



**HAL**  
open science

# Object oriented Bayesian Network for complex system risk assessment

Quan Liu, François Pérès, Ayley Tchangani

► **To cite this version:**

Quan Liu, François Pérès, Ayley Tchangani. Object oriented Bayesian Network for complex system risk assessment. IFAC-PapersOnLine, 2016, 49 (28), pp.31-36. 10.1016/j.ifacol.2016.11.006 . hal-01527240

**HAL Id: hal-01527240**

**<https://hal.science/hal-01527240>**

Submitted on 24 May 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>  
Eprints ID: 17672

To link to this article: DOI:10.1016/j.ifacol.2016.11.006

<http://dx.doi.org/10.1016/j.ifacol.2016.11.006>

**To cite this version:**

Liu, Quan and Pérès, François and Tchangani, Ayley *Object oriented Bayesian Network for complex system risk assessment*. (2016) IFAC-PapersOnLine, vol. 49 (n° 28). pp. 31-36. ISSN 2405-8963

Any correspondence concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

# Object Oriented Bayesian Network for complex system risk assessment

Q.Liu \* F.Pérès \*\* A.Tchangani \*\*\*

\* *Université Fédérale Toulouse Midi-Pyrénées - Laboratoire Génie de Production, INP-ENIT, 65016 France (e-mail: quan.liu@enit.fr).*

\*\* *Université Fédérale Toulouse Midi-Pyrénées - Laboratoire Génie de Production, INP-ENIT, 65016 France (e-mail: francois.peres@enit.fr).*

\*\*\* *Université Fédérale Toulouse Midi-Pyrénées - Laboratoire Génie de Production, INP-ENIT, 65016 France (e-mail: ayeley.tchangani@iut-tabes.fr).*

**Abstract:** In this communication, we present a new way of modelling risk management process in a large scale, complex and dynamic system using an extended object oriented Bayesian network (EOOBN) as the underlying mathematical tool. The communication begins by presenting the proposed extension that permits to take into account the necessity of varying parameters in object oriented Bayesian network (OOBN) and dynamics for practical purposes. The second part is devoted to adaption of existing information propagation algorithms in classical Bayesian network (BN) and dynamic Bayesian network (DBN) to this extended model. A small practical case study is used along the communication to illustrate each modelling step.

*Keywords:* complex system, modelling, OOBN, DBN, risk management

## 1. INTRODUCTION

In large scale industrial processes, with complex interactions among the components and dynamic evolution, an abnormality or a component failure may result in a cascade phenomenon that can lead to unacceptable risks or the collapse of such networked systems. Therefore one must have at its disposal efficient and sound tools, methods or framework to assist managers of such complex systems in their analysis, the prediction of their behaviour, or in their control or management of the effect of possible undesirable events that may affect them. Supplying such a decision support system becomes a challenge for researchers. Modelling risks requires indeed considering the nature of the relationships (influence, causality, etc.), the related uncertainties (about the existence of a relationship, of its intensity or even about the delimitation of the system under consideration), the dynamics (evolution of the model structure and/or parameters over the time) see Bouzarour-Amokrane et al. (2012). The relationships, uncertainty and dynamics characterize the whole system. Considering these characteristics, in modelling process, leads to a complex and large scale system. Within this framework, using classical physical laws to describe it is often unsuitable and the need for new more tractable approaches becomes necessary. In this communication, we attempt to model an industrial process through an extended object oriented Bayesian networks (EOOBN) based on components sharing a same structure. The main idea is first to consider inside these elementary components, uncertainties of interactions among different variables through Bayesian network properties. Then the whole system will be described by associating these components through object oriented

mechanisms. This modelling approach will help monitoring and measuring the evolution of the system for a better understanding and controlling of its behaviour. The main purpose of this communication is thus to model a system in order to facilitate risk management and decision making. The communication is structured as follows. Section 2 provides a brief overview of complex systems and the Bayesian network techniques. Section 3 presents a new extended object oriented Bayesian network and shows in particular how the dynamic evolution is introduced in an object. Adapted inference algorithms associated with the developed EOOBN model are introduced in section 4. Finally, a conclusion, some perspectives and future works are presented in the last section.

## 2. STATE OF THE ART

### 2.1 Complex system

A complex system is a large system with four main characteristics: it is composed with a large number of components, these components are interconnected, effects of these interconnections are uncertain, and some of their characteristics are likely to evolve over time (dynamicity). The goal of this communication is to develop an approach for modelling these complex systems in order to evaluate the risk when one or some of their composed components are destabilized by an external event or an internal cause. Such kind of model can be used to assist the assessment process of indicators or the decision making process, in different domains such as economy, medicine, production and many other fields. In Amaral and Ottino (2004), the author points out issues related to classical

modelling methods based on assumptions which eventually can bias the results by ignoring the aspects of interdependency, dynamic and size of the system. Finding another approach to simulate these complex systems when avoiding a great number of hypothesis is worth of research. Bayesian Networks are very efficient for modelling uncertainties. Meanwhile Dynamic Bayesian Network may be used when there is a temporal dimension in the system behaviour see Murphy (2002). In the case of system with a huge number of components, a possibility to reduce this complexity is to use the so called Object Oriented Bayesian Networks (OOBN) in order to exploit possibilities offered by this modelling technique. The idea of modelling repeatable systems by object oriented techniques has been already considered in a certain number of studies such as Jaeger (2000), Weber and Jouffe (2006) to mention just a few. In the following paragraph, we introduce basics of static and dynamic Bayesian network characteristics.

## 2.2 Bayesian network and Dynamic Bayesian Network

A Bayesian Network is a directed acyclic graph (DAG) that represents a certain relationships (in general causal relationships) between variables in a certain knowledge domain; each node represents a random variable associated with a conditional probability table (CPT) characterizing its parameters. Bayes theorem is the central theory in the mechanism of inference in Bayesian Networks. It permits to propagate some local observations through the graph in order to update a priori knowledge about the state of other nodes, see Nielsen and Jensen (2009), Pearl (1988). Figure 1 shows an example of a Bayesian Network that represents the performance evaluation mechanism of a production machine. In this example, the basic elements (nodes in the corresponding BN) that may influence the productivity (*pieces/hour*) of a machine are: *machine performance*, *workers' motivation*, *upstream production flow* and *products' quality*. All these elements are called variables in the BN. Meanwhile *Products' Quality* and *Upstream Product* influence the *Product*; nodes representing *Products' Quality* and *Upstream Product* are therefore known as parents of node representing *Product*. There are two types of probability tables in a BN see Godichaud et al. (2012b): prior probabilities tables for root variables (variables without parents) like *Machine Performances* and conditional probabilities tables for a variables with parents like *Products' Quality* see Godichaud et al. (2012a), Godichaud et al. (2012b). Indeed, a BN model is not only a static representation of knowledge but also a tool for the evidence inference which updates the probabilities in the network and enables the refinement of the results according to the observed situation see Ben Hassen et al. (2013).

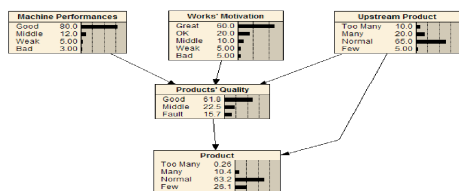


Fig. 1. Representation of machine performance

In order to take into account of possible dynamic behaviour of systems, dynamic Bayesian networks (DBN),

are introduced as possible extension of a BN. The DBN is a series of time-slice BN; a typical DBN is a Hidden Markov Models (HMM), see Rabiner and Juang (1986). Learning techniques in a DBN use the same principles as in the classical BN see Murphy (2002). For instance, in the former example of Fig.1, it is reasonable to consider that machine performances will decline over time, so that the node *Machine Performances* becomes a dynamic node; Fig.2 shows a DBN model of problem considered in Fig.1 with 2 time-slices, where the dynamic node is *MachinePerformances<sub>t<sub>i</sub></sub>* ( $MP_{t_i}$ ). The communication between time-slices uses the transition model determined by a transition matrix A as:

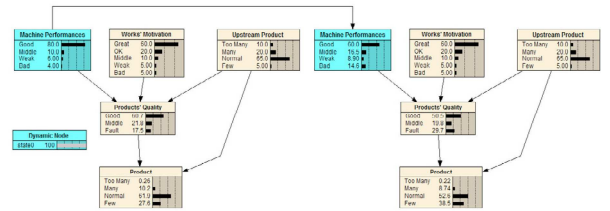


Fig. 2. Dynamic Bayesian network representing a machine performance

- the transition model:  $A = P(MP_{t_i} | MP_{t_{i-1}})$
- the initial state:  $\pi = P(MP_{t_0})$

## 2.3 Object Oriented Bayesian Network

Using BN techniques for modelling risk assessment processes becomes increasingly complex when the size of the system increases. For a large scale system with many interacting elements, constructing a BN to represent its functioning behaviour may be very challenging. Meanwhile, when the size of network grows, the model visibility reduces and the update of parameter becomes burdensome. For this reason, an object oriented techniques might be an alternative to reduce the complexity by highlighting a generic pattern representative of the various dimensions of the problem. An object oriented Bayesian network (OOBN), is a direct application of the object paradigm see Bangsø and Wullemmin (2000), Koller and Pfeffer (1997). The basic element is the class, fragment of a Bayesian network who has three sets of nodes: input, output and internal nodes. The input and output nodes are the interface of a class which can be seen from the outside. The OOBN takes advantage of classic BN but introduces the concept of instance nodes. An instance node is an abstraction of a part of a network which can be used as an elementary component to represent the whole structure. The notion of encapsulation allows the transmission of all properties of the network fragment. An object oriented network can be viewed as a hierarchical description/model of a problem. This makes the modelling easier since the OOBN-fragments at different levels of abstraction are more readable. An OOBN model can be built by asking the experts' opinions or using the learning techniques. In Langseth and Nielsen (2003) and Wullemmin and Torti (2012), the authors give some insights into OOBN structure learning. The construction of such a model can be facilitated by an ontology representation see Liu et al. (2015). Once the structure of the system is defined, the Conditional Probability Tables

(CPTs, also called parameters) have to be parametrized. In Langseth and Bangsø (2001) the author extends the parameter learning algorithm to the objects who have the same structure based on OO assumption. The parameters being identical, this reduces the number of parameters to be specified or learnt. Modelling a complex system by an OOBN allows not only reducing the design work, but also updating calculations. However, most of existing works dealing with this topic consider that parameters do not change from an object to another which is not a realistic assumption in a real world problem modelling context. As our main goal is to use OOBN for the representation of a large, repeatable and inherited system, this shortcoming must be remedied. In the next section, we will describe a proposed extension of OOBN paradigm that we refer to as Extended Object Oriented Bayesian Network or EOOBN for short. An adapted inference mechanism for this EOOBN will be presented in section 4.

### 3. EXTENDED OBJECT ORIENTED BAYESIAN NETWORK

In this section, we present an extended object oriented Bayesian network (EOOBN) which introduces much more flexibility, such as the possibility of having different parameters for different objects and taking into account dynamic behaviour of the system, etc. The main steps for constructing an EOOBN will be considered in the subsequent paragraphs showing how the extension is introduced.

#### 3.1 Extended OOBN

To overcome the limitations of classical OOBN associated with the structure building Bangsø and Wuillemin (2000) and Koller and Pfeffer (1997), and the difficulty to take into account dynamic interactions Koller and Pfeffer (1997), we propose here an extended OOBN to ease parameter variation and dynamic consideration process. The original contribution in this communication is to develop an EOOBN which simulates the large scale system with different parameters for objects having the same structure. Moreover our approach is easy to adapt by collaboration mechanism. Here below are the main components of an EOOBN:

##### Class and Object

In this section, we will consider two levels in EOOBN: definition of a class is made at structure level (that is the nodes and their connexions in the object) and the object itself will be instantiated both through the input values and with respect to its parameters at the object level which are likely to evolve with context or time.

**Class:** A class ( $C$ ) is the structure part ( $S$ ) in a BN independently of the CPT parameters values. It has three kind of nodes namely: input nodes, output nodes and internal nodes. Only the input and output nodes are visible from outside the class.

**Object:** An object ( $O\{S, P\}$ ) in the OOBN is an instantiation of the corresponding class. There are two parts in an object, the structure ( $S$ ) which inherits from the class and the parameters ( $P$ ) which will be defined by experts or learning processes.

We refer to input and output nodes as *communication channel* for the class/object entity because they are in charge of exchanging information for the class/object. Here are some conditions that must be satisfied:

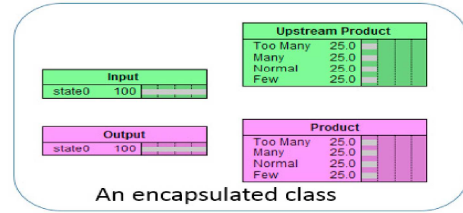


Fig. 3. Class associated with a machine

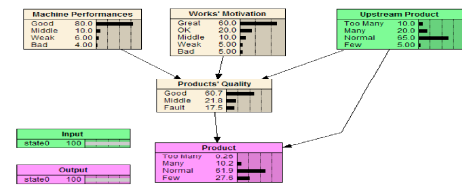


Fig. 4. Object characterizing a machine

- Input nodes cannot have parents inside the class
- Input node is a reference node which is the projection of an output or a normal upstream node
- Internal nodes cannot have neither parents nor children outside the class
- Output nodes cannot have child inside the class

In Figure 1 we give BN model of performance evaluation problem of a single machine. If we consider a real production workshop there may exist many similar machines so that a single machine may be considered as an object; Figure 3 shows a class level of such a machine and Figure 4 depicts the object level model.

An industrial processes (a production line for instance) can be represented through connected items (representing machines for instance). The input node is the *Upstream Product* which comes from the last machine, the output node is the *Product* which will be sent to the next machine. Only the output and input nodes can be seen from the outside (see Fig.3). The internal nodes such as *Productions Quality*, *Works' Motivation* are encapsulated in the class. The class characterizes only the structure of the network. A class can be used only after its instantiation; in this case a class is converted into an object (see Fig.4). The EOOBN not only inherits all the advantages of the classical OOBN such as hierarchy or encapsulation but also has much more flexibility in the quantitative part. The possibility of having different parameters values from an object to another variation of the parameters is achieved by the instantiation of a class. The necessity to follow the behaviour of real world systems appeal for introducing dynamicity in the model. This can be done by extended classical DBN in object oriented frame as shown in the following paragraph.

#### 3.2 Extended DBN by object

Although in Bangsø and Wuillemin (2000) a DBN simulation approach is given based on a self-reference node in

an object, a confusion might appear when trying to add the dynamic part within a large OOBN. To overcome this issue, we introduce the virtual nodes in the EOOBN to simulate the dynamic part. And we obtain an dynamic EOOBN which refers to as a DOOBN.

**Virtual node:** The virtual node is a communication channel for the class/object. It usually stands for the temporal node.

- The virtual node is either an input node or an output node in the class/object
- It is added for dynamic node in the class/object as a communication channel with other time-slice
- The transition model represents the parameters between the virtual input and the dynamic node. Conditional probabilities between a dynamic node and its virtual output are equalled to 1

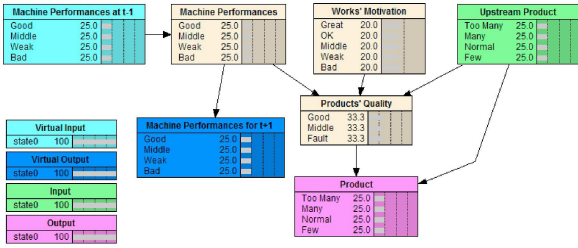


Fig. 5. A dynamic class for a machine with virtual nodes

An example of a dynamic class is given on Fig.5. The internal node *Machine Performances* (*MP* for short) is a dynamic variable, the virtual input *MP* at  $t_i$  receives the temporal information from the time-slice  $t_{i-1}$  and the virtual output *MP* for  $t_{i+1}$  is a projection of  $MP_{t_i}$ , which transfers the current information to the next time-slice  $t_{i+1}$ . The parameters for the dynamic relationships are:

- $A_{t_i, t_{i-1}} = P(MP_{t_i} | MP_{t_{i-1}})$
- $A_{t_{i+1}, t_i} = P(MP_{t_{i+1}} | MP_{t_i}) = 1$

Parameter  $A_{t_i, t_{i-1}}$  is the conditional probability table between time-slice  $t_i$  and  $t_{i-1}$  and parameter  $A_{t_{i+1}, t_i} = 1$  just transfers the information from  $t_i$  to  $t_{i+1}$ . Adding the virtual node in the class/object allows the encapsulation of an object. The communication between the time-slices through the virtual node keeps the independence of each time-slice.

### 3.3 Construction method

The construction of a dynamic EOOBN can be done by carrying out the following steps:

- (1) Formalize the structure  $S$  of a system (by splitting the system into different classes  $C$ )
- (2) Design the structure of each class ( $C$ ) with respect to  $S$  and without considering the dynamic part
- (3) Identify the dynamic node in the class  $N_{t_i}$  and add the virtual input and output node around the dynamic node
- (4) Instantiate the class by introducing the parameters corresponding to the object
- (5) Connect the objects through their communication channels

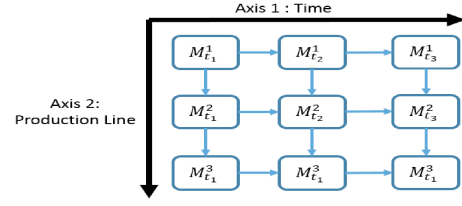


Fig. 6. Example of a dynamic EOOBN for the representation a production line

Figure 6 shows a three time-slices model of a production line formed by three machines. In Fig.5 the  $M^2_{t_2}$  represents the second machine at time  $t_2$ . Machine exchange production information. For instance the input node (*Upstream Product*) of the second machine  $M^2$  receives the output information (*Product*) from the first machine  $M^1$ . Dynamic evolution of Machine Performance is transferred by the virtual nodes. We refer to the second and third steps as *class level*, the fourth step as *object level* and the first and last steps as *global network level*. The EOOBN that we propose in this communication is always organized at two levels: global level and object level. In order for existing inference algorithms to work for the proposed EOOBN some adjustments are required that will be presented in the next section.

## 4. PROPAGATION IN EOOBN

Here we propose the algorithms for solving the inference problem in an EOOBN. In the real world when an observation is made, it can be used to either predict a future behaviour and/or diagnose the causes resulting in the current system state. The capacity of inference is an advantage of BN. As we have mentioned, the information propagation of a classic OOBN is done by either developing all the objects Bangsø and Wullemin (2000) to obtain a classical BN or by using the MSBN Koller and Pfeffer (1997); Xiang and Lesser (2000), losing by the way the sense of object oriented modelling. Traditional inference techniques are not suitable for the EOOBN processing due to the number of parameters and variables. Because of the size of EOOBN the inference work should also take into account the calculated size and time. For this reason, we propose here an adapted algorithm for EOOBN calculation using the advantages associated with the virtual nodes.

### 4.1 Local calculation

At the object level, as input and output nodes separate the objects, inference can be performed locally with no need of extra information provided by its neighbours.

The classical BN algorithms can be applied for the object calculation given that all the parameters are known. Indeed internal and output nodes correspond to previous knowledge and input nodes receive the probability from their neighbourhood. Local calculation can result of four situations:

- (1) There is no observation in the global network
- (2) There is observation(s) in the object
- (3) There is an update information(s) at the input of an object

- (4) There is only one type update information(s) at the output of an object

“No observation in the network” means that the network spreads by itself with time evolution. If there are several types (input and output) of information update, the semi-local calculation is to be used. Since at this time the soundness of a junction tree is broken, an object needs some more information from its neighbourhoods. This information is the conditional probabilities between the input and output nodes which are always the constants and can be computed easily. This extra information keeps the soundness of the object. But the semi-local calculation is only needed when inference is performed for a certain object of the global network. The feature of the local calculation is saving the computing time and size. Because computation can be performed in smaller parts of the global network, there is no need to build a big, complex junction tree. Moreover some objects inherit their structures from a same class, and they have the same junction tree. Due to local calculation possibilities, objects may do the computation independently, allowing possible parallel computation.

#### 4.2 Information flow

The computation in an EOOBN has two levels, the object level is the local calculation, and the global level is the information flow.

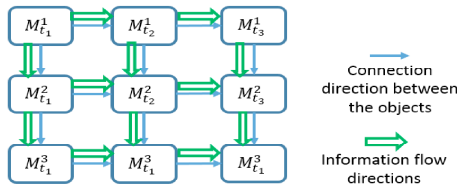


Fig. 7. Information flows without observation

*No observation* When there is no observation, the network spreads by itself. The information follows the same direction as the communication channels (see Fig.7), the machines receive the input informations such as *Upstream Product* and *MP* at  $t_{i-1}$  and process the output like *Product* and *MP* for  $t_{i+1}$  to the downstream objects.

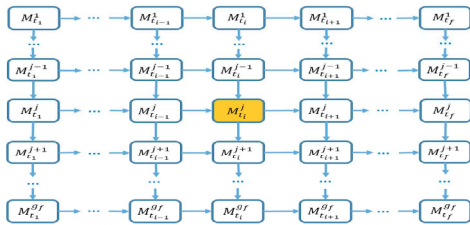


Fig. 8. Information flows introducing and observation

*Observation* If there is an observation in object, (for instance  $M_{t_i}^j$  Fig.8), the propagation of the information between objects is described in Fig.9 and Fig.10. Because the object has the local computation ability, one just needs to control the information flows. The algorithm for the inference is presented in Fig.11.

In this algorithm, the computation needs to be done first at  $M_{t_i}^m$  with  $1 \leq m < j$  and at  $M_{t_i}^j$  with  $t_1 \leq t < t_i$  which

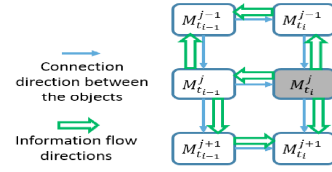


Fig. 9. Information flow between time-slices  $t$  and  $t - 1$

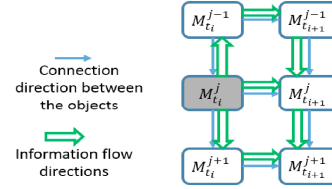


Fig. 10. Information flow between time-slices  $t$  and  $t + 1$

correspond to the two red dash parts. The results of this computation will be stored in a data base for the future calculation and they can be obtained at the same time. This simultaneity reduces the calculation time. After this step, we can also do the inference at the same time in the two blue parts. Moreover, if the querying object is known, the calculation is oriented directly to this object after the first step. Unlike the traditional method, which develops all the objects, first, builds a junction tree and proceeds to the propagation, the algorithm only needs here a few objects to answer the request by local calculation.

## 5. CONCLUSION

A new modelling technique using an extended Bayesian approach referred to as EOOBN and based on the concept of Oriented Object paradigm has been proposed to tackle modelling challenges raised by large scale and complex real world systems. The principles of defining the structure in the class and distributing the parameters for the instantiation have been shown. EOOBN models appear to be much powerful than the classical OOBN while preserving all its benefits, such as encapsulation, hierarchy and top-down design. When a repetitive structure allows the use of such Object Oriented representation, EOOBN is undoubtedly suitable for modelling complex situations encountered in risk management framework. To take into account the possibility of varying parameters and the system dynamics, EOOBN have been adapted and changed to obtain dynamic OOBN referred to as DOOBN where each time-slice is associated with an object and parameters are allowed to change from slices through the introduction of virtual nodes. An illustrative application in an industrial process has been used along the communication to show the interest of such modelling technique. Furthermore, inference algorithms demonstrate the gain of using EOOBN in terms of local calculation and simultaneous computation. In a future work, learning techniques associated with such modelling will be considered.

## REFERENCES

- Amaral, L.A. and Ottino, J.M. (2004). Complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2), 147–162.

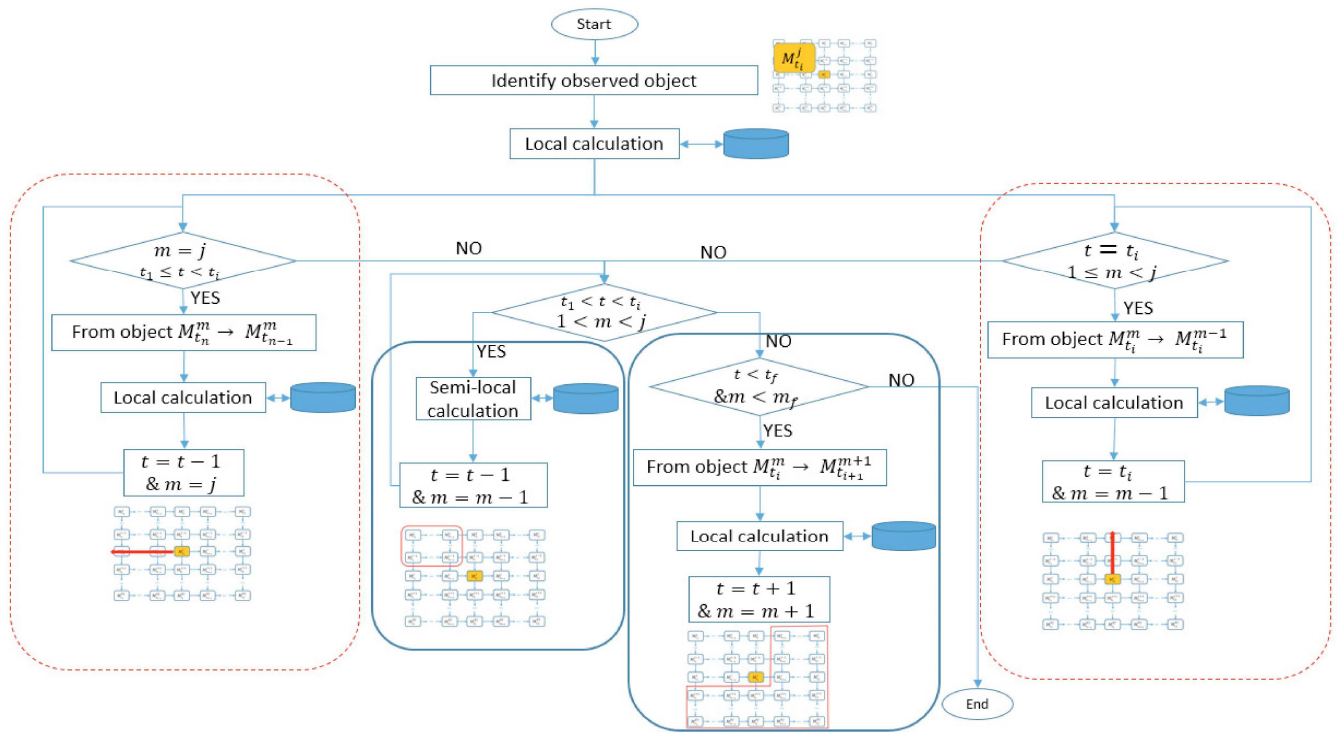


Fig. 11. Information propagation organigram

Bangsø, O. and Wuillemin, P.H. (2000). Object oriented bayesian networks a framework for topdown specification of large bayesian networks and repetitive structures.

Ben Hassen, W., Auzanneau, F., Incarbone, L., Pérès, F., and Tchangani, A.P. (2013). Omdtr using ber estimation for ambiguities cancellation in ramified networks diagnosis. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, 414–419. IEEE.

Bouzarour-Amokrane, Y., Tchangani, A., and Pérès, F. (2012). Defining and measuring risk and opportunity in boer framework for decision analysis. In *Proceedings of 9e International Conference of Modeling and Simulation - MOSIM2012*. Bordeaux, France.

Godichaud, M., Pérès, F., and Tchangani, A. (2012a). Optimising end-of-life system dismantling strategy. *International journal of production research*, 50(14), 3738–3754.

Godichaud, M., Tchangani, A., Pérès, F., and Iung, B. (2012b). Sustainable management of end-of-life systems. *Production Planning & Control*, 23(2-3), 216–236.

Jaeger, M. (2000). On the complexity of inference about probabilistic relational models. *Artificial Intelligence*, 117(2), 297–308.

Koller, D. and Pfeffer, A. (1997). Object-oriented bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, 302–313. Morgan Kaufmann Publishers Inc.

Langseth, H. and Bangsø, O. (2001). Parameter learning in object-oriented bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32(1-4), 221–243.

Langseth, H. and Nielsen, T.D. (2003). Fusion of domain knowledge with data for structural learning in object oriented domains. *The Journal of Machine Learning Research*, 4, 339–368.

Liu, Q., Tchangani, A., Kamsu-Foguem, B., and Pérès, F. (2015). Modelling a large scale system for risk assessment. In *Industrial Engineering and Systems Management (IESM), 2015 International Conference on*, 203–208. IEEE.

Murphy, K.P. (2002). *Dynamic bayesian networks: representation, inference and learning*. Ph.D. thesis, University of California, Berkeley.

Nielsen, T.D. and Jensen, F.V. (2009). *Bayesian networks and decision graphs*. Springer Science & Business Media.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Rabiner, L. and Juang, B.H. (1986). An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1), 4–16.

Weber, P. and Jouffe, L. (2006). Complex system reliability modelling with dynamic object oriented bayesian networks (doobn). *Reliability Engineering & System Safety*, 91(2), 149–162.

Wuillemin, P.H. and Torti, L. (2012). Structured probabilistic inference. *International Journal of Approximate Reasoning*, 53(7), 946–968.

Xiang, Y. and Lesser, V. (2000). Justifying multiply sectioned bayesian networks. In *MultiAgent Systems, 2000. Proceedings. Fourth International Conference on*, 349–356. IEEE.