DEMDE: Decision Making Design based on Bayesian Network for Personalized Monitoring System

1st names
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Personalized monitoring systems (PMS) are used for Decision Making (DeM) to support humans and fully autonomous decision-making in several applications, such as health monitoring and management. However, it is still challenging to design decision-making in PMS. In this work, we propose a systematic modeling approach, called DEMDE, for decision-making design in PMS during the design phase. DEMDE guides the design of a specific Bayesian network (BN) from an instantiated domain model for PMS based on context-aware data fusion using a probabilistic domain model (general BN). We evaluated our proposal by developing a BN for decision-making about sending an alert of high contamination risk in cell culture. The case study demonstrated the application of the DEMDE concepts and the modeling process, including model evaluation through sensitivity analysis to assess the robustness and reliability of the modeled BN.**

***Index Terms*—Probabilistic domain model, Bayesian network, Personalized monitoring system, Context-aware data fusion**

I. Introduction

Personalized Monitoring Systems (PMS) capture and process data with the goal of taking actions, such as detecting, predicting/estimating and making decisions, in response to events that can affect a target entity. In particular, these actions are performed from knowledge about the preferences and behaviours that a target entity can express as specific patterns that characterize it in an environment [1]. PMS, as well as Digital Twins (DT), are used for Decision Making (DeM) to support not only humans but also fully autonomous decision-making [2], [3]. In case of complex system such as PMS and DT, DeM is an important functional requirement that should be well addressed in the early stages of development [3]–[7], specially during the design phase [8]–[10]. This is supported by six key reasons: i) Effective communication: Clear and effective communication during the design phase ensures that all stakeholders have a shared understanding of the system requirements and goals, preventing misunderstandings and ensuring the system meets their needs; ii) Complete understanding: Gaining a complete understanding of the system's requirements and goals allows for a system design that meets the needs of all stakeholders and prevents costly changes and rework during development; iii) Cost reduction: Identifying issues early in the design phase helps to reduce development costs by preventing the need for costly changes later in the process; iv) behaviour prediction: Predicting system behaviour during the design phase enables the identification of potential issues and ensures the system meets the needs of all stakeholders; v) Reuse: Reusing components or designs from previous projects helps to reduce development time and costs while ensuring the system meets stakeholder needs; and vi) Feasibility analysis: Conducting a feasibility analysis during the design phase helps to identify potential issues and ensures the system is both financially and practically feasible.

Despite of benefices described above, it is challenging and time-intensive work to address DeM as a functional requirement in the design phase. It is because PMS represents the convergence of heterogeneous technologies pertaining to different engineering areas and requires more than just integrating a few devices and mathematical models within the four walls of a company or laboratory [1]. Instead, they require the use of methodologies that enable the cooperation and interoperability between different stakeholders, potentially involved in very different application fields, often called silos [6], [11], [12]. However, to the best of our knowledge, there is a lack of specifically tailored approaches for modelling in high level the DeM performed by PMS or DT [6].

Two methods that can be combined to enable the design of the decision-making capabilities of PMS are Domain Model (DM) and Bayesian Networks (BN) [13], [14]. A DM is a representation of the objects, concepts, and relationships within a specific domain and is crucial in defining system requirements and constraints, aiding developers in understanding the problem space in the early stage of system development [1], [12], [15], [16]. On the other hand, BN is a probabilistic model that represents a set of variables and their conditional dependencies via a directed acyclic graph. BN is a versatile and powerful framework to model complex systems and for reasoning and decision-making under uncertainty [17]. The integration of these methods is done by extracting BN from the DM to provide probabilistic inference in decision-making and this is needed because BN helps capture the probabilistic relationships between variables, while DM represents the structure and behavior of the system being modeled. Designing decision-making in a PMS using Bayesian networks extracted from a domain model can have five possibles benefits: i) Accurate representation of the system: a domain model provides a comprehensive understanding of the system, which can be used to build a Bayesian network. This network captures the relationships and dependencies between the various elements of the system, allowing for accurate decision making [13]; ii) Efficient and effective decision making: Bayesian networks provide an efficient and effective way to make decisions in complex systems. They can consider many variables and dependencies and provide probabilistic estimates of the outcomes of various decisions [14]; iii) Adaptive decision making: PMS can learn from data and adapt to changes in the system. A Bayesian network can be updated with new data, allowing the system to make better decisions over time; iv) Transparency and explainability: Bayesian networks are transparent and explainable, allowing users to understand how decisions are made. This can be especially important in safety-critical systems [13]; and v) Risk management: Bayesian networks can assess and manage risk in complex systems. Understanding the dependencies between different variables makes it possible to identify potential risks and take appropriate measures to mitigate them.

Together, DM and BN can be powerful tools for designing decision-making systems that are both effective and efficient, and the DM proposed in [1] for PMS based on context-aware data fusion can be employed for this purpose. Therefore, in this work, we proposed a general modeling approach to DEcision Making DEsign (called DEMDE) in PMS during the design phase of the systems development life cycle (early stage of development). It is a systematic procedure that enables specific BN development from an instantiated domain model for PMS using a probabilistic domain model (general BN). Our proposal was evaluated by modeling a Bayesian network for decision-making in PMS to alert about the possibility of contamination in cell cultures. The modeled case illustrated the DEMDE concepts and all steps of the modeling process, including the model evaluation step performed with sensitivity analysis. It evaluated the robustness and reliability of modeled BN by assessing the impact of changes in the probability distributions of the BN nodes on the beliefs in the target node. In summary, the contributions in this paper are: (i) A probabilistic domain model (general BN) that illustrates the concepts of context-aware data fusion as a set of random variables and their conditional dependencies by the directed acyclic graph. In addition, to the best of our knowledge, this is the first probabilistic domain model to DeM in PMS; and (ii) A modeling process that ensures a structured and systematic approach to build a specific BN leading to better communication among stakeholders and reducing the risk of misunderstandings and errors.

The rest of the paper is organized as follows. Section II discusses some preliminary knowledge about the domain model and Bayesian network. In Section III, the proposed modeling approach is described and evaluated in Section IV. The related works are presented in Section V, followed by a discussion and conclusion in Sections VI and VII.

II. Background

In this section, we will provide an overview of the Domain Model for Personalized Monitoring System (DM-PMS) and Bayesian Network.

A. Domain Model for Personalized Monitoring System (DM-PMS)

The domain model proposed in [1] comprises 25 concepts. Here, we describe only the four core concepts needed to develop the probabilistic domain model. More details can be viewed in [1]. The first concept is **Environment**. A physical or virtual space that enables contexts to happen and has the participation, or it is performed by the Target Entity that inhabits or interacts with the environment. The second one is **Context**. A relevant situation (dynamic data source) that can provide information either to condition expectations or improve the understanding of a given inference or management problem [18]. The third one is **Target Entity**. Any entity (e.g., car, human, animal, etc.) that can interact and express patterns in a context that can occur in an environment. The last one is **Event**. It can be any situation that can affect a target entity and makes it express anomalies in a context. The relationships between these concepts can be viewed in Figure 1. We can see that a Target Entity interacts/inhabits some environment, enabling contexts in which a target entity participates/performs. Furthermore, an Event affects a Target Entity causing it to express anomalies when participating in a Context; in other words, an event makes a Target Entity generate anomaly values in a dynamic context variable since context is a dynamic data source that can be used as a problem variable and/or an ancillary variable related to some events. If an Event does not happen, an anomaly does not occur, and the dynamic context variable will continue to present values in a particular pattern. As context can be expressed as a variable, it can contain patterns and anomalies that the Data Fusion Resource can use to detect, identify, estimate, and predict an event in a PMS. Therefore, we have that a dynamic context variable can be used as a problem variable and ancillary variable to detect/predict some events that can affect a target entity. The details of the modeling process using the DM-PMS can be viewed in [1].

B. Bayesian Network

A BN is a probabilistic model represented by a directed acyclic graph which uses arrows ("directed arcs") to show how various factors – represented by nodes $X = \{X_1, ..., X_n\}$ – influence one another. Each node comes with its own probability table, known as a conditional probability table (CPT), reflecting the chances of various outcomes resulting from the different influences directly affecting it [14]. In

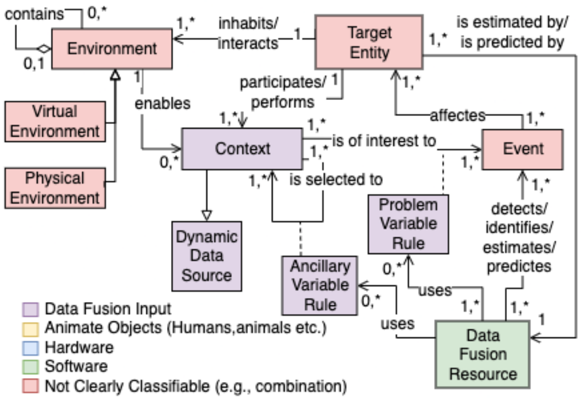


Fig. 1. The core concepts of domain model for Personalized monitoring system (DM-PMS) proposed by [1].

particular, we calculate the joint probability distribution of a given BN as follows

$$P(X) = \prod_{i}^{n} P(X_i | parents(X_i)). \tag{1}$$

Where $parents(X_i)$ denotes the parents of node X_i and $P(X_i)$ is called the joint distribution of X, which is the product of conditional probabilities of node X_i, given its parents. This conditional probability can be defined as a distribution if X_i takes continuous values, or as a probability table if it takes categorical values [19]. In this sense, a BN is considered a type of probabilistic domain model, as it represents the probabilistic information in a domain and the relationships between variables in that domain.

III. DEMDE

DEMDE consists of a probabilistic domain model and a modelling process. Its application in the design phase is done after modelling the PMS domain. This is illustrated in Figure 2.

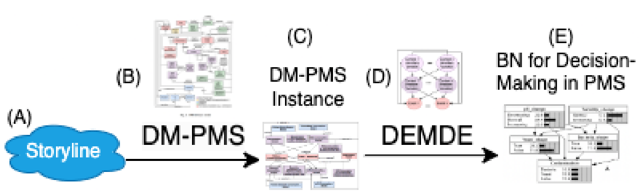


Fig. 2. DEMDE. First, a PMS domain (C) is modelled using a storyline description (A) and DM-PMS (B). Lastly, a BN (E) is extracted from modelled PMS domain (C) using the probabilistic domain model and modelling process of DEMDE (D).

A. *Probabilistic domain model - General BN for DeM in PMS*

Structuring a specific BN from a Conceptual Model (CM) is a good practice in Bayesian network modelling [13]. In this case, the CM can be a general BN to represent the fundamental relationships and dependencies in a broad PMS category, while a specific BN would include more detailed information about a particular PMS. We can highlight four reasons for building a specific BN from a general BN (conceptual model): i) Consistency: By using a general BN as a basis for a specific BN, the resulting models are likely to be more consistent with one another, which can enhance model quality and reliability; ii) Faster development: By starting with a general BN, much of the work has already been done, making the development of the specific BN more efficient. Since much of the groundwork has already been laid, the development of the specific BN can be accomplished more quickly than starting from scratch; iii) Better communication: A general BN can be a useful tool for communicating the structure and assumptions of a system to others, providing a common language and framework for understanding; and iv) Reduced complexity: A general BN can help to simplify complex systems, making them more manageable and easier to understand.

Our general BN is a probabilistic domain model that illustrates the concepts of Event and Context (used in DM-PMS [1], [18]) as a set of random variables and their conditional dependencies by the directed acyclic graph. Given the description of DM-PMS (Section II-A), we have that one or more contexts directly influence the detection/prediction of one or more events. In this case, we use the "context of" concept [1], [18]. Consequently, we can define Events as conditional dependent on the Contexts as Ancillary variable (Contextual Ancillary variables - Ca) and Contexts as Problem variable (Contextual Problem variables - Cp). Furthermore, contexts can be used as an ancillary variable to help/support problem variables, and in this case, we use the "context for" concept [1], [18]. Consequently, we can define Cp as conditional dependent of Ca. Then, based on these local conditional dependencies defined and that Ca are independents, we have the following joint probability for the general BN defined as

$$P(Ev, Cp, Ca) = \prod_{a}^{n_{Ev}} \prod_{b}^{n_{Cp}} \prod_{c}^{n_{Ca}} P(Ev_a | Cp_b, Ca_c) P(Cp_b | Ca_c) P(Ca_c) \tag{2}$$

where n_{Ev}, n_{Cp}, n_{Ca} represent the total number of Events, Cp and Ca identified in a DM-PMS instance. The visual representation of this joint probability distribution is presented in the Figure 2.

B. *Modelling process*

A modelling process is essential to build a specific BN because it ensures a structured and systematic approach to the modelling task [13]. In addition, it ensures that the model accurately represents the domain and captures important relationships between variables, leading to better communication among stakeholders and reducing the risk of misunderstandings and errors [15], [16]. Then, the modelling process to create a specific BN from the general BN and any instance of domain Model for PMS Based on context-aware data fusion proposed in [1] was based on good practice in BN modelling [13], [20] and is composed of the four steps:

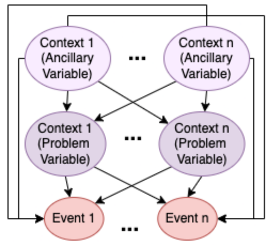


Fig. 3. General Bayesian network. The conceptual model used as basis to build the specific BN in PMS. The conceptual nodes in this general BN are Event, Contexts as Ancillary variable, and Contexts as Problem variable and the total number of them to be used in a specific BN depends of the number of classes in a DM-PMS instance.

Step 1 - Problem formulation. i) Identify the specific BN nodes based on the conceptual nodes (Events (Ev), Contextual Ancillary variables (Ca), and Contextual Problem variables (Cp)) of general BN in an instance of domain Model for PMS; and ii) define the decision problem question(s) to be addressed by the specific BN based on the defined storyline. We can define it using the following questions structures: a) Can the *event* affect significantly the *target entity* given the *context*(s)? b) What is the probability of the *event* given the *context*(s)? c) Should the system perform the *action*(s) based on the probability of *event* given the *context*(s)? These structures are based on the most common decision problem question addressed by BN [14], [21]. In these cases, BN is a central piece of the decision-making process, although it can or not be the whole process.

Step 2 - Model structure development. i) Defining the joint probability for the specific BN by instantiating the Equation 2 based on the specific BN nodes and decision problem question identified and defined previously; ii) Determine the states (for example, true or false) of each specific BN node; and iii) Determining the specific BN graphical structure (visual representation) based on joint probability defined in Step 2.i.

Step 3 - Model parameter estimation. i) Determining the conditional probability tables (CPTs); and ii) Assigning probabilities to the specific BN nodes based on available data, expert opinions, or other sources of information. The CPTs can be built by eliciting knowledge from domain experts (the knowledge-based approach), learning from data (the data-driven approach), or a combination of the two (information fusion) [14].

Step 4 - Model evaluation. Define the evaluation metric based on the problem formulation (specific application) performed in Step 1. This step is necessary to assess the quality and validity of the specific BN. It involves testing the specific BN's performance against data or expert knowledge to ensure that the model accurately reflects the system it represents [13], [20]. Some metrics can be found in [13], [20]. The most commonly used are: i) metrics of model sensitivity and influence, ii) metrics of model complexity, iii) Metrics of model prediction performance, for example, a) error rates and confusion tables, b) ROC curves and AUC, c) k-Fold cross-validation and d) Weighted confusion error rates. Furthermore, when data on the system is limited or unavailable, qualitative forms of model evaluation, such as peer review, are valuable [13]. In such cases, experts in the field can provide their expert opinions and knowledge to evaluate the model, including testing whether the model's behaviour is consistent with the current understanding of the system. This approach can help to identify potential flaws or inconsistencies in the model and improve its overall quality. However, it is essential to note that peer review should be considered an additional step to quantitative model evaluation rather than a replacement, especially when data is available [13].

These steps are often iterative and may involve going back and forth between different steps to refine the model and improve its performance.

IV. EVALUATION

In the literature, modelling approaches such as DEMDE are commonly evaluated through their application in case studies with different scenarios (that is, specific situations that the system is likely to face during operation) that allow illustrating all proposed relationships and concepts [1], [12], [22]–[26]. Our proposed approach is then illustrated with one case study, showing how the concepts in the probabilistic domain model can be applied to concrete application scenarios. The case is the cell-culture contamination prediction system.

A. Case - Cell culture contamination prediction for Bioreactor digital twin system

Here, we explore the use of DEMDE and DM-PMS to design the DeM for a PMS to be used in a bioreactor digital twin to alert about the possibility of contamination in cell cultures. Therefore, the general process used in the evaluation is the following. First, we created a DM-PMS instance applying the DM-PMS, and lastly, we designed the DeM applying the DEMDE approach with the DM-PMS instance, as described in Figure 2.

1) Application of DM-PMS: Here, we defined the Storyline and instantiated a domain model using the DM-PMS.

i) Defined Storyline - John works as a technical researcher (process operator) in a pharmaceutical company that produces monoclonal antibodies (mAb) in a bioreactor using Chinese hamster ovary (CHO) cell culture. The company uses a bioreactor digital twin system for predicting and alerting about contamination in cell culture at an early stage to avoid losses in time, money, and effort. The physical components of the system are a free-floating wireless sensor to monitor online pH and turbidity changes and in situ microscope to detect online bacteria and yeast. The virtual components are soft sensors that predict the probability of changes in pH and turbidity for the next hours based on time series data generated. Furthermore, the alert system works with two business rules: a) if there is an increase of pH and turbidity in the following hours, an alert must be triggered to the process operator because yeast can be detected in the microscope (yeast contamination); and b) if there is a decrease in pH and an increase in turbidity in the following hours, an alert must be triggered to the process operator because bacteria can be detected in the microscope (bacteria contamination).

ii) Cell culture contamination prediction from a User's perspective. At 1 pm, John started the production in a bioreactor. He received an alert message on his mobile phone 24h later about the high probability of contamination due to the high probability of increasing pH and the high probability of increasing turbidity in the following hours. Then, he confirmed the contamination by detecting yeast shapes in the microscope and started a procedure to rescue the contaminated cell culture. The modeling of this prediction can be seen in the diagram (A) of Figure 4 that is a DM-PMS instance. This modeling was performed following the process descried in [1].

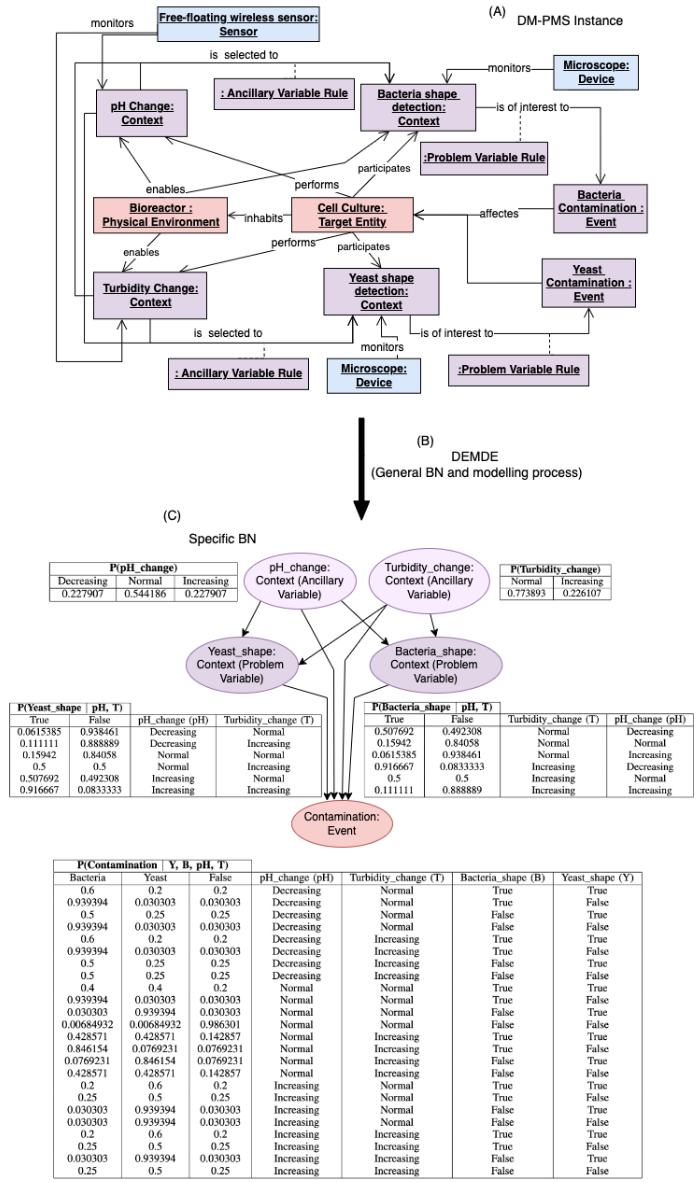
2) Application of DEMDE modeling process: Based on the results of the application of DM-PMS, we created the specific Bayesian network for DeM in Cell culture contamination prediction of bioreactor digital twin system using the DEMDE modeling process described in Section III-B. The application of DEMDE modeling process is represented in Figure 4, and each step performed is described in the following.

Step 1 - Problem formulation. i) The specific BN nodes identified are: pH_change (pH) and Turbidity_change (T) defined as contextual ancillary variable (Ca), Yeast_shape (Y) and Bacteria_shape (B) defined as contextual problem variable (Cp) and Contamination (C) defined as Event; see diagram (C) in Figure 4. ii) The decision problem question defined is the following: Should the digital twin system send a high contamination risk alert to the process operator, based on the high probability (for example, >60%) of yeast contamination given a high probability (for example, >60%) for increasing of pH and turbidity in the following hours?

Step 2 - Model structure development. i) The full joint probability distribution defined based on step 1 is the following

$$P(C, Y, B, pH, T) = \\ P(C|Y, B, pH, T)P(Y|pH, T)P(B|pH, T)P(pH)P(T). \tag{3}$$

ii) The defined states for each specific BN node can be viewed in diagram (C) of Figure 4. The pH_change node, $P(pH)$, has three states representing the probability of pH decreasing, increasing, and being normal in the following hours. The Turbidity_change node, $P(T)$, has two states representing the probability of turbidity being normal and increasing in the following hours. The Yeast_shape node, $P(Y|pH, T)$, has two states representing the probability of detecting (true) and not detecting yeast (false) in the cell culture using the microscope. The Bacteria_shape node, $P(B|pH, T)$, has two states representing the probability of detecting (true) and not detecting bacteria (false) in the cell culture using the microscope. Lastly, the Contamination node, $P(C|Y, B, pH, T)$, has three states representing the probability of bacteria and yeast contamination and not contamination (false) in the cell culture. iii) the visual representation full joint probability are presented in diagram (C) of Figure 4.



P(pH_change)		
Decreasing	Normal	Increasing
0.227907	0.544186	0.227907

P(Turbidity_change)	
Normal	Increasing
0.773893	0.226107

P(Yeast_shape | pH, T)			
True	False	pH_change (pH)	Turbidity_change (T)
0.0615385	0.938461	Decreasing	Normal
0.111111	0.888889	Decreasing	Increasing
0.15942	0.84058	Normal	Normal
0.5	0.5	Normal	Increasing
0.507692	0.492308	Increasing	Normal
0.916667	0.0833333	Increasing	Increasing

P(Bacteria_shape | pH, T)			
True	False	Turbidity_change (T)	pH_change (pH)
0.507692	0.492308	Normal	Decreasing
0.15942	0.84058	Normal	Normal
0.0615385	0.938461	Normal	Increasing
0.916667	0.0833333	Increasing	Decreasing
0.5	0.5	Increasing	Normal
0.111111	0.888889	Increasing	Increasing

P(Contamination | Y, B, pH, T)						
Bacteria	Yeast	False	pH_change (pH)	Turbidity_change (T)	Bacteria_shape (B)	Yeast_shape (Y)
0.6	0.2	0.2	Decreasing	Normal	True	True
0.939394	0.030303	0.030303	Decreasing	Normal	True	False
0.5	0.25	0.25	Decreasing	Normal	False	True
0.939394	0.030303	0.030303	Decreasing	Normal	False	False
0.6	0.2	0.2	Decreasing	Increasing	True	True
0.939394	0.030303	0.030303	Decreasing	Increasing	True	False
0.5	0.25	0.25	Decreasing	Increasing	False	True
0.5	0.25	0.25	Decreasing	Increasing	False	False
0.4	0.4	0.2	Normal	Normal	True	True
0.939394	0.030303	0.030303	Normal	Normal	True	False
0.030303	0.939394	0.030303	Normal	Normal	False	True
0.00684932	0.00684932	0.986301	Normal	Normal	False	False
0.428571	0.428571	0.142857	Normal	Increasing	True	True
0.846154	0.0769231	0.0769231	Normal	Increasing	True	False
0.0769231	0.846154	0.0769231	Normal	Increasing	False	True
0.428571	0.428571	0.142857	Normal	Increasing	False	False
0.2	0.6	0.2	Increasing	Normal	True	True
0.25	0.5	0.25	Increasing	Normal	True	False
0.030303	0.939394	0.030303	Increasing	Normal	False	True
0.030303	0.939394	0.030303	Increasing	Normal	False	False
0.2	0.6	0.2	Increasing	Increasing	True	True
0.25	0.5	0.25	Increasing	Increasing	True	False
0.030303	0.939394	0.030303	Increasing	Increasing	False	True
0.25	0.5	0.25	Increasing	Increasing	False	False

Fig. 4. Building the DeM to alert about the possibility of contamination in cell cultures of case study. In (A), we have an Object Diagram (DM-PMS instance) representing cell culture contamination prediction. This DM-PMS instance is used in DEMDE modeling process (B) to build the specific Bayesian decion Network (C).

Step 3 - Model parameter estimation. The CPTs were learned from the frequency information implicit in a synthetic dataset using Netica software [27]. This synthetic dataset was manually developed based on research papers results [28]–[30] about cell culture contamination by bacteria and yeast, and consists of a total of 427 occurrences, see the Table I. Among these occurrences, 142 are labeled as bacteria contamination, 142 are labeled as yeast contamination, and 143 are labeled as no (False) contamination. The dataset is designed to simulate real-world scenarios and provide a diverse set of samples for training the specific BN presented in diagram (C) of the Figure 4.

TABLE I
OCCURRENCES OF CELL CULTURE CONTAMINATION IN SYNTHETIC DATASET BASED ON THE STATES OF THE NODES PH_CHANGE (PH), TURBIDITY_CHANGE (T), YEAST_SHAPE (Y) AND BACTERIA_SHAPE (B) AND CONTAMINATION (C).

pH	T	B	Y	C	#occurrence
Normal	Normal	False	False	False	143
Decreasing	Normal	True	True	Bacteria	2
Decreasing	Normal	True	False	Bacteria	30
Decreasing	Normal	False	False	Bacteria	30
Decreasing	Normal	False	True	Bacteria	1
Decreasing	Increasing	True	True	Bacteria	2
Decreasing	Increasing	True	False	Bacteria	30
Decreasing	Increasing	False	False	Bacteria	1
Decreasing	Increasing	False	True	Bacteria	1
Normal	Normal	True	True	Bacteria	1
Normal	Normal	True	False	Bacteria	30
Normal	Increasing	True	True	Bacteria	2
Normal	Increasing	False	False	Bacteria	2
Normal	Increasing	True	False	Bacteria	10
Normal	Normal	True	True	Yeast	1
Normal	Normal	False	True	Yeast	30
Normal	Increasing	True	True	Yeast	2
Normal	Increasing	False	False	Yeast	2
Normal	Increasing	False	True	Yeast	10
Increasing	Normal	True	True	Yeast	2
Increasing	Normal	True	False	Yeast	1
Increasing	Normal	False	False	Yeast	30
Increasing	Normal	False	True	Yeast	30
Increasing	Increasing	True	True	Yeast	2
Increasing	Increasing	True	False	Yeast	1
Increasing	Increasing	False	False	Yeast	1
Increasing	Increasing	False	True	Yeast	30

Step 4 - Model evaluation. The inclusion of insignificant variables can increase the complexity of the network and reduce the sensitivity of the model outputs to essential variables. Therefore, we performed a sensitivity analysis using the Netica software [27] to determine how much the beliefs of the contamination node (target node representing the Event concept) could be influenced by a single finding at other nodes of the BN. A single finding refers to an observation or evidence provided at a single node of the Bayesian network, which may affect the beliefs or probabilities associated with other nodes in the network [31]. The results can be viewed in Table II that is showing the following sensitivity metrics. In the first column, we have mutual information. It is a measure of the reduction in uncertainty of a target node that occurs when a finding is made on another node in a Bayesian network [32]. It is calculated based on the entropy reduction in the target node due to the finding on the other node. Mutual information reflects the degree of dependence between two nodes in a network and can be used to quantify the sensitivity of the target node to findings on other nodes. A higher mutual information value indicates a stronger dependence between the nodes, and a greater potential for the finding on the other node to affect the target node. In the second column, we have percentage. It represents the percentage of the mutual information contributed by a particular node (e.g. pH_change) to the mutual information between that node and the Contamination node. This value is calculated by dividing the mutual information between the two nodes by the mutual information of the Contamination node, and multiplying the result by 100. Last column represents the Variance of Beliefs that measures the variability in the beliefs associated with a node.

The node representing the change in pH (mutual info value of 0.56772) has a more significant impact on the contamination node than the node representing the change in turbidity (mutual info value of 0.06675). This is reasonable because a high probability of increase in turbidity can indicate the presence of contamination. However, it does not provide information about the type of contamination; see diagrams A and B in Figure 5. In the diagram A, we have the current beliefs (i.e. posterior probabilities) for the nodes based on synthetic dataset, and the diagram B shows the impact of the finding of 70% for state increasing of turbidity_change node on Contamination node. Comparing these diagrams, we can see that the high probability of increase in turbidity increases the probability of bacteria and yeast contamination (from 33.3% to 40.8$) in the same proportion in the contamination node. However, changes in pH can directly indicate the presence of certain types of contamination; see diagrams C and D in Figure 5. The diagram C shows the finding of 71.2% for decreasing in pH_change node indicate a probability of 69.2% for bacteria contamination. On the other hand, the diagram D shows the finding of 71.2% for increasing of pH_change node indicate a probability of 69,2% for yeast contamination. Furthermore, changes in pH and turbidity increase the probability of detecting certain types of contamination, see diagram E and F in Figure 5. Therefore, although turbidity_change node presents the lowest impact on the contamination node, it is not an unnecessary variable since it became decisive when used with pH_change variable. All of the situations represented in diagrams B, C, D, E, and F in Figure 5 can be used to alert the process operator to check the cell culture about the possibility of contamination. In addition, the situations represented in diagrams D and F in Figure 5 can be used to answer positively the decision problem question defined previously.

TABLE II
SENSITIVITY OF CONTAMINATION NODE TO A FINDING AT ANOTHER NODE

Node	**Mutual Info**	**Percent**	**Variance of Beliefs**
Contamination	1.58496	100	0.4444431
pH_change	0.56772	35.8	0.1598104
Bacteria_shape	0.27849	17.6	0.0603023
Yeast_shape	0.27849	17.6	0.0603022
Turbidity_change	0.06675	4.21	0.0064754

V. RELATED WORKS

There are some approaches in the literature to build specific BN from conceptual models. The authors in [33] develop a Bayesian network for modelling learners using the use case diagram of the Unified Modelling Language, discussing how to go from a dynamic representation of the learner model using UML to a probabilistic representation with Bayesian networks. The authors in [26] propose a method for constructing

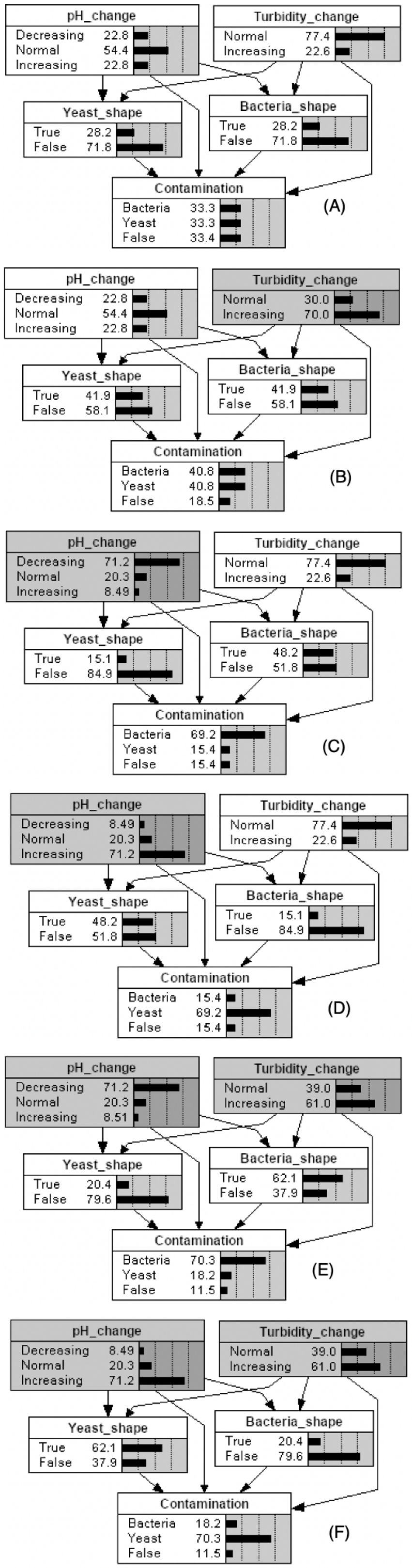


Fig. 5. Bayesian network for DeM in Cell culture contamination prediction of bioreactor digital twin system (case 2). In the diagram (A), we have the current beliefs (i.e. posterior probabilities) for the nodes based on synthetic dataset. In diagram (B) we have that assign a probability 70% to 'increasing' state of Turbidity_change node increase the probabilities of contamination. The diagram (C) shows that assign a probability 71.2% to 'decreasing' state of pH_change node increase the probabilities of Bacteria contamination and the diagram (D) shows opposite. The diagram (E) and (F) show the influence of findings pH_change and Turbidity_change nodes on the beliefs of contamination node.

Bayesian causal maps, which combine the strengths of causal maps and Bayesian networks, to represent domain knowledge and enable decision-making, and illustrates its application in the context of an information technology application outsourcing decision. In [34] was proposed an autonomic decision-making system based on Bayesian networks and ontologies, which was tested in an autonomic communication system and demonstrated improved performance, with the potential for implementation in other contexts. Finally, [25] proposed using probabilistic relational models to infer security risk from architectural metamodel instantiations without requiring the assessment of complex security attributes. All these literature approaches may be explored to try to design DeM in PMS. However, they do not have the important concepts and relationships of the PMS domain, and the modeling process would be basically ad hoc and time-intensive.

VI. Discussion

The case modelled in the evaluation section demonstrates the potential of DEMDE as a systematic approach illustrating the application of the general BN and the modelling process. Furthermore, the modeled case showed that due to the similarities between DT and PMS, the DM-PMS was able to guide the development of a domain model for DT. Besides illustrating the DEMDE concepts, this case is a good opportunity to show the potential of DEMDE and DM-PMS to model a DT system domain. Since it was reported in the literature the need for a DM for DT system development [6]. However, so far, no DM has been proposed. However, one limitation that stakeholders must keep in mind is that the specific BNs created by DEMDE could not be enough in some domains. Then DEMDE can be used to create a reasonable starting point that may need modifications to incorporate new concepts and relationships to achieve a purpose of specific domain.

VII. Conclusion

Generating a Bayesian network from a domain model is a practical approach to designing the decision-making process in a system. The Bayesian network is a probabilistic model that captures the dependencies between the variables in the system. In contrast, the domain model provides a clear and comprehensive understanding of the system's structure and behaviour. By combining these two models, designers can gain a more accurate and holistic view of the system, which can help them make informed decisions. This work proposed a systematic procedure, called DEMDE, that enables the design of specific Bayesian network from an instantiated domain model for PMS using a probabilistic domain model (general BN). It is the first general modelling process for decision-making design in PMS during the design phase of the Systems Development Life Cycle (early stage of development). In future work, we intend to build a named-entity recognition framework (natural language processing technique) to automatically generating specific BN directly from storylines of PMS.

References

[1] C. Iglesias, C. Miceli, and D. Silva, "A domain model for personalized monitoring system based on context-aware data fusion," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, IEEE, 2019.

[2] F. Mostafa, L. Tao, and W. Yu, "An effective architecture of digital twin system to support human decision making and ai-driven autonomy," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 19, p. e6111, 2021.

[3] Y. R. Shrestha, S. M. Ben-Menahem, and G. Von Krogh, "Organizational decision-making structures in the age of artificial intelligence," *California Management Review*, vol. 61, no. 4, pp. 66–83, 2019.

[4] V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of digital twins for smart manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 366–384, 2019.

[5] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[6] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE access*, vol. 8, pp. 108952–108971, 2020.

[7] Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda," *International journal of information management*, vol. 48, pp. 63–71, 2019.

[8] A. Kossiakoff, S. M. Biemer, S. J. Seymour, and D. A. Flanigan, *Systems engineering principles and practice*. John Wiley & Sons, 2020.

[9] G. Kotonya and I. Sommerville, *Requirements engineering: processes and techniques*. Wiley Publishing, 1998.

[10] J. W. Herrmann, *Engineering decision making and risk management*. John Wiley & Sons, 2015.

[11] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. Van Kranenburg, S. Lange, and S. Meissner, *Enabling things to talk*. Springer Nature, 2013.

[12] S. Haller, A. Serbanati, M. Bauer, and F. Carrez, "A domain model for the internet of things," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pp. 411–417, IEEE, 2013.

[13] S. H. Chen and C. A. Pollino, "Good practice in bayesian network modelling," *Environmental Modelling & Software*, vol. 37, pp. 134–145, 2012.

[14] A. C. Constantinou and N. Fenton, "Things to know about bayesian networks: Decisions under uncertainty, part 2," *Significance*, vol. 15, no. 2, pp. 19–23, 2018.

[15] E. Evans and E. J. Evans, *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley Professional, 2004.

[16] E. Evans, *Domain-Driven Design Reference: Definitions and Pattern Summaries*. Dog Ear Publishing, 2014.

[17] A. M. Hanea, A. Christophersen, and S. Alday, "Bayesian networks for risk analysis and decision support," 2022.

[18] L. Snidaro, J. García, J. Llinas, and E. Blasch, *Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge*. Advances in Computer Vision and Pattern Recognition, Springer International Publishing, 2016.

[19] A. T. Misirli and A. B. Bener, "Bayesian networks for evidence-based decision-making in software engineering," *IEEE Transactions on Software Engineering*, vol. 40, no. 6, pp. 533–554, 2014.

[20] B. G. Marcot, "Metrics for evaluating performance and uncertainty of bayesian network models," *Ecological modelling*, vol. 230, pp. 50–62, 2012.

[21] R. M. A. Valdés, V. F. G. Comendador, A. R. Sanz, E. S. Ayra, J. A. P. Castán, and L. P. Sanz, "Bayesian networks for decision-making and causal analysis under uncertainty in aviation," *Bayesian Networks: Advances and Novel Applications*, p. 21, 2019.

[22] B. Alessandro, B. Martin, F. Martin, K. Thorsten, L. Sebastian, M. Stefan, *et al.*, "Enabling things to talk: Designing iot solutions with the iot architectural reference model," 2013.

[23] A. Jansen and J. Bosch, "Software architecture as a set of architectural design decisions," in *5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)*, pp. 109–120, IEEE, 2005.

[24] N. Rozanski and E. Woods, *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley, 2011.

[25] T. Sommestad, M. Ekstedt, and P. Johnson, "A probabilistic relational model for security risk analysis," *Computers & security*, vol. 29, no. 6, pp. 659–679, 2010.

[26] S. Nadkarni and P. P. Shenoy, "A causal mapping approach to constructing bayesian networks," *Decision support systems*, vol. 38, no. 2, pp. 259–281, 2004.

[27] Z. Ni, L. D. Phillips, and G. B. Hanna, "Exploring bayesian belief networks using netica®," *Evidence Synthesis in Healthcare: A Practical Handbook for Clinicians*, pp. 293–318, 2011.

[28] J. A. Ryan *et al.*, *Understanding and managing cell culture contamination*. Corning Incorporated, 1994.

[29] R. Gustavsson, C.-F. Mandenius, S. Löfgren, T. Scheper, and P. Lindner, "In situ microscopy as online tool for detecting microbial contaminations in cell culture," *Journal of biotechnology*, vol. 296, pp. 53–60, 2019.

[30] R. Geraghty, A. Capes-Davis, J. Davis, J. Downward, R. Freshney, I. Knezevic, R. Lovell-Badge, J. Masters, J. Meredith, G. Stacey, *et al.*, "Guidelines for the use of cell lines in biomedical research," *British journal of cancer*, vol. 111, no. 6, pp. 1021–1046, 2014.

[31] M. Romanycia, "Netica-j reference manual," in *Norsys Software Corp*, 2019.

[32] U. B. Kjaerulff and A. L. Madsen, "Bayesian networks and influence diagrams," *Springer Science+ Business Media*, vol. 200, p. 114, 2008.

[33] A. T. Mouenis, A. Souhaib, and K. Mohamed, "Development of bayesian networks from unified modeling language for learner modelling," *International Journal of Ad-vanced Computer Science and Applications*, vol. 6, no. 2, 2015.

[34] J. Aguilar, J. Torres, and K. Aguilar, "Autonomie decision making based on bayesian networks and ontologies," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 3825–3832, IEEE, 2016.