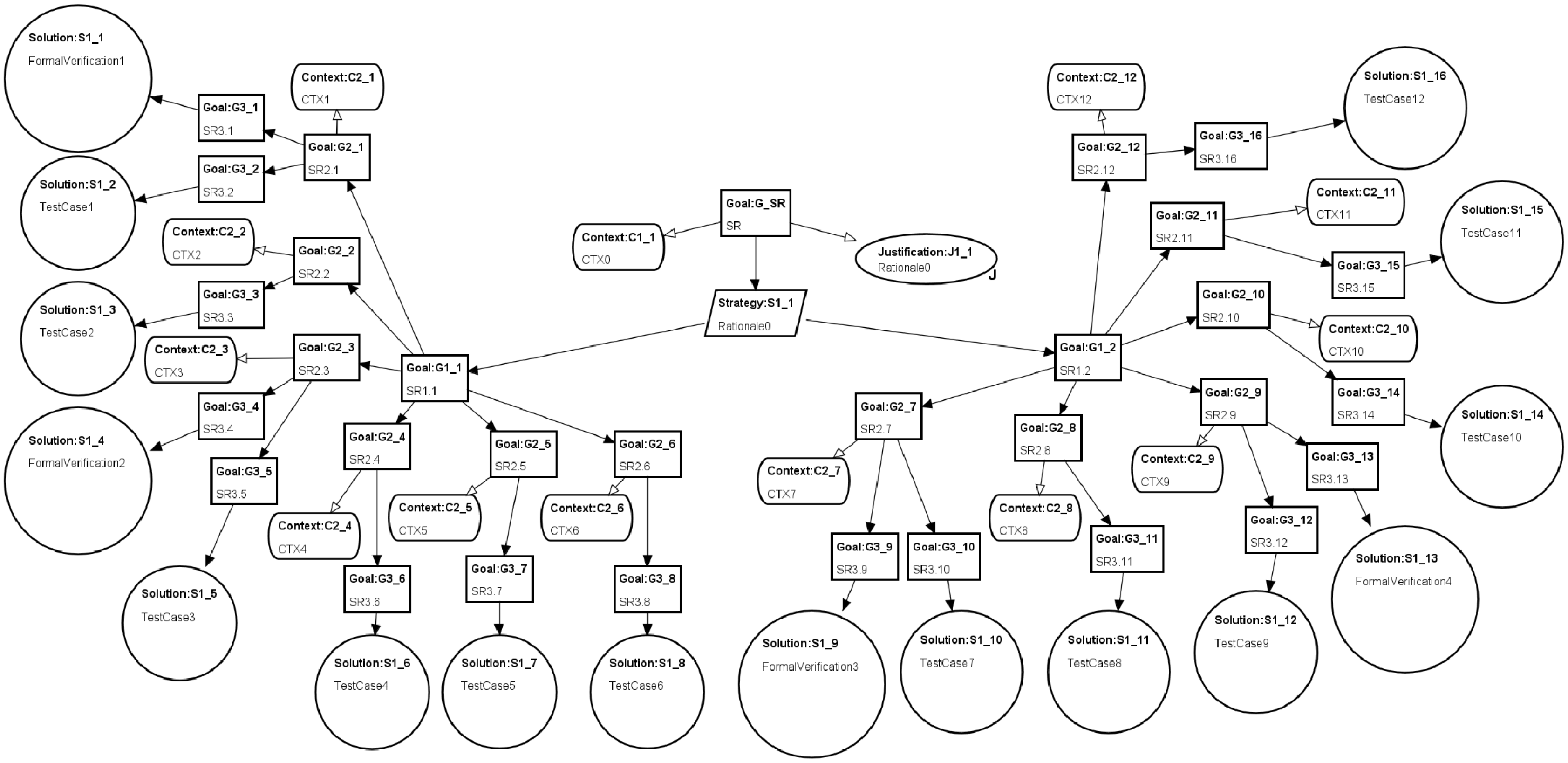


FIGURE 7. GSN represented safety case which transformed from the safety requirements structure.

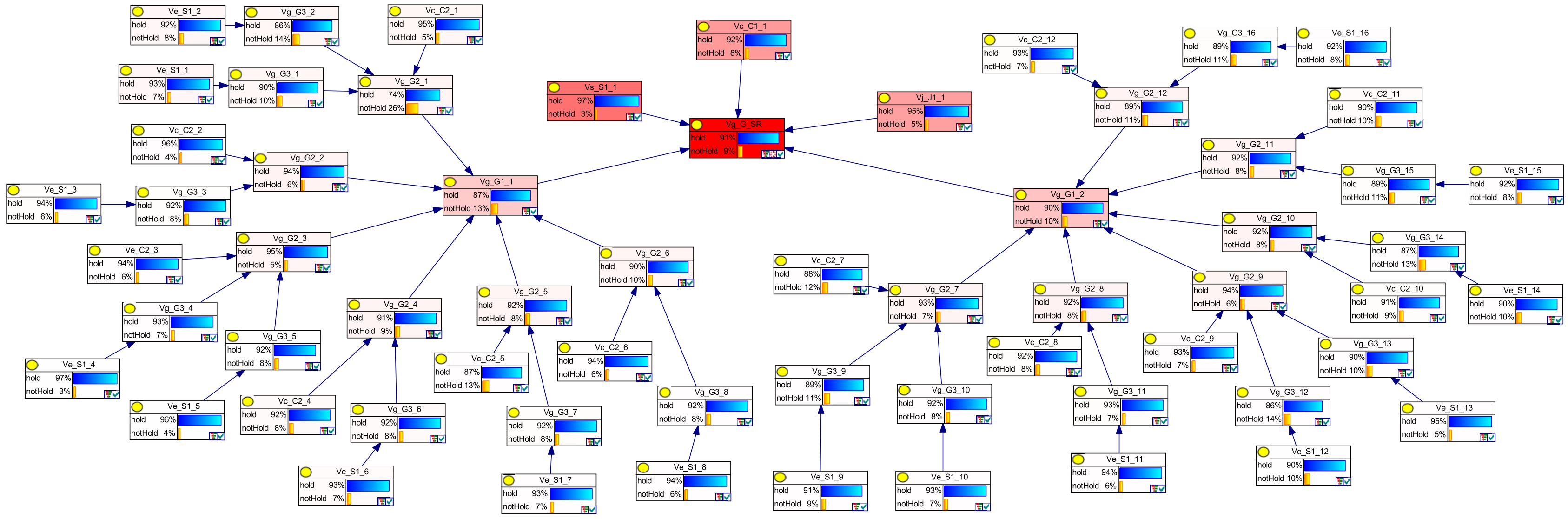

Ve_S1_2 hold 92% notHold 8%
Vg_G3_2 hold 86% notHold 14%
Vc_C2_1 hold 95% notHold 5%
Ve_S1_1 hold 93% notHold 7%
Vg_G3_1 hold 90% notHold 10%
Vg_G2_1 hold 74% notHold 26%
Vc_C2_2 hold 96% notHold 4%
Vg_G2_2 hold 94% notHold 6%
Ve_S1_3 hold 94% notHold 6%
Vg_G3_3 hold 92% notHold 8%
Vg_G1_1 hold 87% notHold 13%
Vs_S1_1 hold 97% notHold 3%
Vc_C1_1 hold 92% notHold 8%
Vg_G_SR hold 91% notHold 9%
Vj_J1_1 hold 95% notHold 5%
Vc_C2_3 hold 94% notHold 6%
Vg_G2_3 hold 95% notHold 5%
Vg_G3_4 hold 93% notHold 7%
Ve_S1_4 hold 97% notHold 3%
Vg_G3_5 hold 92% notHold 8%
Ve_S1_5 hold 96% notHold 4%
Vc_C2_4 hold 92% notHold 8%
Vg_G2_4 hold 91% notHold 9%
Vg_G3_6 hold 92% notHold 8%
Ve_S1_6 hold 93% notHold 7%
Vg_G2_5 hold 92% notHold 8%
Vc_C2_5 hold 87% notHold 13%
Vg_G3_7 hold 92% notHold 8%
Ve_S1_7 hold 93% notHold 7%
Vg_G2_6 hold 90% notHold 10%
Vc_C2_6 hold 94% notHold 6%
Vg_G3_8 hold 92% notHold 8%
Ve_S1_8 hold 94% notHold 6%
Vc_C2_7 hold 88% notHold 12%
Vg_G2_7 hold 93% notHold 7%
Vg_G3_9 hold 89% notHold 11%
Ve_S1_9 hold 91% notHold 9%
Vg_G3_10 hold 92% notHold 8%
Ve_S1_10 hold 93% notHold 7%
Vg_G1_2 hold 90% notHold 10%
Vg_G2_8 hold 92% notHold 8%
Vc_C2_8 hold 92% notHold 8%
Vg_G3_11 hold 93% notHold 7%
Ve_S1_11 hold 94% notHold 6%
Vg_G2_9 hold 94% notHold 6%
Vc_C2_9 hold 93% notHold 7%
Vg_G3_12 hold 86% notHold 14%
Ve_S1_12 hold 90% notHold 10%
Vg_G3_13 hold 90% notHold 10%
Ve_S1_13 hold 95% notHold 5%
Vc_C2_10 hold 91% notHold 9%
Vg_G2_10 hold 92% notHold 8%
Vg_G3_14 hold 87% notHold 13%
Ve_S1_14 hold 90% notHold 10%
Vg_G2_11 hold 92% notHold 8%
Vc_C2_11 hold 90% notHold 10%
Vg_G3_15 hold 89% notHold 11%
Ve_S1_15 hold 92% notHold 8%
Vg_G2_12 hold 89% notHold 11%
Vc_C2_12 hold 93% notHold 7%
Vg_G3_16 hold 89% notHold 11%
Ve_S1_16 hold 92% notHold 8%

FIGURE 8. The BN that transformed from the safety case. The result of sensitivity analysis. Node $V_g\_G\_SR$ is the target node. Nodes in red color are sensitive to the target node.

a probability interval that represents the average probability that the unsafe event to the system has occurred. If the 78.24% were in the tolerable after improvement probability interval, the HMS system is tolerably safe only if the unsafe event has been handled.

C. REQUIREMENTS RETRIEVAL AND CHANGE PROPAGATION

As discussed at the end of the last section, the system is 78.24% safe and needs improvement. The next step is to determine the node variables that are in *notHold* state and are sensitive to the target node variable. Then trace the sensitive nodes to related safety requirements. Finally, analyze the reasons why the safety requirements are not satisfied.

The GeNIe modeler provides the function for sensitive analysis. The result of the analysis is shown in Figure 8. The nodes $V_g\_G1\_1$, $V_s\_S1\_1$, $V_c\_C1\_1$, $V_j\_J1\_1$, and $V_g\_G1\_2$ are sensitive to the target node $V_g\_G\_SR$. The node $V_g\_G1\_1$ is in *notHold* state. Thus, node $V_g\_G1\_1$ should be traced based on the traceability model. The traceability links are embedded between the models of system safety requirements structure, safety case, and the BN. For example, the requirement ID $SR3.4$ is stored at the goal node $G3\_4$ of the safety case. The goal node ID $G3\_4$ is stored at the node $V_g\_G3\_4$ of the BN. So, the requirement $SR3.4$ is retrieved through the traceability link.

The change to requirements can be modification, addition, and deletion. If a requirement is revised or deleted, its child requirements may also need to be modified or deleted. For example, the requirement $SR2.4$ in Figure 6 was modified or deleted, its context $CTX4$, the child requirement $SR3.6$ and the scheme *TestCase*4 are also modified or deleted. Then trace the requirements to nodes $G2\_4$, $C2\_4$, $G3\_6$, and $S1\_6$ of the safety case in Figure 7. Finally, trace these goal nodes to nodes $V_g\_G2\_4$, $V_g\_C2\_4$, $V_g\_G3\_6$, and $V_e\_S1\_6$ of the BN in Figure 8. In this case, not only the conditional probabilities of nodes $V_g\_G2\_4$, $V_g\_C2\_4$, $V_g\_G3\_6$, and $V_e\_S1\_6$ need to be revised, but also the conditional probabilities of nodes on these nodes also need to be revised, e.g., $V_g\_G1\_1$. If a new requirement $SR3.6'$ is added to node $SR2.4$, then trace $SR2.4$ to $G2\_4$ of the safety case and $V_g\_G2\_4$ of the BN. Finally, add a new node $G2\_4'$ to $G2\_4$, and a new node $V_g\_G2\_4'$ to $V_g\_G2\_4$. Whereafter, a reevaluation can be conducted.

VI. DISCUSSION

This section provides a discussion based on the application results. The first is the feasibility of the proposed framework, which proves that the proposed framework has overcome the limitations discussed in Section I. Second, problems need to be solved to apply the proposed framework into practice. Third, the pros and cons of the proposed framework.

A. FEASIBILITY

Safety case consists of layered arguments, in which a parent claim is supported by child claims or evidence. Claims are about the safety requirements of a system, and evidence relates to engineering activities about the system. A system is safe when the top-level claim is satisfied by the lower-level claims and evidence. Generally, it is tolerable if some claims are not supported or have weak evidence support. In this case, the safety of a system depends entirely on the expertise to judge. It may not be straightforward to know the safety of a system. Thus, the safety case is transformed into BN for quantitative safety evaluation. As seen from the result in section V-B and Figure 8, even when the node variable $V_g\_G1\_1$ is in *notHold* state, the possibility of the system being safe can still be obtained.

The transformation from the safety case to the BN depends on the semantic meaning of GSN notation elements and their causal relationships, as discussed in Section IV-D. For example, the goal $G1\_1$ is supported by sub-goals of $G2\_1$, $G2\_2$, $G2\_3$, $G2\_4$, $G2\_5$, and $G2\_6$ in Figure 7. The satisfaction of the sub-goals causes the goal $G1\_1$ to hold. Thus, this relation is represented in the BN in that the state of node $V_g\_G1\_1$ is affected by the states of $V_g\_G2\_1$, $V_g\_G2\_2$, $V_g\_G2\_3$, $V_g\_G2\_4$, $V_g\_G2\_5$, and $V_g\_G2\_6$ as shown in Figure 8.

Quantitative evaluation of the system's safety is not the end. If the evaluation result is not acceptable, safety improvement to the system is required. In the case of this paper, the node variables in state *notHold* are mainly focused. The reason is that the requirements corresponding to these variables are not satisfied. Then a sensitive analysis is applied to look for the node variables which are more sensitive to the target node. The higher sensitive node variables in state *notHold* are traced to related safety requirements. In Figure 8, the node $V_g\_G1\_1$ is more sensitive to the target node $V_g\_G\_SR$ and is in *notHold* state, as introduced in Section V-C. Then trace the node $V_g\_G1\_1$ to the node $G1\_1$ of safety case in Figure 7 and to the requirement $SR1.1$ of Figure 6. Finally, analyze the requirement $SR1.1$ for further improvement.

Changes to safety requirements affect the safety of the system. Re-evaluation is a must when changes have occurred. To re-evaluate the system safety, the safety case and BN must be modified according to the changed requirements. The changes propagated from safety requirements to the BN through safety case is based on the traceability model. The last paragraph of Section V-C demonstrated how modification, addition, and deletion of the requirements propagated to the BN.

B. PROBLEMS

To apply the proposed framework into practice, several problems need to be solved. The problems in this paper refer to the activities that are time-consuming or relatively more difficult. The problems listed here may not exhaustive.

1) Safety requirements elicitation and requirements structure construction.

TABLE 6. Definitions in order to apply the STPA technique.

Unsafe event	System level hazard	System level requirement
The chronic disease deteriorated under the nursing of the HMS system and needs medical interference	The HMS system does not deliver the correct medicine in time. (H1)	The HMS system must deliver the correct medicine in time.
	The HMS system does not provide first-aid service in time. (H2)	The HMS system must provide first-aid service in time.

TABLE 7. Unsafe control actions that can result in system-level hazards and their causal factors.

Hazard ID	Unsafe control actions	Causal factor
H1	The data analysis component does not provide "deliver medication" when the analysis result indicates prescribing new medicine or changing doses.	The data analysis component believed that there is no need to administer medicine based on a flawed control algorithm.
	The data analysis component provides "deliver medication" too late when the analysis result indicates prescribing new medicine or changing doses.	The data analysis received delayed preprocessed data from the data preprocessing component.
	The data preprocessing component does not provide "deliver medication" when the network is down, and the analysis result indicates prescribing new medicine or changing doses.	The data preprocessing component believed that there was no need to administer medicine based on a flawed control algorithm. The data preprocessing component received the wrong data from sensors.
	The data preprocessing component provided "deliver medication" too late when the network is down, and the analysis result indicates prescribing new medicine or changing doses.	The data preprocessing component received delayed data from sensors.
	Actuators do not provide "inform medication change" after receiving the command to deliver medicine.	The command "deliver medication" has not been executed.
	Actuators provide "inform medication change" too late after receiving the command to deliver medicine.	The command "deliver medication" was executed too late.
H2	The data analysis component does not provide "deliver first-aid service" when the analysis result indicates the first-aid service is required.	The data analysis component believed that there is no need to deliver first-aid service based on a flawed control algorithm. The data analysis component received the wrong preprocessed data from the data preprocessing component.
	The data analysis component provided "deliver first-aid service" too late when the analysis result indicates the first-aid service is required.	The data analysis component received delayed preprocessed data from the data preprocessing component.
	The data preprocessing component does not provide "deliver first-aid service" when the network is down, and the analysis result indicates the first-aid service is required.	The data preprocessing component believed that there was no need to deliver first-aid service based on the flawed control algorithm. The data preprocessing component received the wrong data from sensors.
	The data preprocessing component provides "deliver first-aid service" too late when the network is down, and the analysis result indicates that the first-aid service is required.	The data preprocessing component received delayed data from sensors.
	Actuators do not provide "send ambulance" after receiving the command to deliver first-aid service.	The command "deliver first-aid service" has not been executed.
	Actuators provide "send ambulance" too late after receiving the command to deliver first-aid service.	The command "deliver first-aid service" was executed too late.

- Safety requirements are elicited based on identified system hazards. Both requirements elicitation and identification of system hazards are labor-intensive and thus error-prone. There are only 31 requirements in the application example, which would be large in real projects. They and the requirements structure construction consumed roughly half the time the author applied the proposed framework to the example HMS system. Luckily, there are tools and formal procedures, e.g., STPA and SysML, that can assist in doing this work.

2) Transform the safety requirements structure into the safety case and the BN.

- The transformation is conducted by hand since there are only 31 requirements, and the complexity is manageable manually. If the requirements are large and the complexity is unmanageable manually, there must be a way to handle the problem. One possible solution could be model-driven engineering [32].

3) Probability elicitation of node variables of BN.
   - Probability elicitation is a time-consuming process in applying BN, especially by experts. First, the number of conditional probabilities to be elicited is large. Second, elicitation always comes with biases. Nearly 30% of the time the author spent was doing the elicitation.
4) Traces between requirements and BN through safety case.
   - Traceability is still a manual process in the application example. Therefore, it is prone to errors. If the requirements become large, traceability could be unmanageable. If the trace process is automated, the problem could be solved.
5) Identification of what requirements have been changed and assessment of whether they can affect the safety of the system.
   - As discussed in Section IV-F, changes can be identified by techniques of requirements maintenance. However, to assess whether changed requirements will affect system safety must be a manual process.

C. PROS AND CONS

This section introduces the advantages and disadvantages of the proposed framework.

The advantages are as follows:

1) It can quantitatively evaluate the safety of an HMS system.
2) It can trace related requirements for improving system safety when a system is evaluated unsafe.
3) Changes to requirements can be propagated to the BN in order to re-evaluate the system safety.

The disadvantages could be that:

1) Some work is done manually, e.g., the process of transforming safety case into BN. So, it may be time-consuming and error-prone.
2) The accuracy of the evaluation depends on the probabilities elicited for the BN. Probability elicitation could only be performed manually. The data used to learn the probabilities seem infertile. The quality of the BN may not be easily managed.

VII. CONCLUSION

This paper proposes a framework for quantitatively evaluating the safety of an HMS system. Compared with the conventional way of system safety evaluation, i.e., safety cases, the contributions are as follows. First, it enables the quantitative evaluation of system safety by transforming the safety case into a BN. Second, it can trace from an evaluation result to related safety requirements to improve system safety. Traceability depends on trace links between system safety requirements, safety cases, and BN. Third, changes to safety requirements can be propagated through trace links to the BN to re-evaluate the system's safety. The application results of the proposed framework to an example HMS system illustrate these advantages.

There are some work can be done in the future, for example,

- Automate some work of the proposed framework, e.g., transformation from requirements structure to safety case and the BN.
- Methods to determine the SIL that fits the HMS system environment are used to assess the evaluation results.
- The approach or procedure for eliciting the probability of BN aims to ensure the evaluation accuracy.

REFERENCES

[1] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni, "A survey on health monitoring systems for health smart homes," *Int. J. Ind. Ergonom.*, vol. 66, pp. 26–56, Jul. 2018.

[2] A. I. Paganelli, A. G. Mondéjar, A. C. da Silva, G. Silva-Calpa, M. F. Teixeira, F. Carvalho, A. Raposo, and M. Endler, "Real-time data analysis in health monitoring systems: A comprehensive systematic literature review," *J. Biomed. Informat.*, vol. 127, Mar. 2022, Art. no. 104009.

[3] I. Habli, S. White, M. Sujan, S. Harrison, and M. Ugarte, "What is the safety case for health IT? A study of assurance practices in England," *Saf. Sci.*, vol. 110, pp. 324–335, Dec. 2018.

[4] F. Magrabi, M. Baker, I. Sinha, M.-S. Ong, S. Harrison, M. R. Kidd, W. B. Runciman, and E. Coiera, "Clinical safety of England's national programme for IT: A retrospective analysis of all reported safety events 2005 to 2011," *Int. J. Med. Informat.*, vol. 84, no. 3, pp. 198–206, Mar. 2015.

[5] S. E. Kjeldsen, "Hypertension and cardiovascular risk: General aspects," *Pharmacological Res.*, vol. 129, pp. 95–99, Mar. 2018.

[6] *Road Vehicles—Functional Safety—Part 1: Vocabulary*, International Organization for Standardization, Geneva, Switzerland, ISO/TC 22/SC 32, ISO Standard 26262-1:2018, 2018. [Online]. Available: https://www.iso.org/standard/68383.html

[7] J. Spriggs, *The Goal Structuring Notation: A Structured Approach to Presenting Arguments*. London, U.K.: Springer, 2012, doi: 10.1007/978-1-4471-2312-5.

[8] *Goal Structuring Notation Community Standard Version 3*, SCSC Assurance Case Working Group, Safety-Critical Systems Club, CA, USA, GSN Community Standard SCSC-141C, 2021.

[9] *Safety Management Requirements for Defence Systems—Part 1: Requirements & Part 2: Guidance on Establishing a Means of Complying with Part 1*, Ministry of Defence, London, U.K., Standard MoD 00-56, 2007.

[10] D. Nešić, M. Nyberg, and B. Gallina, "A probabilistic model of belief in safety cases," *Saf. Sci.*, vol. 138, Jun. 2021, Art. no. 105187.

[11] P. J. Graydon and C. M. Holloway, "An investigation of proposed techniques for quantifying confidence in assurance arguments," *Saf. Sci.*, vol. 92, pp. 53–65, Feb. 2017.

[12] E. Denney, G. Pai, and I. Habli, "Towards measurement of confidence in safety cases," in *Proc. Int. Symp. Empirical Softw. Eng. Meas.*, Sep. 2011, pp. 380–383.

[13] X. Zhao, D. Zhang, M. Lu, and F. Zeng, "A new approach to assessment of confidence in assurance cases," in *Proc. Comput. Saf., Rel., Secur.*, F. Ortmeier and P. Daniel, Eds. Berlin, Germany: Springer, 2012, pp. 79–91.

[14] J. Pearl, "Markov and Bayesian networks: Two graphical representations of probabilistic knowledge," in *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. San Francisco, CA, USA: Morgan Kaufmann, 1988, ch. 3, pp. 77–141.

[15] T. Koski and J. Noble, *Bayesian Networks: An Introduction*. Hoboken, NJ, USA: Wiley, 2009.

[16] S. Grigorova and T. S. E. Maibaum, "Taking a page from the law books: Considering evidence weight in evaluating assurance case confidence," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Nov. 2013, pp. 387–390.

[17] L. Duan, S. Rayadurgam, M. P. E. Heimdahl, A. Ayoub, O. Sokolsky, and I. Lee, "Reasoning about confidence and uncertainty in assurance cases: A survey," in *Proc. Softw. Eng. Health Care*, M. Huhn and L. Williams, Eds. Cham, Switzerland: Springer, 2017, pp. 64–80.

[18] E. Denney, G. Pai, and I. Habli, "Dynamic safety cases for through-life safety assurance," in *Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng.*, vol. 2, May 2015, pp. 587–590.

[19] D. Heckerman, *A Tutorial on Learning with Bayesian Networks*. Berlin, Germany: Springer, 2008, pp. 33–82.

[20] D. Weyns, *Software Engineering of Self-Adaptive Systems*. Cham, Switzerland: Springer, 2019, pp. 399–443.

[21] S. Winkler and J. Von Pilgrim, "A survey of traceability in requirements engineering and model-driven development," *Softw. Syst. Model.*, vol. 9, no. 4, pp. 529–565, Sep. 2010.

[22] F. A. C. Pinheiro, "Requirements traceability," in *Perspectives on Software Requirements*. Boston, MA, USA: Springer, 2004, pp. 91–113.

[23] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, and P. Mäder, "Traceability fundamentals," in *Software and Systems Traceability*. London, U.K.: Springer, 2012, pp. 3–22.

[24] B. Iooss and A. Saltelli, "Introduction to sensitivity analysis," in *Handbook of Uncertainty Quantification*. Cham, Switzerland: Springer, 2017, pp. 1103–1122.

[25] F. I. Pérez, "Writing 'usable' nuclear power plant (NPP) safety cases using bowtie methodology," *Process Saf. Environ. Protection*, vol. 149, pp. 850–857, May 2021.

[26] G. N. Leveson and P. J. Thomas, *STPA Handbook*. Cambridge, MA, USA: MIT Partnership for Systems Approaches to Safety and Security (PSASS), 2018.

[27] R. Palin, D. Ward, I. Habli, and R. Rivett, "ISO 26262 safety cases: Compliance and assurance," in *Proc. 6th IET Int. Conf. Syst. Saf.*, Sep. 2011, pp. 1–6.

[28] C. Hobbs and M. Lloyd, "The application of Bayesian belief networks to assurance case preparation," in *Achieving Systems Safety*, C. Dale and T. Anderson, Eds. London, U.K.: Springer, 2012, pp. 159–176.

[29] J. Guiochet, Q. A. Do Hoang, and M. Kaaniche, "A model for safety case confidence assessment," in *Computer Safety, Reliability, and Security*, F. Koornneef and C. van Gulijk, Eds. Cham, Switzerland: Springer, 2015, pp. 313–327.

[30] D. Jeremy, H. Elizabeth, and J. Ken, *Requirements Engineering*, 4th ed. Cham, Switzerland: Springer, Aug. 2017.

[31] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, 1st ed. Hoboken, NJ, USA: Wiley, Feb. 2009.

[32] B. Marco, C. Jordi, and W. Manuel, *Model-Driven Software Engineering in Practice*, 2nd ed. Cham, Switzerland: Springer, 2017.

[33] *Ecore—Eclipsepedia*. Accessed: Aug. 7, 2023. [Online]. Available: https://wiki.eclipse.org/Ecore

[34] *OMG Unified Modeling Language*, Object Management Group, Milford, MA, USA, Standard 2.5.1, 2017. [Online]. Available: http://www.omg.org/spec/UML/

[35] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet Things*, vols. 1–2, pp. 81–98, Sep. 2018.

[36] M. Esposito, A. Minutolo, R. Megna, M. Forastiere, M. Magliulo, and G. De Pietro, "A smart mobile, self-configuring, context-aware architecture for personal health monitoring," *Eng. Appl. Artif. Intell.*, vol. 67, pp. 136–156, Jan. 2018.

[37] M. Maksimov, S. Kokaly, and M. Chechik, "A survey of tool-supported assurance case assessment techniques," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–34, Sep. 2019.

[38] J. L. de la Vara, M. Borg, K. Wnuk, and L. Moonen, "An industrial survey of safety evidence change impact analysis practice," *IEEE Trans. Softw. Eng.*, vol. 42, no. 12, pp. 1095–1117, Dec. 2016.

[39] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA, USA: MIT Press, 2012.

[40] *OMG Systems Modeling Language*, Object Management Group, Milford, MA, USA, Standard 1.7 Beta, 2022. [Online]. Available: https://www.omg.org/spec/SysML/1.7/Beta1

[41] S. Renooij, "Probability elicitation for belief networks: Issues to consider," *Knowl. Eng. Rev.*, vol. 16, no. 3, pp. 255–269, Sep. 2001.

[42] J. Rohmer, "Uncertainties in conditional probability tables of discrete Bayesian belief networks: A comprehensive review," *Eng. Appl. Artif. Intell.*, vol. 88, Feb. 2020, Art. no. 103384.

[43] J.-L. Boulanger, "Modeling," in *Certifiable Software Applications 3*, J.-L. Boulanger, Ed. Amsterdam, The Netherlands: Elsevier, 2018, pp. 75–96.

[44] M. J. Druzdzel and L. C. van der Gaag, "Elicitation of probabilities for belief networks: Combining qualitative and quantitative information," 2013, *arXiv:1302.4943*.

[45] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems—Part 5: Examples of Methods for the Determination of Safety Integrity Levels*, International Electrotechnical Commission, Geneva, Switzerland, TC 65/SC 65A, International Standard IEC 61508-5, 2010.

[46] *OMG Meta Object Facility (MOF) Core Specification*, Object Management Group, Milford, MA USA, International Standard formal/2019-10-01, 2019.

[47] D. Çetinkaya, "Model driven development of simulation models: Defining and transforming conceptual models into simulation models by using metamodels and model transformations," Ph.D. dissertation, Syst. Eng. Sect., Delft Univ. Technol., Middle East Tech. Univ., Delft, The Netherlands, 2013. [Online]. Available: https://doi.org/10.4233/uuid:3db45913-1662-429f-a385-ed53f5ac41fd

[48] E. Castillo, J. M. Gutierrez, and A. S. Hadi, "Sensitivity analysis in discrete Bayesian networks," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 27, no. 4, pp. 412–423, Jul. 1997.

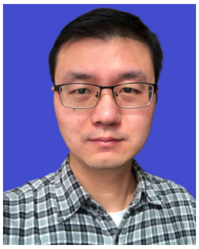
[49] V. M. H. Coupé, L. C. Van Der Gaag, and J. D. F. Habbema, "Sensitivity analysis: An aid for belief-network quantification," *Knowl. Eng. Rev.*, vol. 15, no. 3, pp. 215–232, Sep. 2000.

[50] C. Li and S. Mahadevan, "Sensitivity analysis of a Bayesian network," *ASCE-ASME J. Risk Uncertainty Eng. Syst., B, Mech. Eng.*, vol. 4, no. 1, Mar. 2018, Art. no. 011003, doi: 10.1115/1.4037454.

[51] M. J. Druzdzel, "SMILE: Structural modeling, inference, and learning engine and GeNIe: A development environment for graphical decision-theoretic models," in *Proc. Amer. Assoc. Artif. Intell.*, 1999, pp. 902–903.

[52] B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," *IEEE Trans. Softw. Eng.*, vol. 27, no. 1, pp. 58–93, Jan. 2001.

[53] D. Weyns and R. Calinescu, "Tele assistance: A self-adaptive service-based system exemplar," in *Proc. IEEE/ACM 10th Int. Symp. Softw. Eng. Adapt. Self-Manag. Syst.*, May 2015, pp. 88–92.



ZHENGGUO YANG received the B.E. degree from Information Engineering University, China, in 2008, the master's (M.E.) degree from Tianjin University, China, and the M.S. and Ph.D. degrees from the Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2012 and 2020, respectively. He was a Researcher with JAIST, until 2020. He is currently a Lecturer with Shangqiu Normal University, China. His research interests include home safety problem detection and safety of smart home systems, e.g., health monitoring systems.