Mapping SysML Diagrams Into Bayesian Networks: A Systems Engineering Approach for Fault Diagnosis

Arthur Henrique de Andrade Melani
Mechatronics and Mechanical System Engineering Department,
Polytechnic School of the University of São Paulo,
São Paulo 05508-010, Brazil
e-mail: melani@usp.br

Gilberto Francisco Martha de Souza
Mem. ASME
Mechatronics and Mechanical System Engineering Department,
Polytechnic School of the University of São Paulo,
São Paulo 05508-010, Brazil
e-mail: gfmsouza@usp.br

The growing complexity of equipment and systems has motivated the search for automated methods of fault diagnosis. Fault diagnosis represents the process of identifying the origin of a fault through the observation of a series of effects that it causes in the system. The method proposed in this paper for system fault diagnosis takes advantage of two very different techniques: Bayesian networks (BN) and systems modeling language (SysML). SysML allows the modeling of requirements, structure, behavior and parameters to provide a robust description of a system, its components, and its environment. This system model is used, in the proposed method, to obtain the BN graph in a novel structured procedure. The BN graph obtained must, in turn, present the components that are most likely responsible for a certain fault of the system under study. The BN model uses components reliabilities to solve the diagnosis problem. A case study of a water storage system is presented and it shows how the method can contribute to an assessment of the monitoring process of a system even in the early stages of its design. With this kind of information, the designer can assess the need for changes in the system to make it more reliable or better monitored. [DOI: 10.1115/1.4045975]

1 Introduction

According to Mobley [1], maintenance costs are an important part of the total operating costs of all manufacturing or production plants. Depending on the specific plant, maintenance costs can represent between 15 and 60% of the total production cost.

Nowadays, thanks to the advances in technology in the areas of monitoring and data acquisition, the interpretation of sensor readings makes it possible for plant managers to monitor the operational status of any individual piece of equipment, and accordingly to adopt a condition-based maintenance (CBM) approach based on fault diagnosis in order to avoid critical damages or degradation in advance. Fault diagnosis, according to Papadopoulos and McDermid [2], represents the process of identifying the origin of a fault through a series of effects that it causes in the system where it happens.

Venkatasubramanian et al. [3] list a number of desirable characteristics in a fault diagnosis system (FDS), of which the following stand out:

(1) Agility in detection and diagnosis: reducing the time spent to diagnose faults is critical in systems that pose risks to operators and the environment, as well as contributing to the reduction of total maintenance time;
(2) Ability to discriminate different faults: the FDS must list single units or component sets that are most likely responsible for the faults of the system;
(3) Ease of elucidation: the FDS should explain, in a simple way, why a certain unit or set of components is responsible for the fault of the system;
(4) Ability to identify multiple faults: the FDS should be able to identify whether a performance loss of a system is originated from a fault of a single piece of equipment or from a combination of faults of several units;
(5) Robustness: the results presented by the FDS should suffer little or no interference from noise present in the reading of the sensors and of uncertainties in relation to the operation of the monitored system;
(6) Adaptability: FDS must be adaptable to changes in the monitored system, both in its structure and in operating conditions.

The methods used in the implementation of fault diagnosis, in general, do not meet all these characteristics [4]. The choice of method to be used in a given system must be made according to the design needs.

The method proposed in this paper for system fault diagnosis takes advantage of two very different techniques: Bayesian networks (BN) and systems modeling language (SysML).

Bayesian networks [5] are diagrams that organize knowledge about a given system by means of a mapping of causes and effects. Systems based on Bayesian networks are able to automatically generate predictions or diagnoses, even when there is not complete information for this, using probabilistic calculations. They have been increasingly applied in diagnosis problems.

The use of techniques derived from model based system engineering (MBSE) is fundamental for the complete understanding of the functioning of a given system, and therefore contributes to the implementation of a fault diagnosis system. One of these techniques is SysML [6], a graphical language commonly used for modeling systems that can include hardware, software, data, people, installations, and other elements within the physical environment. The language supports the modeling of requirements, structure, behavior and parameters to provide a robust description of a system, its components, and its environment.

The development of a fault diagnosis methodology using Bayesian networks in conjunction with SysML language exploits the advantages found in the application of such techniques, allowing the maintenance team to quickly mitigate a fault, reducing system downtime.

The objective of this research is to develop a method for fault diagnosis based on a model of the system under study. This model will be developed using SysML and it will be used to obtain a Bayesian network graph through a novel structured procedure. The Bayesian network obtained will, in turn, be able to present the components that are most likely responsible for a certain

Manuscript received October 17, 2019; final manuscript received December 24, 2019; published online May 25, 2020. Assoc. Editor: Faisal Khan.

system fault. The fault in question will be observed by reading the indications of sensors present in the system.

One of the great advantages of the proposed method is the possibility of obtaining the Bayesian network to diagnose faults during the initial phases of a design, since SysML is widely used during the design of new and complex systems. By means of the Bayesian network obtained, the designer can identify, with some degree of uncertainty, the probability of a fault occurring during a certain system mode of operation and, furthermore, identify which component may be at a faulty state given the reading of the sensors. The information obtained by the BN can contribute to an assessment of the monitoring system. With this information still in the early stages of the design, the designer can assess the need for system architecture changes to make it more reliable and better monitored.

2 Fault Diagnosis

The goal of fault diagnosis is to identify the root causes of process abnormalities by using appropriate models, algorithms, and system observations. Therefore, a fault diagnosis system helps operations staff to detect, isolate, and identify faults, as well as to aid in troubleshooting [7].

One of the main advantages in using a fault diagnosis system is that it allows the operation team to take action to get around the identified problem quickly. When a failure occurs in a certain piece of equipment of the system, this affects the reading of the sensors responsible for monitoring its operation. These sensor readings are then processed and interpreted. If one of these measurable variables exceeds a threshold that identifies a dangerous process state, the monitoring system can either sound an alarm or stop the plant's operation in order to protect both plant and staff.

However, if a dangerous process state is not identified but the sensors readings are still deviating from normal operation, these measurements are categorized as symptoms and the FDS, then, reasons to determine the kind and location of the fault. Given the results from the FDS, the fault is categorized into hazard classes, supporting the decision-making process on defining how this fault is going to be managed.

As the complexity of systems increases, it becomes increasingly important to automate the fault diagnosis process in order to obtain systems with high reliability and availability. The automation of the diagnostic process can contribute to the prediction of failures and also optimize the response time of the maintenance team, reducing system downtime. Different fault diagnosis approaches are briefly described in Sec. 2.1.

2.1 Fault Diagnosis Approaches. Although this classification may change depending on the author, there are generally three different approaches to developing a fault diagnosis system [7]: model based, signal based, and knowledge based.

In the model-based fault diagnosis approach, mathematical process models are established to express dependencies between different measurable signals. The actual behavior of the system is then compared with its expected behavior obtained from the theoretical model. A relevant difference between the actual and the expected behavior means that the system may be faulty.

According to Lampis [4], a mathematical model created for a specific system usually cannot be used for other systems. In addition, it is difficult to adapt the model to design changes in the system itself. Due to its low adaptability and the difficulty to obtain mathematical models for complex systems, model-based fault diagnosis is considered an expensive approach.

Literature on model-based fault diagnosis is vast and comprehensive. Isermann [8], for example, presents how different pieces of equipment, such as actuators, motors, pumps, and pipelines, can be mathematically modeled so that their faults can be identified and diagnosed.

Due to the difficulty to obtain the theoretical models, the model-based approach is usually applied to specific components of a bigger, more complex system. Chen et al. [9] use this approach on an automated manual transmission shifting actuator, while Zhang et al. [10] apply it to pedal-by-wire systems. The model-based approach has also been used for spacecraft thrusters [11], air handling units [12], lithium-ion batteries [13,14], and other applications.

Differently from the model-based approach, according to Cai et al. [7], the signal-based fault diagnosis approach uses sensors signals to diagnose possible abnormalities and faults by comparing these detected signals with historical monitored data of the system. Experience in the system operation, therefore, is key for signal-based fault diagnosis, which means this approach can be difficult to be used for new systems. Also, the lack of knowledge regarding some extraordinary events due to the fact that they occur with extremely low frequency can also limit the use of this approach, especially when safety is a requirement [4].

On the other hand, signal-based reasoning can be useful when the understanding of the system is poor (i.e., its mathematical model is very difficult to be determined and model-based fault diagnosis is not an easy option) and knowledge of previous cases and actions taken is adequate [4,15]. Different techniques of signal analysis are employed in the literature related to signal-based fault diagnosis.

This approach is frequently used in electrical components, such as power converters. Chen and Lu [16] developed a signal-based fault diagnosis method for power converters of switched reluctance motors by analyzing changes in the measured root-mean-square current characteristics. Freire et al. [17] analyzed the Park's vector phase angle to determine open-circuit faults in converters of synchronous generator drives.

Signal-based fault diagnosis can also be used in mechanical components, such as gearboxes and motors. Feng and Zuo [18] used Fourier spectrum and demodulated spectra of amplitude envelope for torsional vibration signal analysis in order to diagnose faults of planetary gearboxes. Hong and Dhupia [19], on the other hand, combined correlated kurtosis and dynamic time warping techniques for monitoring gear faults.

Rather than models or signal patterns, knowledge-based fault diagnosis is based on a large amount of historical data and systems expert's knowledge [7]. A fault diagnosis process that is knowledge-based will comprise a set of rules (determined by historical data and system experts' knowledge) and an inference method that, by combining such rules with systems measured variables, derives a decision about the system's operating condition. According to Lampis [4], this approach is considered straightforward because each rule regards a different piece of information on the system, such as the relation between symptom and fault or between component and subsystem.

The symptoms that a faulty system presents can be evaluated using two types of knowledge, the heuristic and the analytical. Analytical knowledge about the system refers to measurable signals (i.e., sensor reading). A particular fault symptom is identified by checking these measurable values, i.e., if the reading of a sensor has exceeded a tolerance value.

Heuristic knowledge refers to the generally qualitative observations made by the operation team. Heuristic information is collected through inspections or maintenance performed on equipment and is presented in the form of noises, smells, vibrations, etc. Statistical values such as mean time between failures (MTBF) and failure probabilities, which are usually acquired from experience with the same or similar equipment, can also be considered heuristic [8].

The analytic and heuristic symptoms are, then, used for diagnosing system faults. Fault diagnosis can be done by classification methods or inference methods. Classification methods determine faults from symptoms patterns. Inference methods determine faults from fault-symptom trees, if-then rules, or other reasoning procedures [8].

Different techniques can be used to implement a knowledge-based fault diagnosis process, as reported in the literature review published by Gao et al. [20]. expert systems, for example, have been used for fault diagnosis in different systems, such as vehicles [21], chemical processes [22], and energy systems [23].

Failure modes and effects analysis (FMEA), a technique extensively used in reliability analysis, has also been used in the development of FDS. Case et al. [24] use the FMEA analysis in the creation of a diagnostic service tool for the analysis of automatic transmissions of automobiles. Barkai [25] describes the application of FMEA analysis, together with expert systems, to obtain a diagnostic system for off-road vehicles. Price and Taylor [26] propose the automation of the generation of FMEA tables for the identification of multiple faults through the use of AUTO-STEVE software.

Another technique from the area of reliability analysis that is used in fault diagnosis is fault tree analysis (FTA). Hurdle et al. [27] used FTA in the development of a fault diagnosis method that uses sensor readings present in the system as top events of fault trees. Contini et al. [28] used noncoherent fault trees in the modeling of security systems and showed that the results obtained can be more complete with this approach, when compared with the use of regular fault trees.

Techniques used in risk analysis have also been used in fault diagnosis. The hazard and operability analysis has been used by Hu et al. [29], in conjunction with dynamic Bayesian networks, to develop an intelligent fault diagnosis system for a fluidized catalytic cracking unit. Hidalgo and Souza [30] have also used hazard and operability analysis, FMEA, and FTA to develop an expert system for fault diagnosis.

Bayesian networks have been widely used for the diagnosis of system failures. The great advantage of this method is its probabilistic nature, which allows the insertion of uncertainty in the analysis.

2.2 Fault Diagnosis Based on Bayesian Networks.

Bayesian networks are graphical models that can represent the dependencies between the variables of a domain, being, for this reason, very much used in the resolution of problems of prediction and diagnosis. They emerged in the field of Artificial Intelligence in the 1980s to facilitate analyzes in which there are uncertainties [31].

A Bayesian network is a directed acyclic graph (DAG) in which nodes represent the variables of a universe and arcs represent the dependencies between them. The specifications of a BN are given below [32]:

(1) Each node represents a random variable, which can be discrete or continuous;
(2) A set of directed lines, or arcs, connect the nodes in pairs. If there is an arc from a node X to node Y, node X is called the parent of node Y. The graph has no directed cycles, i.e., a BN is a DAG;
(3) Each node Xi has a conditional probability distribution, given by P (Xi | Parents (Xi)), which quantifies the effect of the parent node on the child node.

According to Neapolitan [5], a BN contains a qualitative component, represented by the graph, and a quantitative component, represented by the conditional probabilities associated with each DAG node.

The presence of an arc connecting two nodes can be interpreted as a direct influence that the parent node exerts on the child node. According to Russell and Norvig [32], this suggests that causes should be the parent nodes and effects should be the child nodes. Figure 1 shows an example of a basic Bayesian network. In it, node C is said to be the parent of nodes E and B, B is parent of D and E, and node A is independent of other nodes.

In addition to the BN graph, it is necessary to define a set of conditional probabilities for each variable, or node, in order to quantitatively specify the influence of the parent nodes on the child nodes. If the variables are discrete, such conditional probabilities can be presented in a conditional probability table (CPT). In it, each line contains the conditional probability of the node in question, given a possible combination of values that the parent nodes can assume.

As an example, the CPTs of each node are shown in Fig. 2. For a better understanding, all nodes are binary, i.e., they have only two mutually exclusive states (true, v, or false, f). It can be seen from CPTs that the probability of a node being true or false depends on the state of its parents. The size of the conditional probabilities table depends on how many parents a particular node has. This means that the larger the number of parents or states that each parent can assume, the larger the CPT will be.

Bayesian networks can fully represent the domain under study, as well as the table of joint probabilities. Suppose a BN that represents the dependency relations of the n variables of a domain, X_1, …, X_n. Then, using the product rule [32]

$$P(x_1, \ldots, x_n) = P(x_n|x_{n-1}, \ldots, x_1) \times P(x_{n-1}, \ldots, x_1) \quad (1)$$

where $P(x_1, \ldots, x_n)$ is the same as $P(X_1 = x_1, \ldots, X_n = x_n)$. Reapplying the product rule n times

$$P(x_1, \ldots, x_n) = P(x_n x_{n-1}, \ldots, x_1) \times P(x_{n-1} x_{n-2}, \ldots, x_1) \times \cdots \times P(x_2|x_1) \times P(x_1) \quad (2)$$

Equation (2) can be rewritten as follows, using the so-called Chain Rule:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1}, \ldots, x_1) \quad (3)$$

Considering the BN graph, Eq. (3) can be rewritten as follows:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)) \quad (4)$$

Equation (4) is true only if $\text{Parents}(X_i) \subseteq \{x_{i-1}, \ldots, x_1\}$. Such a condition is satisfied by numbering the parent nodes before the child nodes. By analyzing the above equation, it is possible to realize that any joint probability can be obtained by multiplying conditional probabilities present in the Bayesian CPTs.

The Bayesian network allows the probability of a certain variable to be updated by observing the state of another variable, that is, it makes possible the calculation of the posterior probability. This process, called inference, can be performed with three different objectives [33]:

(1) Causes: given the causes, the probabilities of the effects are calculated;
(2) Diagnostics: given the effects, the probabilities of the causes are calculated;

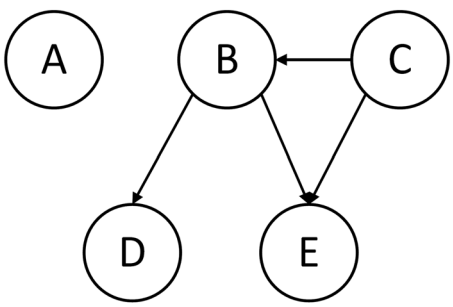

Fig. 1 Example of a Bayesian network

(3) Intercausal: given a cause, the probability of another cause is calculated.

Figure 3 shows Bayesian networks that exemplify the three objectives of making an inference. In BN, parent nodes represent causes and children nodes, effects.

Inference can be performed through algorithms which, in turn, can obtain exact or approximate solutions. The advantage of using algorithms that calculate approximate solutions is that the processing time is much shorter, a fundamental consideration in cases where BNs are very complex.

For the execution of the exact inference, where X is the variable to be used to calculate the posterior probability, E is the list of evidence variables (e is the list of observed values for these variables) and Y is the list of unobserved variables (y is the list of possible values for these variables), the following equation can be used:

$$P(Xe) = \frac{P(X,e)}{P(e)} = \alpha P(X,e) \tag{5}$$

where $\alpha = 1/P(e)$. Equation (5) above can be rewritten as follows [32]:

$$P(Xe) = \alpha \sum_{y} P(X,e,y) \tag{6}$$

The posterior probability, therefore, is obtained from the CPTs and (4), which provide the term $P(X, e, y)$. By analyzing Eq. (6), it is easy to see that, as the complexity of the Bayesian network increases, the summation present in the equation makes its use computationally very time-consuming. There are techniques that simplify this calculation, such as the elimination of variables [32], but in very complex networks not even these techniques can make the exact inference feasible.

The approximate inference decreases the processing time of the posterior probability using stochastic simulation. Russell and Norvig [32], present algorithms used to make the approximate inference.

In addition to being excellent for representing the relationship between component failures and their effects on the system, Bayesian networks are highly recommended tools for modeling diagnostic systems, since the inference process allows the identification of the possible causes of these effects. Figure 4 shows a general procedure for the construction and use of BN for fault diagnosis.

One of the most challenging steps in using Bayesian networks for fault diagnosis is the modeling of its structure. According to Lampis [4], there is no structured way to construct a Bayesian network; besides, it is difficult to obtain the conditional probabilities present in the CPTs. Some authors, therefore, propose the use of different methods to help in the construction of the BN graph, which can be categorized into two types: the knowledge representation methods and the machine learning methods [7].

In machine learning methods, algorithms are developed in order to learn the structure of the Bayesian network based on faults and fault symptoms data. Lin et al. [34], for example, used K2 algorithm, which is a score-based algorithm that learns probabilistic networks from databases, to support quality of service management and qualitative diagnosis on a peer-to-peer network.

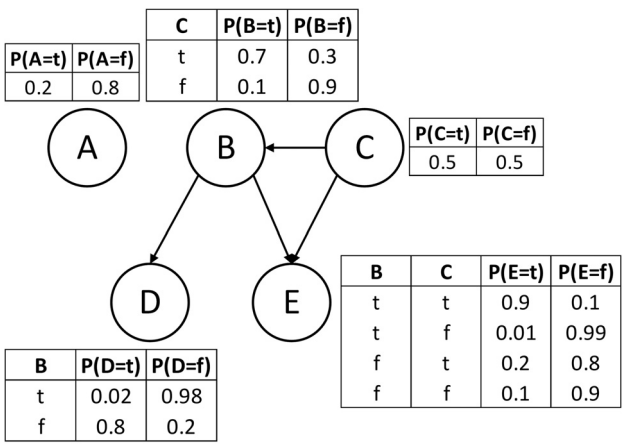

Fig. 2 Example of a BN alongside the CTPs of each node

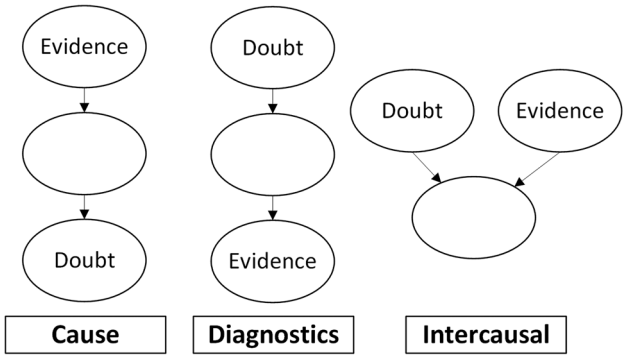

Fig. 3 Inference (adapted from Ref. [33])

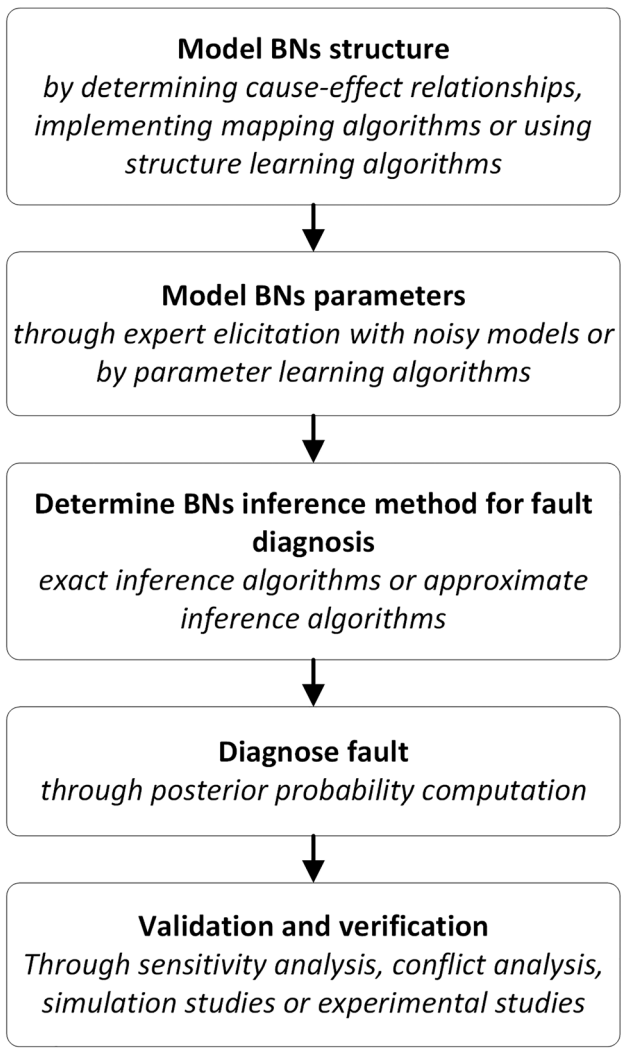

Fig. 4 Flowchart of BN-based fault diagnosis [7]

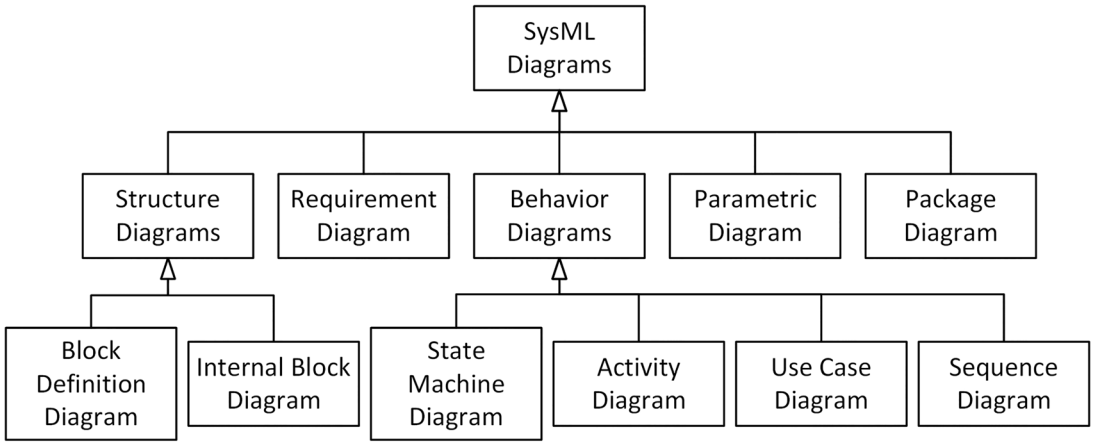

Fig. 5 SysML diagram types [45]

Yan and Peng [35] performed fault diagnosis of hydraulic–electrical simulation systems based on a BN, in which its graph was learnt from a statistical strategy algorithm. Jin et al. [36] proposed a Bayesian network for fixture fault diagnosis in an autobody assembly process. Authors have also used mutual information test that exposes the causal relationship between sensor readings and faults to obtain the BN graph.

Machine learning methods are able to generate very efficient BN graphs and can produce very accurate results regarding fault diagnosis. However, these methods are extremely dependent on data for learning. If there is not enough data for their algorithms to determine the causal relationships between faults and sensor readings, the results obtained may not be representative of the behavior of the real system.

The methods of knowledge representation, on the other hand, are based on the knowledge of system experts for the development of Bayesian network structure. They can also obtain this graph by mapping other formal knowledge models into the BN.

BN structure can be developed through brainstorms with experts, who must take into account all factors that can influence the outcome of fault diagnosis. Mechraoui et al. [37], and Xu [38] develop Bayesian networks through the knowledge of experts on the system under study and, through inference, can identify the possible causes of a deviation in the reading of a system sensor. These works also take advantage of the fact that BN can represent uncertainties regarding the behavior of the system.

Several authors used mapping algorithms to obtain BN structure. Bobbio et al. [39] proposed a FDS of redundant microprocessor system by directly mapping FTAs into BNs. Chiremsel et al. [40] developed a similar approach for safety instrumented systems used in the oil and gas industry. Lampis and Andrews [41] improved this proposal by using noncoherent fault trees instead of regular ones and developed a FDS for a water tank system. Lo et al. [42] have synthesized the BN graph from bond graphs for fault diagnosis of a single tank system.

Knowledge representation methods are highly recommended if there is not enough data to implement machine-learning methods. However, since these methods depend on expert's knowledge, they can produce inaccurate models. That is why mapping algorithms are preferred because they provide a more structured way for the expert to represent its knowledge. In Sec. 2.3, Systems Modeling Language (SysML), the systems engineering (SE) technique used in the proposed method to map the BN graph is presented.

2.3 Systems Engineering, Systems Modeling Language and Fault Diagnosis. According to INCOSE [43], a system is defined as "an integrated set of elements, subsystems, or assemblies that accomplish a defined objective." These elements include products (hardware, software, and firmware), processes, people, information, techniques, facilities, services, and other support elements." ISO/IEC/IEEE [44] defines it as a combination of interacting elements organized to achieve one or more stated purposes.

Designing and managing systems is the focus of SE. The need for this field of engineering arose with the increasing complexity of systems, which required new strategies, techniques and procedures to be developed.

With the objective of assisting in the development of complex systems, SE practices are becoming more formalized, structured and rigorous. The model-based systems engineering is an SE approach that can help to manage system complexity.

A model-based approach in the SE scope aims to maintain and synchronize all the information about the system in a consistent and complete way. However, for MBSE to be actually practiced, according to Friedenthal et al. [45], a robust and standardized

Table 1 SysML diagrams and its correspondence with UML [46]

SysML diagram	Purpose	UML analog
Activity diagram (ACT)	Shows system behavior as control flows and data flows Useful for functional analysis	Activity
Block definition diagram (BDD)	Shows system structure as components along with their properties, operations, and relationships	Class
Internal block diagram (IBD)	Shows the internal structures of components, including their parts and connectors	Composite structure
Package diagram (PKG)	Shows how a model is organized into packages, views, and viewpoints	Package
Parametric diagram (PAR)	Shows parametric constraints between structural elements	N/A
Requirement diagram (REQ)	Shows system requirements and their relationships with other elements	N/A
Sequence diagram (SD)	Shows system behavior as interactions between system components	Sequence
State machine diagram (STM)	Shows system behavior in terms of states that a component experiences in response to some events	State machine
Use case diagram (UC)	Shows systems functions and the actors performing them	Use case

modeling language is critical. One of the modeling languages used in MBSE is SysML.

SysML is based on the unified modeling language (UML) and was created by an informal association of SE experts called SysML Partners, organized by Cris Kobryn in 2003. It supports the specification, design, analysis, and verification of complex systems. By means of a solid semantic foundation, SysML is able to represent various aspects of a system, such as structure, behavior, and requirements [45].

SysML includes nine diagrams, as shown in Fig. 5. Seven of those diagrams come from UML 2, while the requirement and the parametric diagrams were exclusively developed for SysML. Table 1 shows the main purpose of each SysML diagram and its correspondence with UML diagrams.

One very important aspect of SysML is the possibility of cross-connecting different model elements that appear in different diagrams. These cross-connections are able to represent deeper relationships within the elements of the model. There are four

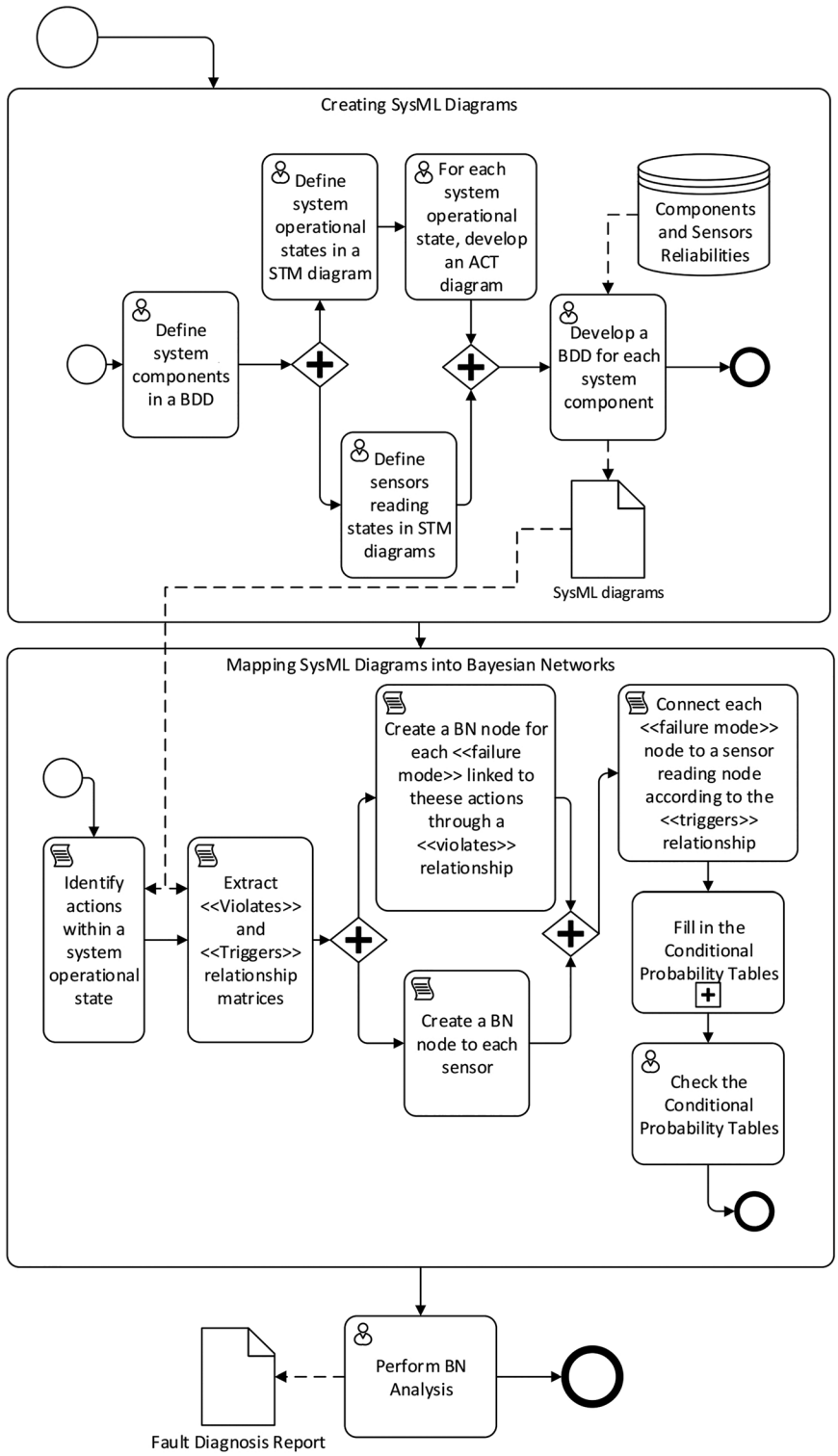

Fig. 6 A BPMN diagram of the proposed method

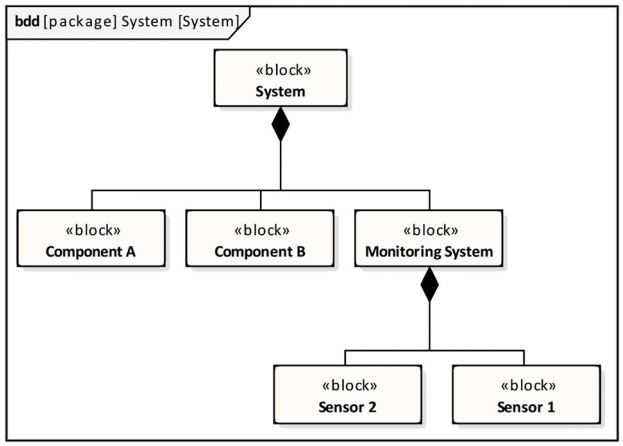

Fig. 7 Example of a system BDD

main types of cross-connections between model elements: allocate, satisfy, value biding, and verify.

The allocate relationship is used to determine which block (or system component) is responsible for executing a given action or behavior. Allocation, therefore, links a behavior type diagram (activity diagram (ACT), for example), with a structure diagram (IBD, for example).

The satisfy relationship determines which block is responsible for accomplishing a given requirement. In other words, the satisfy relationship, which links a structure type diagram with a requirement diagram, shows which system component is meant to satisfy a system requirement.

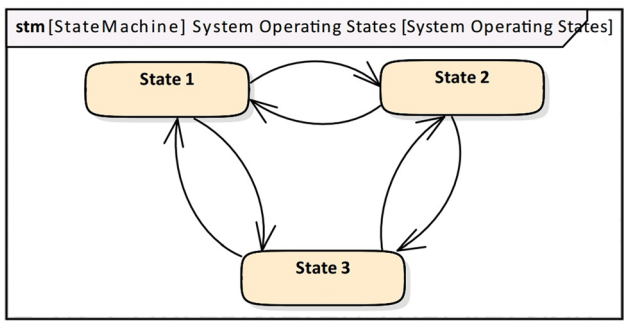

Fig. 8 Example of STM diagram with system operating states

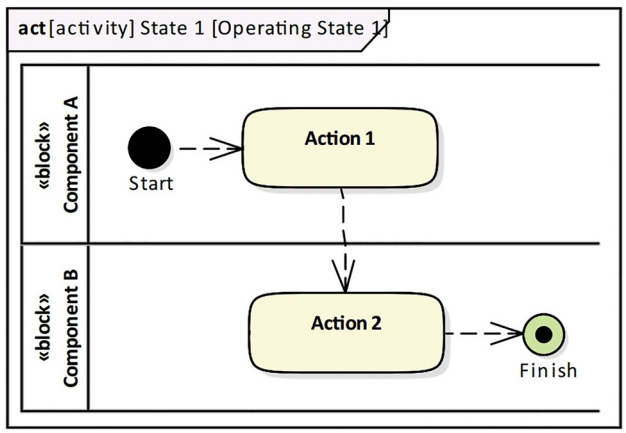

Fig. 9 Example of an ACT diagram of one of the system operating states

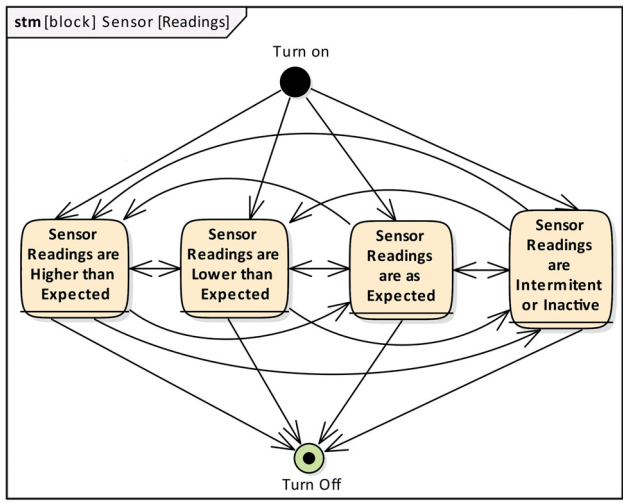

Fig. 10 Example of STM diagram with sensor reading states

The value binding relationship is used to link a value property from a block to systems equations. In other words, system component characteristics that are represented by values, such as mass, and volume, can be assigned to system equations presented in the Parametric Diagram through this value binding relationship.

The verify relationship is used to represent how a test can be run to make sure a requisite is being satisfied. This verification is represented by linking a parameter in the parametric diagram to a requirement in REQ. It should be noted that the satisfy relationship does not imply that the requirement is actually been satisfied by the block, just that the block is supposed to satisfy it, that is why the verify relationship is useful.

The purpose of having the nine diagrams and the cross-connecting relationships described above is to allow the system to be represented by different perspectives and at the same time to maintain consistency among the different views [45].

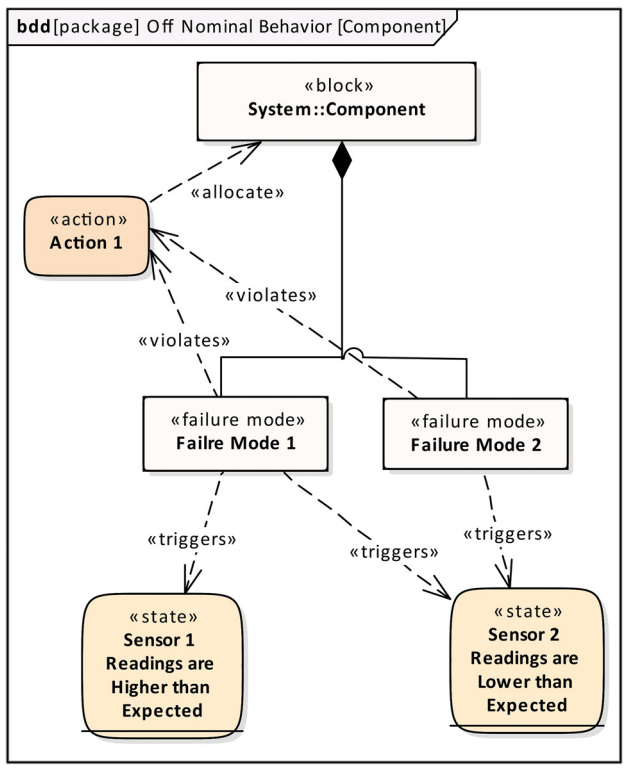

Fig. 11 Example of a BDD of a single system component

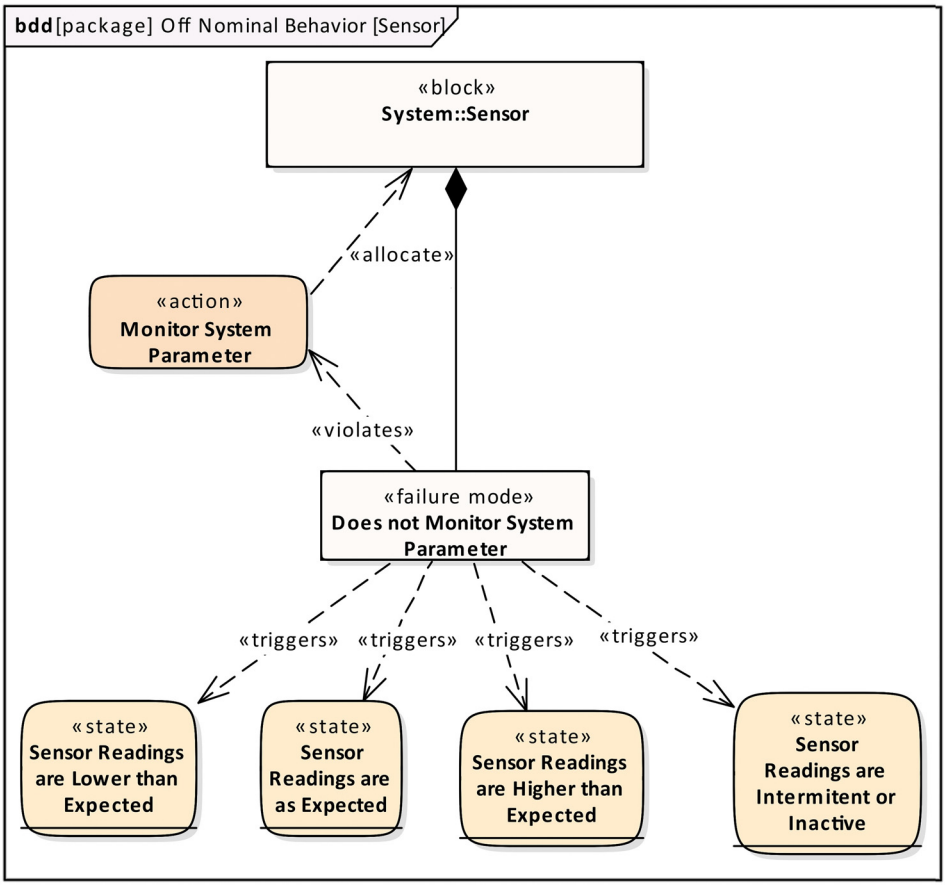

Fig. 12 Example of a BDD of a system sensor

Several authors have used SysML for supporting reliability and safety analysis. The advantage of using a system model such as SysML in these analyses is that comparison and trade-offs between alternative designs can be done in an easier way. Helle [47], for example, proposes a methodology that uses SysML for an automated safety analysis.

Since FMEA and FTA techniques are widely used for reliability and safety analysis, some researches have focused on the automation of generation of FMEA tables and FTA diagrams. David et al. [48], for example, present a reliability study of complex physical systems by using SysML for automating FMEA. Hecht et al. [49] propose a different approach for an automated generation of FMEA mainly based on state machine diagrams (STMs) from SysML.

Yakymets et al. [50] and Mhenni et al. [51] propose similar ways of generating a FTA from an IBD. Recently, Mhenni et al. [52] presented the SafeSysE, a tool that automatically generates both FMEA and FTA from SysML diagrams, and applied it to a electromechanical actuator.

Since SysML language is relatively new, few researchers use such a tool to diagnose system faults. In fact, the only published work found on the subject is that of Suiphon et al. [53], which promotes an initial discussion of how SysML can be used to diagnose faults, but without presenting a method for such.

From the study of the modeling of a system in SysML language, it was found that this modeling conveys a lot of information about it. This amount of information can be used to diagnose system faults. Indeed, the fact of being able to relate the structural parts of the model to dynamic behaviors can make it easier to trace the faulty components of a system. Section 3 presents the proposed method for mapping ACT into BN for system fault diagnosis.

3 The Proposed Method

As in current diagnostic methods, the detection of a fault is made when the system deviates from its nominal behavior. This deviation is detected by sensors that monitor the operation of the

Table 2 ≪violates≫ Relationship matrix

		System operational state 1			
		Action 1	Action 2	Monitor system parameter 1	Monitor system parameter 2
Component A	Failure mode 1	●			
	Failure mode 2	●			
Component B	Failure mode 3		●		
	Failure mode 4		●		
	Failure mode 5				
Sensor 1	Failure mode 6			●	
Sensor 2	Failure mode 7				●

Table 3 ≪triggers≫ Relationship matrix

		Component A		Component B			Sensor 1	Sensor 2
		Failure mode 1	Failure mode 2	Failure mode 3	Failure mode 4	Failure mode 5	Failure mode 6	Failure mode 7
Sensor 1 readings	Readings are higher than expected	●			●	●	●	
	Readings are as expected						●	
	Readings are lower than expected			●			●	
	Sensor readings are intermittent or inactive						●	
Sensor 2 readings	Readings are higher than expected			●		●		●
	Readings are as expected							●
	Readings are lower than expected	●	●					●
	Sensor readings are intermittent or inactive							●

system. As soon as a fault is detected, an inference engine must reason about which system component is most probably responsible for the detected deviation, i.e., which is the faulty component. The inference engine used in this method is the Bayesian network, which uses as input not only the reading of the sensors but also the reliability of the components of the system.

In order for the Bayesian network to perform the inference process, its graph must be previously defined. For this to happen, SysML diagrams are developed and mapped into a BN. Figure 6 presents, using the business process model and notation (BPMN), the framework of the proposed method for system fault diagnosis. Sections 3.1 and 3.2 describe in detail the two major steps of the method, which are the development of SysML diagrams and their translation into BNs.

3.1 Creating Systems Modeling Language Diagrams. One of the main features of SysML is its ability to represent all the knowledge about a given system and, from there, contribute to several analysis and consequent improvements. In order to obtain the Bayesian network for fault diagnosis, some SysML diagrams must be developed and some specific information must be inserted into them. These requirements, however, do not limit or impair the designer's ability to create other diagrams or add more information to those already created. This means that the proposed method can be part of the design of new systems without harming its development.

As Fig. 6 shows, the first diagram to be developed is the block definition diagram (BDD) of the system. This diagram is important for the method because it identifies all components of the system whose faults should be diagnosed. Figure 7 illustrates how this BDD should look like.

Next, a STM diagram should be developed to represent the system operating states, or system operating modes. A system can have several operating states and behave differently in each one of them, meaning that its sensors readings may be different in each state. This means that the fault diagnosis process must take into account the mode of operation the system is in to avoid misleading results. Figure 8 shows how this diagram should look like. It is important to note that when a fault or failure occurs, the system can be reconfigured and operate in another way. This is very common in systems that have redundant components (when one fails, the other starts to operate in its place). In this case, each mode of operation (before and after the fault) must be represented in the STM.

Once the operating states of the system have been defined, it is necessary to know how the system behaves in each of them, that is, what are the actions of each of its components. For that, an ACT diagram like the one in Fig. 9 is developed, where each action is allocated to its respective component.

Besides knowing how the system behaves, it is also necessary to know how the readings of its sensors must be interpreted. An STM must, therefore, be developed in order to represent the different reading states that will be used to reason about the component failure that may be in progress, as shown in Fig. 10. It is important to note that any different behavior from the readings of a sensor that can help to distinguish a fault (such as "readings are above expectation" or "fluctuating readings") must be represented in this diagram. The transition from one reading state to another is determined by thresholds that can be set by the system expert or by detection methods.

Last but not least, a BDD of each component (including sensors) needs to be developed. In this diagram the failure modes of this component as well as its actions should be represented. It is important to remember that a component can perform different functions, or actions, in each mode of operation, but all of them must be represented in this diagram. In addition, it is necessary to show which of these actions are impacted or impaired by failure modes. This is done using the ≪violates≫ relationship. It is also necessary to represent how a failure mode impacts the readings of

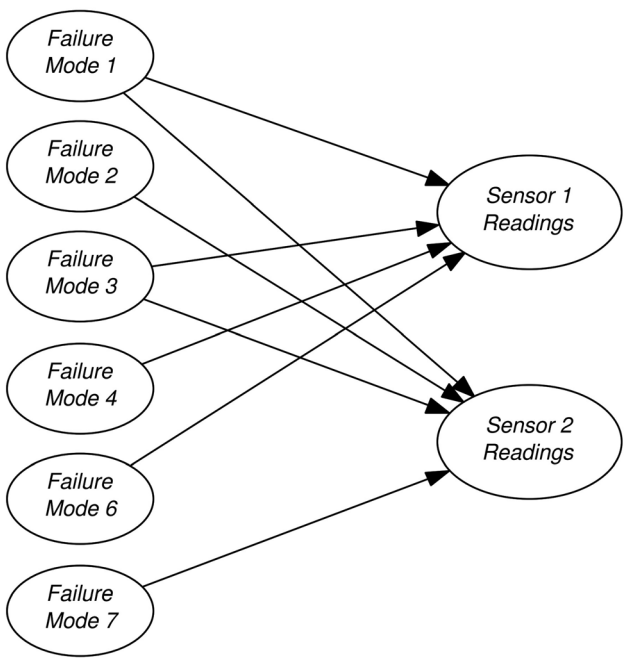

Fig. 13 Obtained Bayesian network

system sensors, and this is done by a ≪triggers≫ relationship. Examples of this diagram are shown in Figs. 11 and 12.

It is important to note that although ≪allocate≫ and ≪violates≫ relationships already exist in the SysML formalism, the ≪triggers≫ relationship is being introduced in this work. According to Friedenthal et al. [45], SysML can be customized for specific domains and therefore the ≪triggers≫ relationship is proposed here to enhance the analysis to be done with the model.

The purpose of using the ≪triggers≫ relationship is to show how the progression of a fault into a failure can be noticed by a sensor. This means that the faults to be detected in the method correspond to the progression of the failure modes represented in the BDD.

Figure 12 illustrates how the BDD of a sensor may look like if it is unknown how it will behave in case of failure. It is noted that sensor failure implies that its reading may behave in any of the states previously defined in the STM sensor.

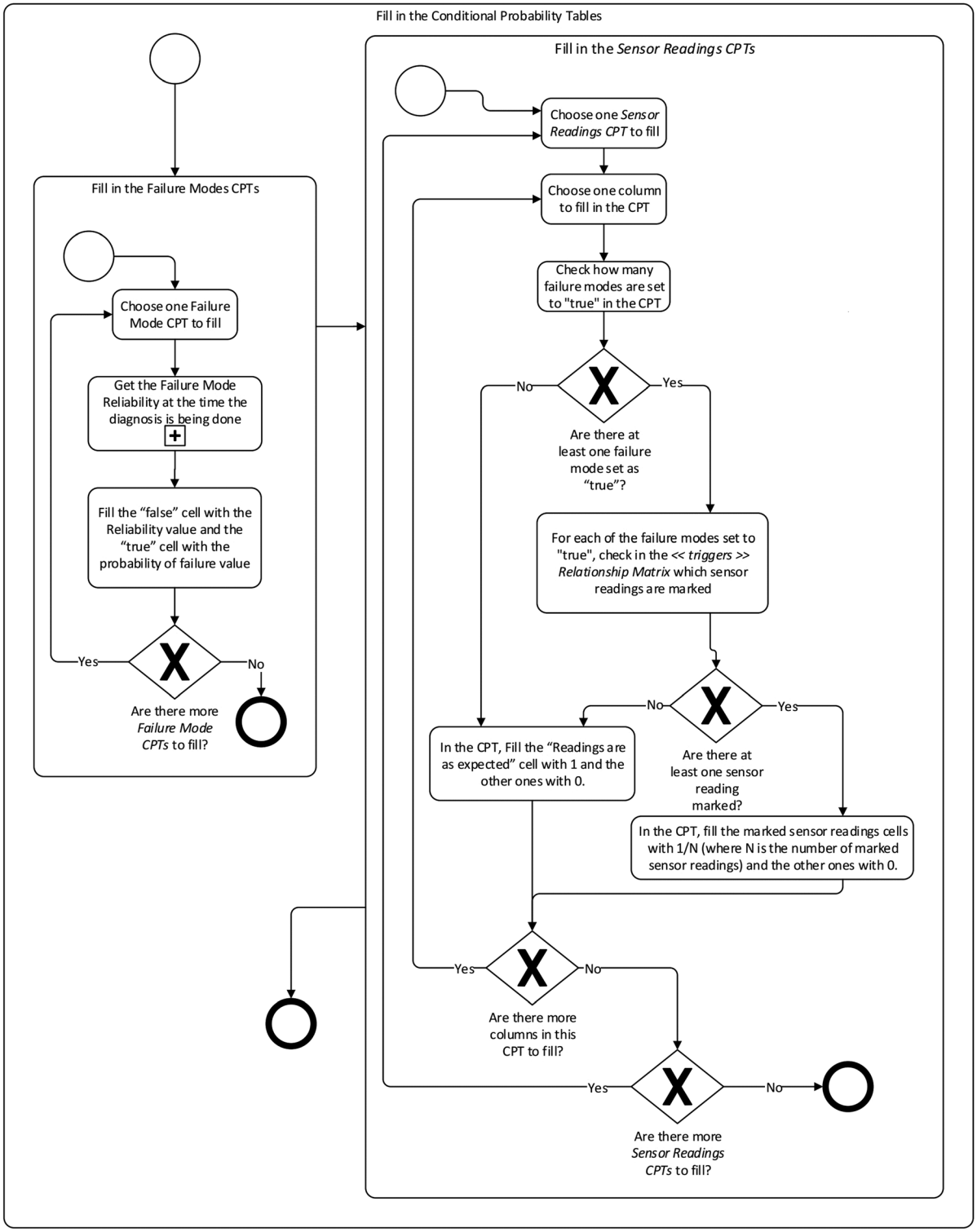

Fig. 14 Flowchart of how to fill in the Bayesian network CPTs

Table 4 CPT of "sensor 1 readings" node

Failure mode 1	True								False							
Failure mode 3	True				False				True				False			
Failure mode 4	True		False		True		False		True		False		True		False	
Failure mode 6	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Readings are higher than expected	0.25	0.5	0.25	0.5	0.25	1	0.25	1	0.25	0.5	0.25	0	0.25	1	0.25	0
Readings are as expected	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	1
Readings are lower than expected	0.25	0.5	0.25	0.5	0.25	0	0.25	0	0.25	0.5	0.25	1	0.25	0	0.25	0
Sensor readings are intermittent or inactive	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0

Table 5 CPT of "sensor 2 readings" node

Failure mode 1	True								False							
Failure mode 2	True				False				True				False			
Failure mode 3	True		False		True		False		True		False		True		False	
Failure mode 7	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Readings are higher than expected	0.25	0.5	0.25	0	0.25	0.5	0.25	0	0.25	0.5	0.25	0	0.25	1	0.25	0
Readings are as expected	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	1
Readings are lower than expected	0.25	0.5	0.25	1	0.25	0.5	0.25	1	0.25	0.5	0.25	1	0.25	0	0.25	0
Sensor readings are intermittent or inactive	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0	0.25	0

Once all the above diagrams have been developed, the necessary knowledge to develop the Bayesian network for fault diagnosis is already represented in SysML language. The next step of the method, therefore, is to map such knowledge into a BN graph, as stated in Fig. 6.

3.2 Mapping Systems Modeling Language Diagrams Into Bayesian Networks. The first step in mapping the SysML diagrams to BNs is to identify the actions of each operating state of the system. This process is done because each operational state must have a unique Bayesian network, since the behavior of the components and, consequently, of the sensors, may be different in each of them.

Once these actions have been identified, the ≪violates≫ relationship matrix is extracted. The goal of this matrix is to show which failure modes are linked to which actions by the ≪violates≫ relationship using a mark (like a black dot, for example), as illustrated in Table 2. In addition, the ≪triggers≫ relationship matrix must also be extracted, showing which failure modes are linked to which sensor readings by the ≪triggers≫ relationship, as illustrated by Table 3.

These relationship matrices contribute, in the proposed method, to the development of the BN graph and to the task of filling in the CPTs. To obtain the graph, two types of nodes will be considered in the BN: those representing failure modes and those representing the sensor reading. The failure mode nodes will have only two states: true or false. The sensor reading nodes will have as many states as shown in the corresponding STM diagram.

The first step to the creation of the BN graph is to verify in the ≪violates≫ matrix which failure modes are linked to the actions of a system operating state. Those with the mark get a node in the Bayesian network. All sensors also get a node in the BN, and to show the dependence between the failure mode nodes and the sensor nodes, i.e., to connect them through an arc, the existence of a mark between some sensor reading state and a failure mode in the ≪triggers≫ matrix should be verified. Figure 13 shows the Bayesian network obtained using the previously presented matrices.

The last and most challenging step in obtaining the BN graph is the task of correctly filling in the CPTs. Both types of nodes, those representing failure modes and those representing sensors readings, have their CPTs filled in according to the BPMN diagram presented in Fig. 14.

Since the failure mode node has only two mutually exclusive states, true or false, its conditional probability tables for components must be filled in with its reliability and failure probability values for a given time of operation. Reliability, $R(t)$, is the likelihood that a product, machine, system, or equipment will perform its function without failure for a predetermined period of time within the specifications for which it was designed [54]. Once the reliability is defined, the failure probability, P(t), can be obtained by using the following equation:

$$P(t) = 1 - R(t) \tag{7}$$

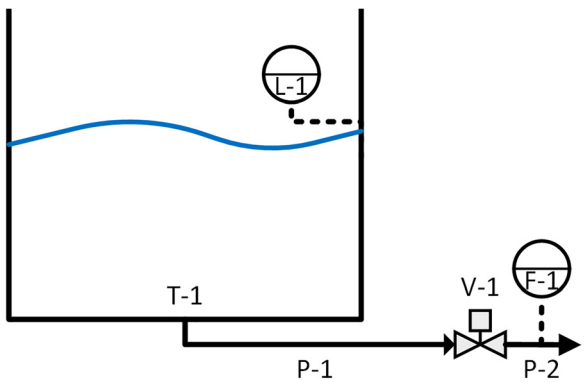

Fig. 15 Water storage system

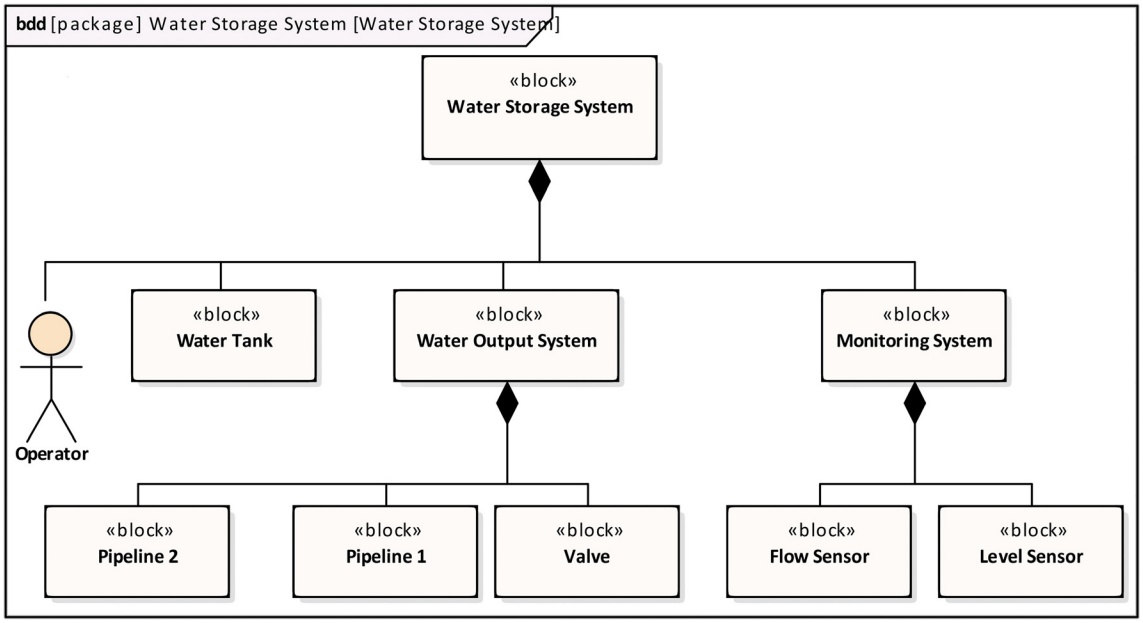

Fig. 16 BDD of water storage system

The reliability of any given piece of equipment is intrinsically dependent upon its nature, operating conditions, and environment. Because of this, it can be represented by different probability distributions and can be calculated by parameters such as failure rate.

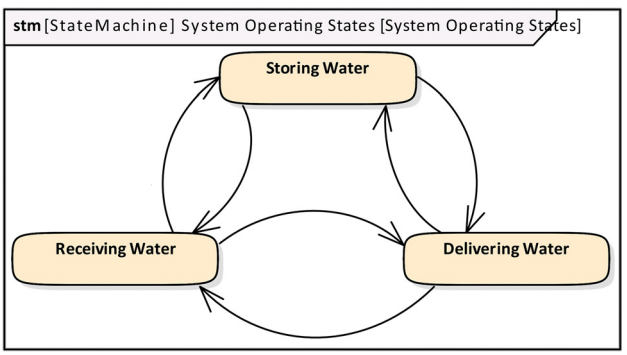

Fig. 17 STM of water storage system

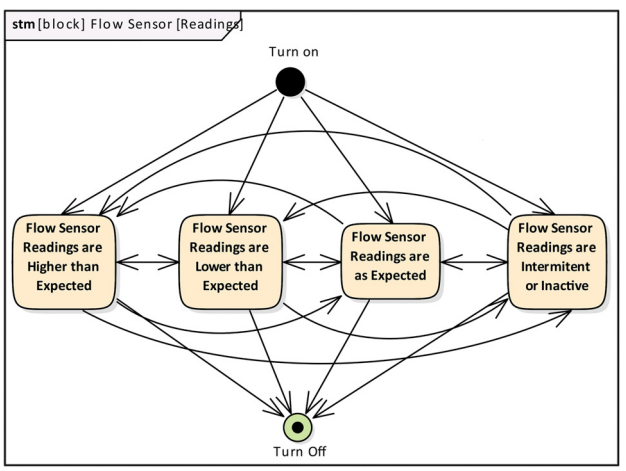

Fig. 18 STM diagram with flow sensor reading states

Since reliability and probability of failure are used in the proposed method, the availability of realistic, specific, and updated data is of utmost importance. Such data may be obtained through failure history of the piece of equipment and, if such histories do not exist, may be obtained from databases such as OREDA [55] and NPRD [56]. In the latter case, however, it is important to verify that the piece of equipment listed in the database has the same or similar nature, operating conditions, and environment as the one under study. Alternatively, these values could also be obtained in a database created in a SysML environment, as suggested by Cressent et al. [57]. This type of information can also be displayed in the block definition diagram (as suggested in Fig. 6).

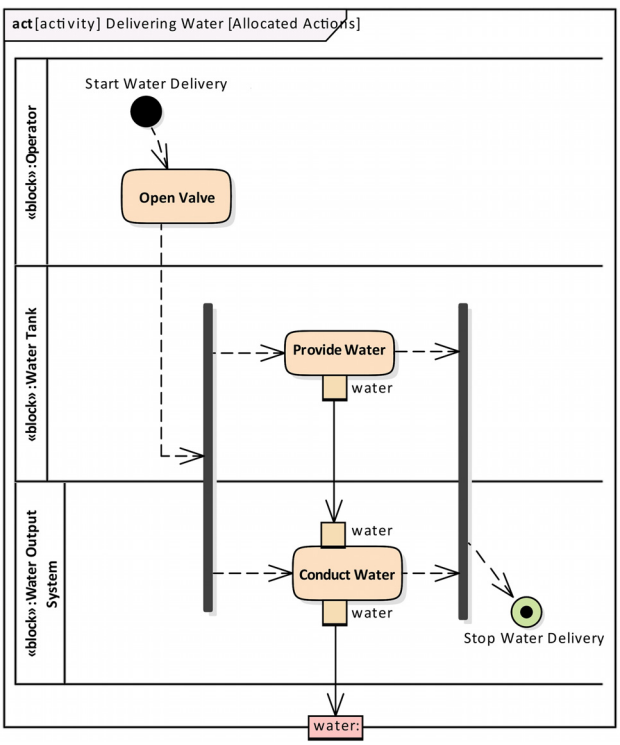

Fig. 19 ACT diagram for the delivering water operating state

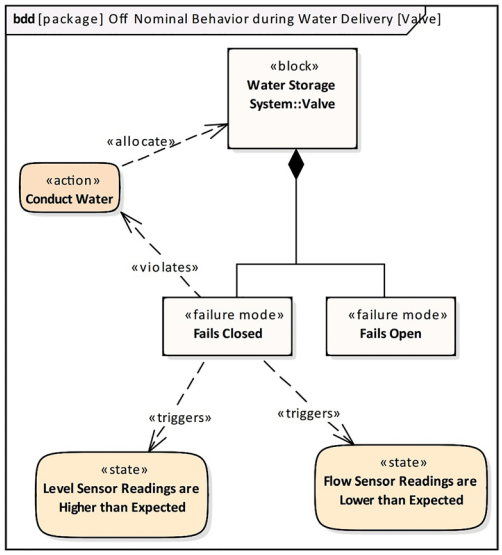

Fig. 20 BDD of the valve

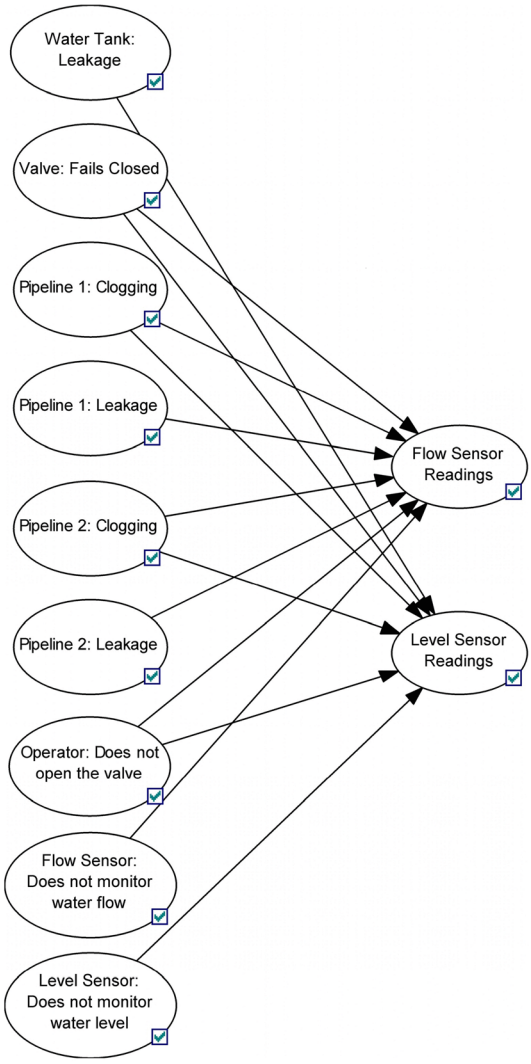

Fig. 21 BN graph

For nodes representing sensors readings, filling in their CPTs is particularly problematic when multiple failure modes occur. Since the developed SysML diagrams do not contemplate the effect of simultaneous faults on the sensor readings (because the exponentially growth of the number of failure modes combinations would make such a representation unfeasible in diagram form), an alternative approach is presented in Fig. 14. It is considered that if different failure modes (or rather the faults whose progression will lead to these failure modes) trigger different readings from the same sensor but they occur simultaneously, then the probability that any of these readings will occur is the same. This is clearly an approximation, since a combination of faults may result in different sensor readings compared to the individual occurrence of these faults. Such method approach can be manually corrected by the designer after the automatic filling in of the CPTs.

Tables 4 and 5 show the obtained CPTs for the two sensors presented in the example from the previously presented ≪violates≫ and ≪triggers≫ relationship matrices.

Once the graph is defined, along with the CPTs, the BN can be used to perform fault diagnosis. It is hoped that the results shown by the BN will not only reveal which failure is most likely based on sensor readings but also which failures are not being properly monitored.

The proposed method, therefore, uses SysML to help system experts in obtaining a BN for fault diagnosis in a structured and organized manner. The main purpose of SysML is to concentrate all knowledge about a system in one place. Through its nine diagrams, it is possible to represent how a system works, what components it is made of, how it interacts with its operators and maintainers, how it behaves under different operating conditions, etc. If used correctly, the authors believe that not only fault diagnosis, but other types of analysis can be developed using the information represented in SysML, such as reliability, availability, maintainability, efficiency, risk management, and asset

Table 6 ≪violates≫ Relationship matrix

		Delivering water				
		Open valve	Provide water	Conduct water	Monitor water flow	Monitor water level
Water tank	Leakage		●			
Valve	Fails closed			●		
	Fails open					
Pipeline 1	Clogging			●		
	Leakage			●		
Pipeline 2	Clogging			●		
	Leakage			●		
Operator	Does not open the valve	●				
Flow sensor	Does not monitor water flow				●	
Level sensor	Does not monitor water level					●

Table 7 ≪triggers≫ Relationship matrix

		Flow sensor				Level Sensor			
		Readings are higher than expected	Readings are as expected	Readings are Lower than expected	Sensor readings are intermittent or inactive	Readings are higher than expected	Readings are as expected	Readings are lower than expected	Sensor readings are intermittent or inactive
Water tank	Leakage							●	
Valve	Fails closed			●		●			
	Fails open								
Pipeline 1	Clogging			●		●			
	Leakage			●					
Pipeline 2	Clogging			●		●			
	Leakage			●					
Operator	Does not open the valve			●		●			
Flow sensor	Does not monitor water flow	●	●	●	●				
Level sensor	Does not monitor water level					●	●	●	●

Table 8 Reliability and failure probability for the CPTs

		λ	True	False
Water tank	Leakage	2.00×10^{-6}	0.017367	0.982633
Valve	Fails closed	1.00×10^{-5}	0.083873	0.916127
	Fails open	1.00×10^{-5}	0.083873	0.916127
Pipeline 1	Clogging	1.00×10^{-6}	0.008722	0.991278
	Leakage	1.50×10^{-5}	0.123133	0.876867
Pipeline 2	Clogging	1.00×10^{-6}	0.008722	0.991278
	Leakage	1.50×10^{-5}	0.123133	0.876867
Operator	Does not open the valve	1.60×10^{-5}	0.130781	0.869219
Flow sensor	Does not monitor water flow	1.20×10^{-6}	0.010457	0.989543
Level sensor	Does not monitor water level	1.40×10^{-6}	0.012189	0.987811

Table 9 Posterior probabilities of the TRUE state of each failure mode

		Water tank	Valve	Pipeline 1		Pipeline 2		Operator	Flow sensor	Level sensor
Flow sensor readings	Level sensor readings	Leakage	Fails closed	Clogging	Leakage	Clogging	Leakage	Does not open the valve	Does not monitor water flow	Does not monitor water level
HIGHER	HIGHER	0.009	0.381	0.04	0.123	0.04	0.123	0.595	1	0.014
HIGHER	NORMAL	< 0.001	< 0.001	< 0.001	0.123	< 0.001	0.123	< 0.001	1	0.004
HIGHER	LOWER	0.837	0.053	0.006	0.123	0.006	0.123	0.083	1	0.166
HIGHER	INTERMITTENT	0.017	0.084	0.009	0.123	0.009	0.123	0.131	1	1
NORMAL	HIGHER	0.015	0.091	0.009	0.029	0.009	0.029	0.141	0.237	0.766
NORMAL	NORMAL	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	0.003	0.003
NORMAL	LOWER	0.852	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	0.004	0.151
NORMAL	INTERMITTENT	0.017	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	0.004	1
LOWER	HIGHER	0.009	0.385	0.04	0.124	0.04	0.124	0.6	0.003	0.006
LOWER	NORMAL	< 0.001	< 0.001	< 0.001	0.527	< 0.001	0.527	0.002	0.011	0.007
LOWER	LOWER	0.807	0.157	0.016	0.363	0.016	0.363	0.245	0.008	0.196
LOWER	INTERMITTENT	0.017	0.21	0.022	0.308	0.22	0.308	0.327	0.007	1
INTERMITTENT	HIGHER	0.009	0.381	0.04	0.123	0.04	0.123	0.595	1	0.014
INTERMITTENT	NORMAL	< 0.001	< 0.001	< 0.001	0.123	< 0.001	0.123	< 0.001	1	0.004
INTERMITTENT	LOWER	0.837	0.053	0.006	0.123	0.006	0.123	0.083	1	0.166
INTERMITTENT	INTERMITTENT	0.017	0.084	0.009	0.123	0.009	0.123	0.131	1	1

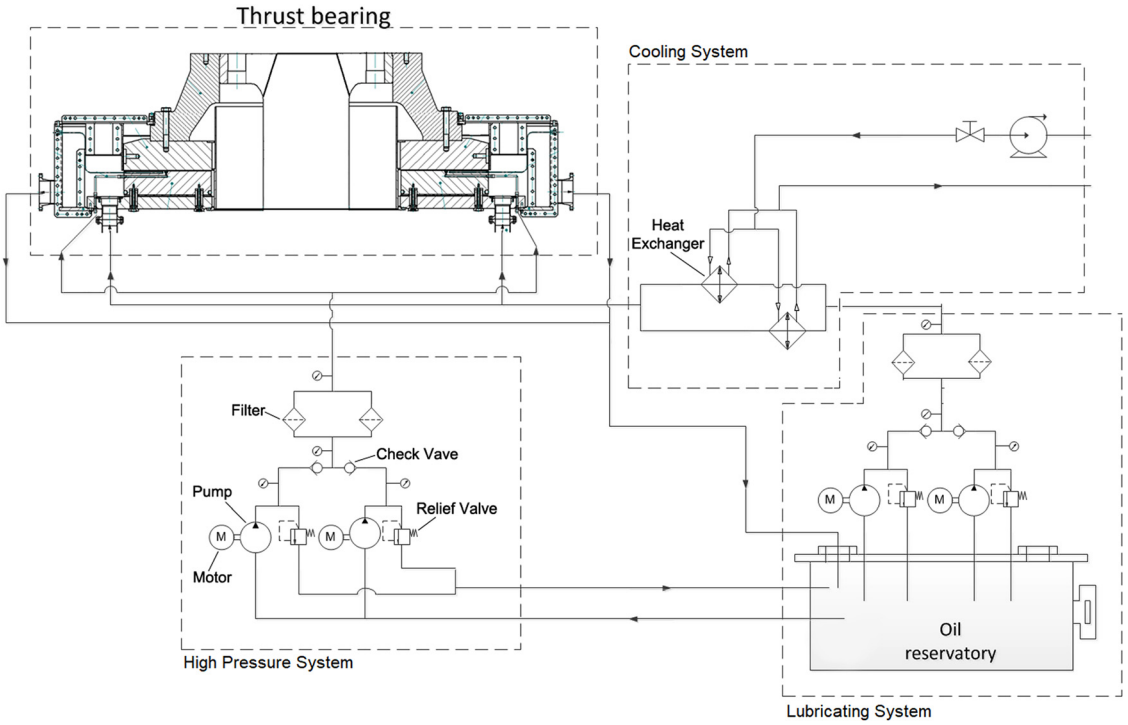

Fig. 22 Hydraulic turbine thrust bearing system

management. For each analysis to be developed on a system, however, there was not a single source of information. Now, with the use of SysML, system knowledge is unified. Section 4 shows a case study of the presented method.

4 A Case Study: Water Storage System

To illustrate how the proposed method works and what are some of its capabilities, it is applied in a simple water storage system. The system under consideration consists of a tank ($T1$),

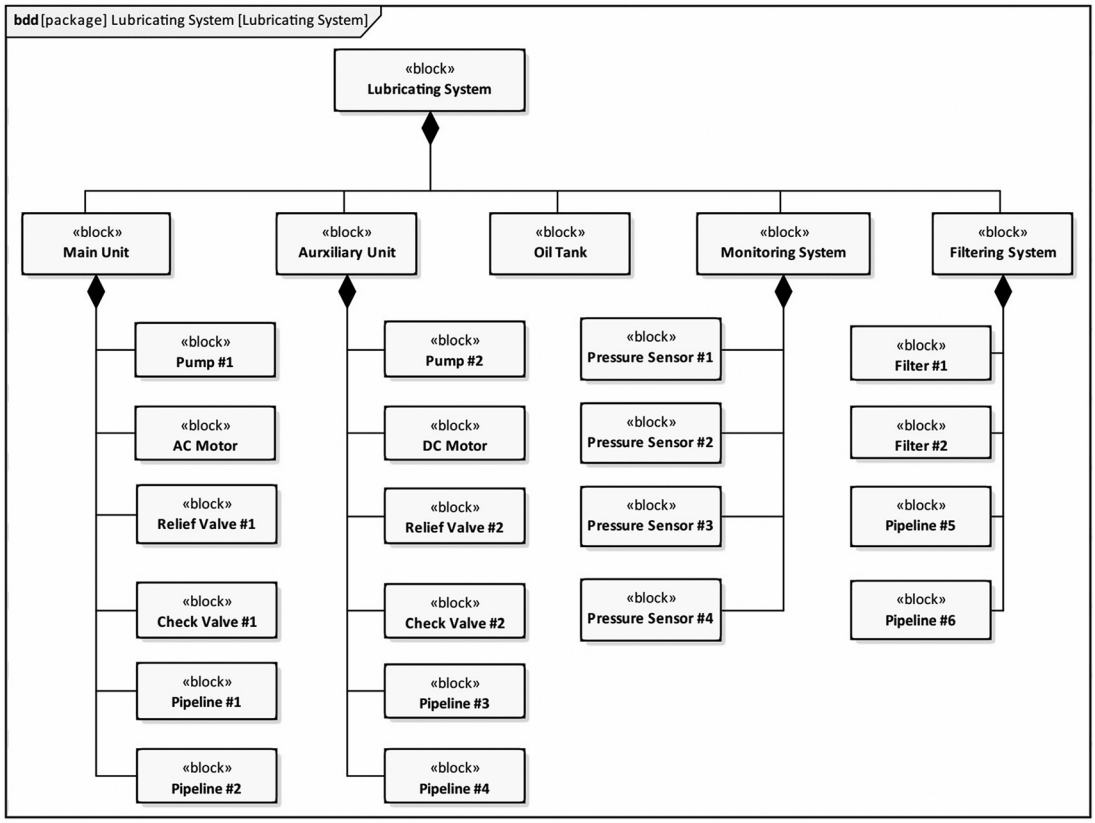

Fig. 23 BDD of lubricating system

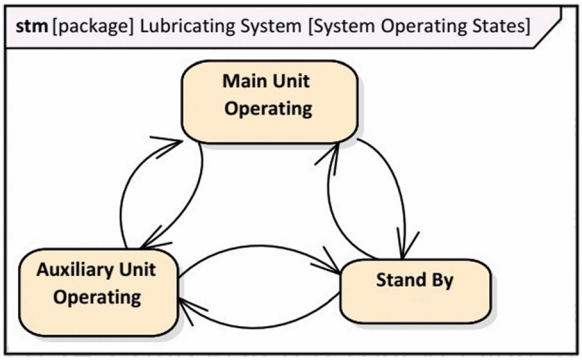

Fig. 24 STM of lubricating system

responsible for storing water, and a valve ($V1$) and two pipes ($P1$ and $P2$), responsible for distributing the water present in the tank when necessary. A water level sensor in the tank (L-1) and a water flow sensor (F-1) in the $P2$ pipeline are responsible for monitoring the system. Figure 15 illustrates the described system.

The first step for the development of the method is the creation of the SysML diagrams of the described system. Figure 16 shows the BDD of the system, while Fig. 17 shows its STM. The system has three states of operation: "storing water," "receiving water," and "delivering water," and a BN for each of them must be obtained, although in this work only the Bayesian network for the delivering water mode is presented. Enterprise Architect was the software used to model the SysML diagrams [58].

Figure 18 shows the STM for the flow sensor. A similar diagram has been developed for the level sensor. Figure 19 shows the ACT for the delivering water operational state, which starts with the operator opening valve 1. The water output system then conducts the water provided by the tank out of the system. By being responsible for the "open valve" action, the operator is also inserted into the Bayesian network as a parent node. This means that one of the features of the proposed method is to make it possible to consider human error as the cause of a possible system failure. Figure 20 shows as an example the valve BDD.

After the necessary diagrams for the method having been constructed, the relationship matrices are extracted, as shown in Tables 6 and 7. Such tables will support the construction of the BN graph, according to the previously presented method steps. Figure 21 shows the graph obtained from these tables. Genie Modeler was the software used to model the Bayesian network [59].

Once the BN graph has been obtained, the analyst must complete the CPTs. For parent nodes, i.e., failure mode nodes, the reliability values of each failure mode are required. With the exception of operator reliability, these values were calculated by Eq. (8) using failure rates (λ) obtained from the database [55], considering one year of operation. The reliability of the components which need to be inserted in the respective CPTs is presented in Table 8.

$$R = e^{-\lambda t} \tag{8}$$

Regarding the operator's failure to perform his task, several authors propose different methods for the calculation of human reliability [60–64]. However, diagnostic methods generally disregard human errors as the cause of system failures. This happens either because of a limitation of the method used or because such an error is not relevant to the analysis. The proposed method is able to consider human error in the diagnosis process, since SysML can also represent human actions necessary for the system to properly work. In this example, it has been assumed an operator's failure rate, shown in Table 8, so that the method can be fully presented. For an actual application, it is recommended to use a method for acquiring such a probability.

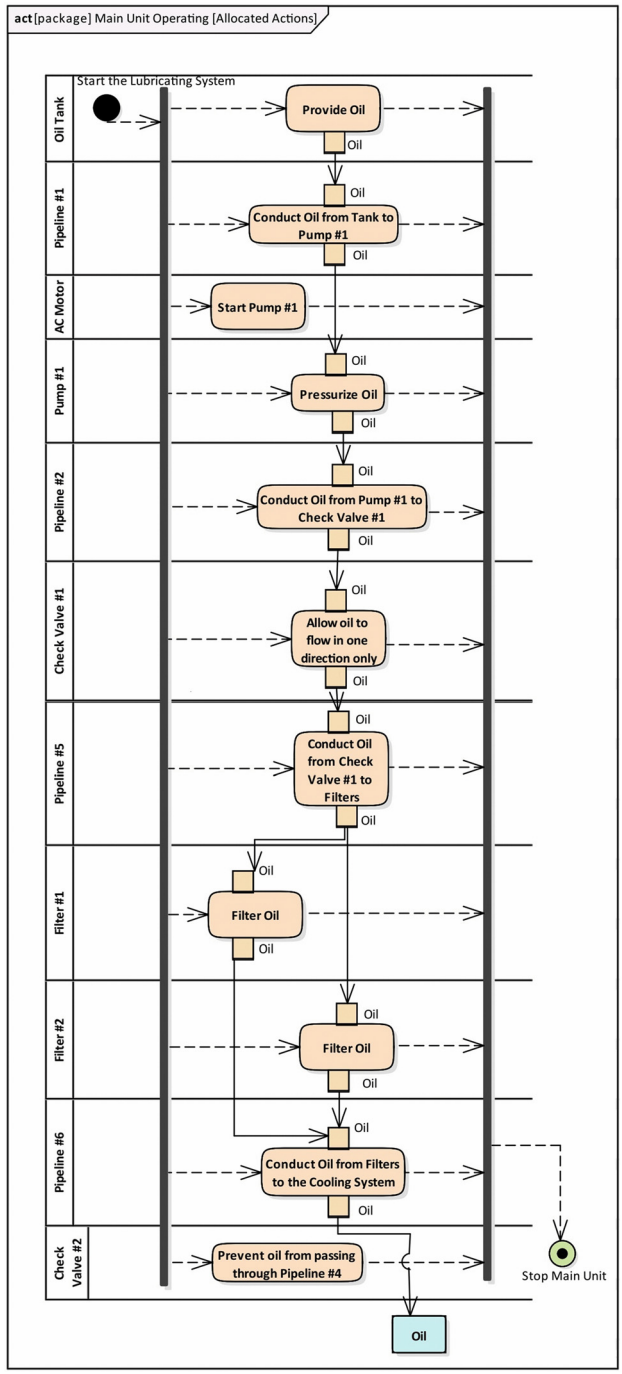

Fig. 25 ACT diagram for the main unit operating state

Table 9 shows, for each possible combination of system sensor readings, the diagnostic results obtained by the BN through a posteriori probability calculation. For example, if the flow sensor has a normal reading (as expected), but the level sensor has a lower than expected reading, the BN indicates that a leak in the water tank is most likely (with a posterior probability of 85.2%).

It can be seen that if any of the sensors has a reading incompatible with any of the component failure modes, BN indicates that the sensor is at fault. For example, if the flow sensor shows a higher than expected reading, its failure probability is 100%, as no failure mode of the other components would imply this reading.

In addition, it can be noted that some failure modes always have the same posterior probability, no matter the combination of the sensor readings. Leakages from both pipelines, as well as clogging from both of them, for example, have always the same

outcome. This happens because such faults have the same effects on sensors as well as the same reliability. This means that the current monitoring system cannot differentiate between these faults. For this distinction to be made, it is necessary that more sensors, in different positions or measuring different parameters, be added to the system.

If a given fault is never shown to be the most likely to have occurred by BNs a posteriori probability calculation, it means that the monitoring system is not able to differentiate it from other faults. Knowing these results, it is up to the designer or system operator to decide whether the monitoring system needs to be updated so that such a fault can be properly observed. If the occurrence of such a failure has serious consequences, whether financial, environmental, or in personnel safety, such update is of the utmost importance.

5 A Case Study: Lubricating System of Thrust Bearing

To emphasize the generality of the proposed method, it is applied in a lubricating system of hydraulic turbine thrust bearing used in hydro-electric power plant from São Paulo state in Brazil. The hydro-electric power plant has a generation capacity of 120 MW.

Thrust bearings or hydrodynamic bearings operate on the principle of hydrodynamic lubrication and are used to carry loads in applications where roller bearings are unsuitable due to dimensional limitations, demands for operational lifespan, or high loading requirements. In it, the load carrying surfaces are completely separated by an oil film, eliminating the risk of surface wear as long as a film of sufficient thickness is maintained.

Figure 22 shows the lubricating and cooling systems of hydraulic turbine thrust bearing, which is composed by an oil reservoir and three other subsystems: the high-pressure system, the lubricating system, and the cooling system. The high-pressure system ensures the formation of the oil film during starting and stopping of the generating unit by elevating oil pressure, avoiding the bearing metal-to-metal contact anchor. The lubricating system ensures the formation of the oil film between stator and rotor during all the steady-state operation of the turbine, when the oil pressure needed is not as high as in the starting and stopping of the unit. Finally, the cooling system is responsible for refrigerating the oil that circulates in the system through the use of heat exchangers.

The proposed method is applied only on the lubricating system. There are two oil pumps in the lubrication system, the main and the auxiliary. Only one of them should be on when the system is in operation. In addition, the system also has relief valves to prevent pipeline overpressure and check valves to drive oil into the bearing.

The first step for the development of the method is the creation of the SysML diagrams of the described system. Figure 23 shows the BDD of the system, while Fig. 24 shows its STM. The system has three operating states: "main unit operating," "auxiliary unit operating," and "stand by," which means a BN for each of them must be obtained. In this work, though, only the BN for the "main unit operating" state is presented.

Figure 25 shows the ACT for the "main operating unit" state, which presents the function of each of the components working in this operating state.

After also building the STM of each sensor and the BDD for each component, the relationship matrices are extracted and the BN graph is built, as shown in Fig. 26. Once the BN graph has been obtained, the analyst must complete the CPTs. For parent nodes, i.e., failure mode nodes, the reliability values of each failure mode are required. These values, calculated similarly to the previous case study, are presented in Table 10.

Since the system has four sensors, each with four possible reading states, there are 256 possible combinations of sensor readings that can be used for a posteriori probability calculation. Table 11 shows, for some of the possible combination of system sensor readings, the diagnostic results obtained by the BN through a posteriori probability calculation. For example, if all four pressure sensors present a normal reading (as expected), BN indicates that

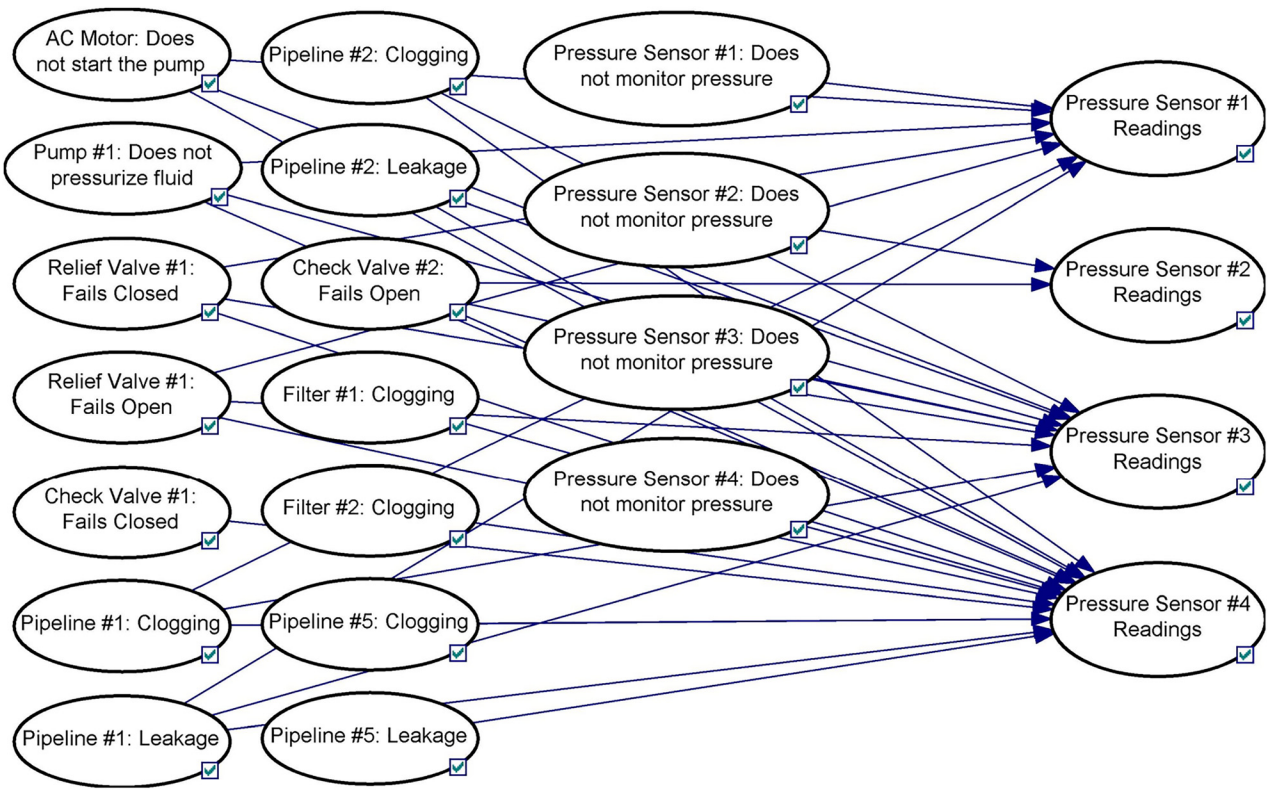

Fig. 26 BN graph for the main unit operating state

Table 10 Reliability and failure probability for the CPTs

		λ	True	False
Oil tank	Leakage	2.00×10^{-6}	0.017367	0.982633
AC motor	Does not start the pump	2.10×10^{-5}	0.168031	0.831969
Pump #1	Does not pressurize fluid	1.70×10^{-5}	0.138362	0.861638
Relief valve #1	Fails closed	1.00×10^{-5}	0.083873	0.916127
	Fails open	1.00×10^{-5}	0.083873	0.916127
Filter #1	Clogging	3.10×10^{-5}	0.23781	0.76219
Check valve #1	Fails closed	8.00×10^{-6}	0.067681	0.932319
	Fails open	8.00×10^{-6}	0.067681	0.932319
Pipeline #1	Clogging	1.00×10^{-6}	0.008722	0.991278
	Leakage	1.50×10^{-5}	0.123133	0.876867
Pipeline #2	Clogging	1.00×10^{-6}	0.008722	0.991278
	Leakage	1.50×10^{-5}	0.123133	0.876867
Pressure sensor #1	Does not monitor pressure	1.80×10^{-6}	0.015644	0.984356
Pressure sensor #2	Does not monitor pressure	1.80×10^{-6}	0.015644	0.984356
Pressure sensor #3	Does not monitor pressure	1.80×10^{-6}	0.015644	0.984356
Pressure sensor #4	Does not monitor pressure	1.80×10^{-6}	0.015644	0.984356

Table 11 Posterior probabilities of the TRUE state of each failure mode

Pressure sensor #1 readings		NORMAL	NORMAL	NORMAL	HIGHER
Pressure sensor #2 readings		NORMAL	NORMAL	HIGHER	NORMAL
Pressure sensor #3 readings		NORMAL	NORMAL	LOWER	HIGHER
Pressure sensor #4 readings		NORMAL	LOWER	LOWER	HIGHER
AC motor	Does not start the pump	1.14×10^{-7}	1.85×10^{-5}	0.00167524	0.00056489
Pump #1	Does not pressurize fluid	9.41×10^{-8}	1.52×10^{-5}	0.00137944	0.00046515
Relief valve #1	Fails closed	5.53×10^{-7}	4.58×10^{-5}	0.00217702	0.99999597
	Fails open	2.45×10^{-7}	2.51×10^{-5}	0.00157734	0.01132349
Check valve #1	Fails closed	0.00075378	0.1271972	0.06769114	0.04785644
Pipeline #1	Clogging	2.65×10^{-8}	2.70×10^{-6}	0.00016818	0.00123947
	Leakage	3.74×10^{-7}	3.82×10^{-5}	0.0023744	0.01749873
Pipeline #2	Clogging	5.96×10^{-7}	0.00010013	0.00926188	0.00343954
	Leakage	8.41×10^{-6}	0.00141365	0.13075869	0.04855915
Check valve #2	Fails open	2.61×10^{-8}	4.39×10^{-6}	0.99005974	0.00015081
Filter #1	Clogging	0.00264855	0.44693379	0.2378469	0.16815355
Filter #2	Clogging	0.00264855	0.44693379	0.2378469	0.16815355
Pipeline #5	Clogging	9.71×10^{-5}	0.01639138	0.00872308	0.00616706
	Leakage	0.00137136	0.23141253	0.12315192	0.08706622
Pressure sensor #1	Does not monitor pressure	0.00527006	0.0053181	0.00963597	0.00655336
Pressure sensor #2	Does not monitor pressure	0.00526977	0.00527411	0.01515761	0.00541975
Pressure sensor #3	Does not monitor pressure	0.00527936	0.00684549	0.00545421	0.00568214
Pressure sensor #4	Does not monitor pressure	0.01113723	0.00990403	0.0052819	0.00741729

all failure modes have a very low probability of happening. If only the fourth sensor presents a lower than expected reading, however, BN indicates that the most probable scenario for that to happen is a clogged filter, since both filters have a posteriori probability of 44.7%. On a different scenario, if the first sensor presents a normal reading, but the second one shows a higher than expected reading and the last two show a lower than expected reading, BN indicates that there is a 99.9% probability of check valve #2 failing open. Finally, if all sensors present a higher than expected reading except for the second one, BN indicates a fault in the relief valve, with a 99.9% of a posteriori probability.

By analyzing the results obtained by the method, it is possible to notice some limitations of the monitoring system. For example, while the oil tank plays an important role in the operation of the lubricating system, its failure cannot be identified by the online sensors present in the system. For this to happen, an oil level sensor could be installed. Diagnostic accuracy in this case is limited by the lack of information obtained by system monitoring. The results of the method can therefore contribute to decision-making regarding the number and types of sensors to be installed in a system. The designer of a new system or the operator of an existing one can decide whether an improvement in monitoring can contribute to the diagnosis of critical failures based on the results of the method.

6 Conclusions

The proposed method translates SysML diagrams into BN graphs by means of a novel structured procedure. The graphs obtained are used to perform system fault diagnosis by calculating posterior probabilities.

One of the great advantages of the proposed method is the possibility of developing a fault diagnosis process during the initial phases of a project, since SysML is widely used during the design of new and complex systems. The resulting BNs will allow the designer to identify, during a certain mode of operation, which component is at fault given the reading of the sensors.

It should be noted, however, that the contributions of the method are not limited to the design phase of the system. It can also be useful when it is necessary to implement a fault diagnosis process in an already operating system. In addition, it can also contribute to an assessment regarding the need to upgrade the original system in order to make it better monitored.

Since the mapping process is automated, a software that accomplishes this task can be developed, drastically reducing the time of implementation. Such software can, in addition to streamlining the mapping process, provide an interface that helps the system expert refine BN conditional probability tables.

In order to test the adaptability of the method, the authors intend to apply it to different systems, evaluating its effectiveness in different situations. In addition, future work will explore the ability to perform other types of system analysis, not limited to Bayesian networks or fault diagnosis using SysML system representation. The main purpose of SysML is to concentrate all knowledge about a system in one place. Through its nine diagrams, it is possible to represent how a system works, what components it is made of, how it interacts with its operators and maintainers, how it behaves under different operating conditions, etc. If used correctly, the authors believe that not only fault diagnosis, but other types of analysis can be developed using the information represented in SysML, such as reliability, availability, maintainability, efficiency, risk management, and asset management.

For each analysis to be developed on a system, however, there was not a single source of information. Now, with the use of SysML, system knowledge is unified. The proposed method shows that it is possible to perform fault diagnosis through SysML and the authors intend, in future works, to present other system analysis to be done using SysML, showing how adaptable and comprehensive is this technique.

Funding Data

- Fundação para o Desenvolvimento Tecnológico da Engenharia (FDTE).
- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) (Finance Code 001) (Funder ID: 10.13039/501100002322).

References

[1] Mobley, R. K., 2002, *An Introduction to Predictive Maintenance*, 2nd ed., Butterworth-Heinemann, Woburn, MA.

[2] Papadopoulos, Y., and McDermid, J., 2001, "Automated Safety Monitoring: A Review and Classification of Methods," Int. J. Cond. Monit. Diagn. Eng. Manag., **4**(4), pp. 1–32.

[3] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., and Yin, K., 2003, "A Review of Process Fault Detection and Diagnosis—Part I: Quantitative Model Based Methods," Comput. Chem. Eng., **27**(3), pp. 293–311.

[4] Lampis, M., 2010, *Application of Bayesian Belief Networks to System Fault Diagnostics*, Loughborough University, Loughborough, UK.

[5] Neapolitan, R. E., 2004, *Learning Bayesian Networks*, 1st ed., Prentice Hall, Upper Saddle River, NJ.

[6] Delligatti, L., 2013, *SysML Distilled: A Brief Guide to the Systems Modeling Language*, 1st ed., Pearson Education, Upper Saddle River, NJ.

[7] Cai, B., Huang, L., and Xie, M., 2017, "Bayesian Networks in Fault Diagnosis," IEEE Trans. Ind. Inf., **13**(5), pp. 2227–2240.

[8] Isermann, R., 2011, "Fault-Diagnosis Applications - Model-Based Condition Monitoring: Actuators," *Drives, Machinery, Plants, Sensors, and Fault-Tolerant Systems*, 1st ed., Springer Berlin Heidelberg, Berlin, Heidelberg, Germany.

[9] Chen, Q., Ahmed, Q., Rizzoni, G., Frisk, E., and Zhai, H., 2015, "Model-Based Fault Diagnosis of an Automated Manual Transmission Shifting Actuator," IFAC-PapersOnLine, **28**(21), pp. 1479–1484.

[10] Zhang, J., Rizzoni, G., Cordoba-Arenas, A., Amodio, A., and Aksun-Guvenc, B., 2017, "Model-Based Diagnosis and Fault Tolerant Control for Ensuring Torque Functional Safety of Pedal-by-Wire Systems," Control Eng. Pract, **61**, pp. 255–269.

[11] Henry, D., Peuvedic, C. L., Strippoli, L., and Ankersen, F., 2015, "Robust Model-Based Fault Diagnosis of Thruster Faults in Spacecraft," IFAC-PapersOnLine, **28**(21), pp. 1078–1083.

[12] Mulumba, T., Afshari, A., Yan, K., Shen, W., and Norford, L. K., 2015, "Robust Model-Based Fault Diagnosis for Air Handling Units," Energy Build., **86**, pp. 698–707.

[13] Dey, S., Biron, Z. A., Tatipamula, S., Das, N., Mohon, S., Ayalew, B., and Pisu, P., 2016, "Model-Based Real-Time Thermal Fault Diagnosis of Lithium-Ion Batteries," Control Eng. Pract., **56**, pp. 37–48.

[14] Chen, Z., Xiong, R., Tian, J., Shang, X., and Lu, J., 2016, "Model-Based Fault Diagnosis Approach on External Short Circuit of Lithium-Ion Battery Used in Electric Vehicles," Appl. Energy, **184**, pp. 365–374.

[15] Gao, Z., Cecati, C., and Ding, S. X., 2015, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis Model-Based and Signal-Based Approaches," IEEE Trans. Ind. Electron., **62**(6), pp. 3757–3767.

[16] Chen, H., and Lu, S., 2013, "Fault Diagnosis Digital Method for Power Transistors in Power Converters of Switched Reluctance Motors," IEEE Trans. Ind. Electron., **60**(2), pp. 749–763.

[17] Freire, N. M. A., Estima, J. O., and Marques Cardoso, A. J., 2013, "Open-Circuit Fault Diagnosis in PMSG Drives for Wind Turbine Applications," IEEE Trans. Ind. Electron., **60**(9), pp. 3957–3967.

[18] Feng, Z., and Zuo, M. J., 2013, "Fault Diagnosis of Planetary Gearboxes Via Torsional Vibration Signal Analysis," Mech. Syst. Signal Process, **36**(2), pp. 401–421.

[19] Hong, L., and Dhupia, J. S., 2014, "A Time Domain Approach to Diagnose Gearbox Fault Based on Measured Vibration Signals," J. Sound Vib., **333**(7), pp. 2164–2180.

[20] Gao, Z., Cecati, C., and Ding, S. X., 2015, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis With Knowledge-Based and Hybrid/Active Approaches," IEEE Trans. Ind. Electron., **62**(6), pp. 3768–3774.

[21] Mostafa, S. A., Ahmad, M. S., Mohammed, M. A., and Obaid, O. I., 2012, "Implementing an Expert Diagnostic Assistance System for Car Failure and Malfunction," Int. J. Comput. Sci. Issues, **9**(2), pp. 1–7.

[22] Nan, C., Khan, F., and Iqbal, M. T., 2008, "Real-Time Fault Diagnosis Using Knowledge-Based Expert System," Process Saf. Environ. Prot, **86**(1), pp. 55–71.

[23] Toffolo, A., and Lazzaretto, A., 2008, "Energy System Diagnosis by a Fuzzy Expert System With Genetically Evolved Rules," Int. J. Thermodyn., **11**(3), pp. 115–121.

[24] Case, K., Nor, A., and Teoh, P. C., 2010, "A Diagnostic Service Tool Using FMEA," Int. J. Comput. Integr. Manuf, **23**(7), pp. 640–654.

[25] Barkai, J., 1999, "Automatic Generation of a Diagnostic Expert System From Failure Mode and Effects Analysis (FMEA) Information," SAE Paper No. 1999-01-0060.

[26] Price, C., and Taylor, N., 1997, "Multiple Fault Diagnosis From FMEA," Proceedings of the 14th National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, Providence, RI, July 27–31, pp. 1052–1057.

[27] Hurdle, E. E., Bartlett, L. M., and Andrews, J. D., 2009, "Fault Diagnostics of Dynamic System Operation Using a Fault Tree Based Method," Reliab. Eng. Syst. Saf., **94**(9), pp. 1371–1380.

[28] Contini, S., Cojazzi, G. G. M., and Renda, G., 2008, "On the Use of Non-Coherent Fault Trees in Safety and Security Studies," Reliab. Eng. Syst. Saf., **93**(12), pp. 1886–1895.

[29] Hu, J., Zhang, L., Cai, Z., and Wang, Y., 2015, "An Intelligent Fault Diagnosis System for Process Plant Using a Functional HAZOP and DBN Integrated Methodology," Eng. Appl. Artif. Intell., **45**, pp. 119–135.

[30] Hidalgo, E. M. P., and de Souza, G. F. M., 2015, "The Application of Risk and Reliability Techniques to Acquire Knowledge in the Development of Expert System for Fault Diagnosis," XXIV European Safety and Reliability Conference ESREL, Wroclaw, Poland, Sept. 14–18, pp. 825–834.

[31] Pearl, J., 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1st ed., Morgan Kaufman, San Francisco, CA.

[32] Russell, S., and Norvig, P., 2010, *Artificial Intelligence a Modern Approach*, 3rd ed., Prentice Hall, Upper Saddle River, NJ.

[33] Maturana, M. C., 2010, "Aplicação de Redes Bayesianas na Análise da Contribuição Do Erro Humano em Acidentes de Colisão," Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil.

[34] Lin, X., Cheng, B., and Chen, J., 2010, "Context-Aware End-to-End QoS Qualitative Diagnosis and Quantitative Guarantee Based on Bayesian Network," Comput. Commun., **33**(17), pp. 2132–2144.

[35] Yan, Z., and Peng, L., 2011, *Application of Bayesian Network in Failure Diagnosis of Hydro-Electrical Simulation System*, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, pp. 691–698.

[36] Jin, S., Liu, Y., and Lin, Z., 2012, "A Bayesian Network Approach for Fixture Fault Diagnosis in Launch of the Assembly Process," Int. J. Prod. Res., **50**(23), pp. 6655–6666.

[37] Mechraoui, A., Medjaher, K., and Zerhouni, N., 2008, "Bayesian Based Fault Diagnosis: Application to an Electrical Motor," IFAC Proc. **41**, pp. 7381–7386.

[38] Xu, B. G., 2012, "Intelligent Fault Inference for Rotating Flexible Rotors Using Bayesian Belief Network," Expert Syst. Appl., **39**(1), pp. 816–822.

[39] Bobbio, A., Portinale, L., Minichino, M., and Ciancamerla, E., 2001, "Improving the Analysis of Dependable Systems by Mapping Fault Trees Into Bayesian Networks," Reliab. Eng. Syst. Saf., **71**(3), pp. 249–260.

[40] Chiremsel, Z., Said, R. N., and Chiremsel, R., 2016, "Probabilistic Fault Diagnosis of Safety Instrumented Systems Based on Fault Tree Analysis and Bayesian Network," J. Fail. Anal. Prev., **16**(5), pp. 747–760.

[41] Lampis, M., and Andrews, J. D., 2009, "Bayesian Belief Networks for System Fault Diagnostics," Qual. Reliab. Eng. Int., **25**(4), pp. 409–426.

[42] Lo, C. H., Wong, Y. K., and Rad, A. B., 2011, "Bond Graph Based Bayesian Network for Fault Diagnosis," Appl. Soft Comput. J., **11**(1), pp. 1208–1212.

[43] INCOSE, 2015, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4th ed., Wiley, Hoboken, NJ.
[44] ISO/IEC/IEEE, 2015, "Systems and Software Engineering—System Life Cycle Processes," Standard No. ISO/IEC/IEEE15288, New Delhi, India.
[45] Friedenthal, S., Moore, A., and Steiner, R., 2015, *A Practical Guide to SysML: The Systems Modeling Language*, 3rd ed., Morgan Kaufmann, Waltham, MA.
[46] Debbabi, M., Hassaïne, F., Jarraya, Y., Soeanu, A., and Alawneh, L., 2010, *Verification and Validation in Systems Engineering*, 1st ed., Springer Berlin Heidelberg, Berlin, Heidelberg, Germany.
[47] Helle, P., 2012, "Automatic SysML-Based Safety Analysis," Proceedings of the Fifth International Workshop on Model Based Architecting and Construction of Embedded Systems—ACES-MB '12, Innsbruck, Austria, Sept. 30, pp. 19–24.
[48] David, P., Idasiak, V., and Kratz, F., 2010, "Reliability Study of Complex Physical Systems Using SysML," Reliab. Eng. Syst. Saf., **95**(4), pp. 431–450.
[49] Hecht, M., Dimpfl, E., and Pinchak, J., 2014, "Automated Generation of Failure Modes and Effects Analysis From SysML Models," IEEE 25th International Symposium on Software Reliability Engineering, Naples, Italy, Nov. 3–6, pp. 62–65.
[50] Yakymets, N., Jaber, H., and Lanusse, A., 2013, "Model-Based System Engineering for Fault Tree Generation and Analysis," Proceedings First International Conference on Model-Driven Engineering and Software Development, Barcelona, Spain, Feb. 19–Feb. 21, pp. 210–214.
[51] Mhenni, F., Nguyen, N., and Choley, J. Y., 2014, "Automatic Fault Tree Generation From SysML System Models," IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Besacon, France, July 8–11, pp. 715–720.
[52] Mhenni, F., Nguyen, N., and Choley, J., 2018, "SafeSysE: A Safety Analysis Integration in Systems Engineering Approach," IEEE Syst. J., **12**(1), pp. 161–172.
[53] Suiphon, B., Simeu-Abazi, Z., and Gascard, É., 2014, "Premiers Pas Vers le Diagnostic de Défaillances Par Exploitation d 'un Modèle SysML—First Steps Toward Fault Diagnosis Exploiting a SysML Model," 19° Congrès de Maîtrise Des Risques et Sûreté de Fonctionnement, Dijon, France, Oct. 21–23, pp. 1–10.
[54] Melani, A. H. A., Martha de Souza, G. F., Murad, C., Caminada Netto, A. A., and Nabeta, S. I., 2017, "Petri Net Based Reliability Analysis of Thermoelectric Plant Cooling Tower," Proceedings of the 24th ABCM International Congress of Mechanical Engineering, Curitiba, Brazil, Dec. 3–8.
[55] OREDA, 2002, *Offshore Reliability Data Handbook*, 4th ed., Det Norske Veritas, Hovik, Norway.
[56] NPRD, 1995, *Nonelectronic Parts Reliability Data*, 3rd ed., Reliability Analysis Center, Rome, NY.
[57] Cressent, R., David, P., Idasiak, V., and Kratz, F., 2013, "Designing the Database for a Reliability Aware Model-Based System Engineering Process," Reliab. Eng. Syst. Saf., **111**, pp. 171–182.
[58] SparxSystems, 2019, "Enterprise Aschitect," SparxSystems, Victoria, Australia, accessed Jan. 16, 2019, https://sparxsystems.com/products/ea/
[59] BayesFusion, 2019, "Genie Modeler," BayesFusion, Pittsburgh, PA, accessed Jan. 16, 2019, https://www.bayesfusion.com/genie/
[60] Pasquale, V. D., Miranda, S., Iannone, R., and Riemma, S., 2015, "A Simulator for Human Error Probability Analysis (SHERPA)," Reliab. Eng. Syst. Saf, **139**, pp. 17–32.
[61] Mandal, S., Singh, K., Behera, R. K., Sahu, S. K., Raj, N., and Maiti, J., 2015, "Human Error Identification and Risk Prioritization in Overhead Crane Operations Using HTA, SHERPA and Fuzzy VIKOR Method," Expert Syst. Appl., **42**(20), pp. 7195–7206.
[62] Pyy, P., 2001, "An Analysis of Maintenance Failures at a Nuclear Power Plant," Reliab. Eng. Syst. Saf., **72**(3), pp. 293–302.
[63] De Felice, F., Petrillo, A., and Zomparelli, F., 2016, "A Hybrid Model for Human Error Probability Analysis," IFAC-PapersOnLine, **49**(12), pp. 1673–1678.
[64] Castiglia, F., Giardina, M., and Tomarchio, E., 2015, "THERP and HEART Integrated Methodology for Human Error Assessment," Radiat. Phys. Chem., **116**, pp. 262–266.