



Editorial

Reducing the gap between experts' knowledge and data: The TOM4D methodology



Laura Pomponio^{a,b,*}, Marc Le Goc^a

^a Laboratoire des Sciences de l'Information et des Systèmes (LSIS), UMR CNRS 7296, AMU Domaine universitaire de Saint Jérôme, Avenue Escadrille Normandie Niemen, 13397 Marseille Cedex 20, France

^b Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS), CONICET, UNR, Ocampo y Esmeralda, S2000ECP Rosario, Argentina

ARTICLE INFO

Article history:

Received 10 March 2012

Received in revised form 16 June 2014

Accepted 4 July 2014

Available online 16 July 2014

Keywords:

Methodologies and tools

Data and knowledge

Knowledge Engineering

Knowledge modelling

Dynamic process modelling

ABSTRACT

Dynamic process modelling is generally accomplished from experts' knowledge through Knowledge Engineering (KE); however, the obtained models are sometimes deficient for interpreting the input data flow coming from the real process evolution perceived through sensors. This shortcoming lies in specialists' tacit knowledge, difficult to elicit, and in that certain process phenomena are unknown or unforeseen to experts. An alternative to complement the modelling task is to resort to a Knowledge Discovery in Database (KDD) process. Nevertheless, most KE approaches do not address the processing of knowledge obtained from data. This work proposes a KE methodology called Timed Observation Modelling For Diagnosis (TOM4D) which allows building dynamic process models from experts' knowledge and data where the obtained models can be compared and combined with models obtained through a KDD process in order to define a model more suitable to the dynamic process reality.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

A knowledge based system (KBS) which, by means of sensors and actuators, supervises a process taking place in a dynamic system, requires a model of the observed process. This model is, in principle, defined from experts' knowledge and can be built through the Knowledge Engineering (KE) discipline which provides methods, techniques and tools which facilitate and improve the task.

A model of experts' knowledge, called *Expert Model* or *Knowledge Model*, obtained through KE will be generally made up of a model of the process, and of a model describing how the expert reasons about this one. However, the process model may differ meanly from the process "reality" perceived through its instrumentation (sensors and actuators); and consequently, it may not meet the expectations in the supervision task carried out by a KBS. The problem originates principally in two issues. A first point is the expert's tacit knowledge, inherent in a human being, and the knowledge engineer's subjective interpretation; both central to KE. Another question is that certain properties, as for example those that are temporal, taking place during the process could be unknown or unforeseen by the expert.

An interesting alternative to deal with these problems is to resort to processes of Knowledge Discovery in Database (KDD), which use data mining techniques in order to acquire knowledge from data when obtaining a model of these ones. This model allows complementing experts' knowledge in order to build a more suitable process model. In an ideal world, the data model could be collated with the experts' model, and even it might extend the latter, in order to define a model that better reflects the process "reality".

* Corresponding author.

E-mail addresses: pomponio@fceia.unr.edu.ar, laura.pomponio@lsis.org (L. Pomponio).

Nevertheless, as we shall detail below, relating both models is generally complex due to principally two issues. First, the different natures of representation formalisms used in KE and KDD. Second, establishing the meaning, regarding experts' semantics, of models obtained from data is difficult owing to the different levels of conceptual abstraction in which the models are built.

In order to deal with these difficulties this work presents an approach for modelling dynamic systems from experts' knowledge and data describing a certain evolution of a process. Section 2 describes the gap between experts' knowledge and data. Section 3 expounds some concepts used in our approach. In Section 4 the main approaches of related works are presented. Section 5 introduces the Timed Observations Theory, which is the fundamental base of our proposal. Section 6 presents the TOM4D methodology which is the central focus of this work and whose application is described through a running example in Section 7. Section 8 positions our work regarding KE and KDD. Finally, Section 9 concludes this article. Besides, Appendix A describing some details of the example presented in Section 7 is included.

2. The gap between experts' knowledge and data

2.1. Different representation formalisms

Given a process about which an expert has knowledge, a model M_e of this process can be constructed from expert knowledge by applying KE techniques, as Fig. 1 illustrates. In turn, the process can be observed by a program which records data in a database describing its evolution over time. Thus, these data can be analysed by applying data mining techniques in a KDD process in order to obtain a model M_d of the process. Ideally, if it were possible to compare easily M_e and M_d , both models could be combined in order to obtain a more precise process model M_{PR} .

However, relating both models presents a first difficulty which lies in the different natures of the representation formalisms used in KE and KDD. Diverse KE methodologies and tools such as Roles-Limiting Methods and Generic Tasks [23], and later, CommonKADS [1,2], Protégé [3], MIKE [4,5], KAMET II [6,7], VITAL [8], etc. take as input experts' knowledge and produce as result a model of this knowledge being represented by frames, diagrams, related concepts, logic sentences, etc. However, KE approaches do not address the processing of knowledge obtained by means of a KDD process where a data model is obtained from a data set which is processed through techniques such as Decision Trees [9], Hidden Markov Chain [10], Neural Networks [11], Bayesian Networks [12], K Nearest-Neighbour [13], and SVM [14]. Generally, this data model is a numeric or probabilistic model which is hardly comparable with the expert's knowledge model represented as described above; consequently, the combination of the knowledge model with the data model is neither direct nor trivial.

In particular, we consider that the problem of combining both study fields resides in the lack of a more general theory on which to base both disciplines.

2.2. Different conceptual abstraction levels

Comparing and combining a knowledge model with a data model become difficult not only for their different representation modes but also because even assuming that there exists a formalism which allows describing both models, the discursive or conceptual level could not be the same. That is to say, the meaning of certain symbols in an Expert Model M_e could be meaningless in the context of the corresponding data model M_d , and vice versa, owing to the different levels of conceptual abstraction in which the discourses of each model are built. As Fig. 2 depicts, the reason for this is the large gap of conceptual semantic level that exists between experts' knowledge and data. An expert can describe why an engine does not work but, perhaps, in his or her discourse nothing appears about continuous functions, probabilities, etc. which are present in a data model obtained from sensors. Thus, the

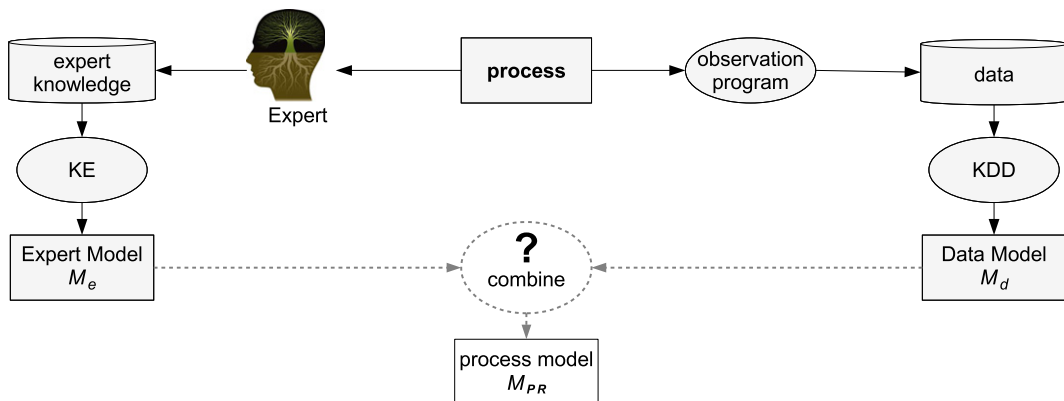


Fig. 1. Building a process model from two knowledge sources.

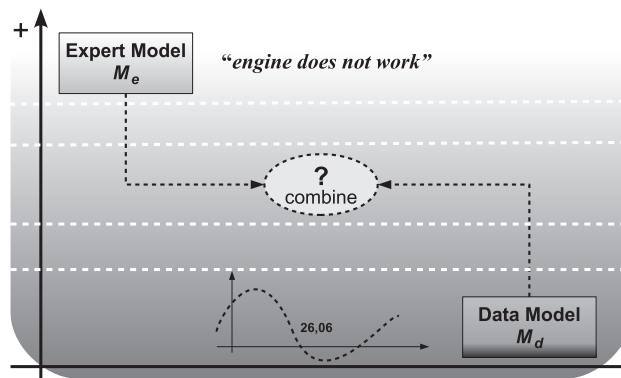


Fig. 2. Different levels of conceptual abstraction in which the discourses of each model are built.

abstraction level of the expert's knowledge model is higher than that one of the data model. Therefore, the data do not have direct significance in the expert's knowledge model and the expert's concepts are meaningless in the database. Owing to this, sometimes analysing the relation between possible models obtained from these two sources is not an alternative. Consequently, a translation from data into expert's vocabulary is indispensable, but at the same time carrying out this task can be extremely complex.

Our proposal aspires then to reduce the distance between experts' knowledge and data by proposing an approach which allows relating the aforesaid different abstraction levels.

3. Background

3.1. CommonKADS knowledge model

CommonKADS [2,1] is a methodology which offers a structured approach in the development of KBSs. In particular, the CommonKADS Knowledge Model is utilised in our approach. This model describes the types and structures of the knowledge required to accomplish a particular task; and thus, it acts as a tool that helps to clarify the structure of a knowledge-intensive information-processing task.

A CommonKADS Knowledge Model (or Expert Model) is a three layer model specifying different types of knowledge: *domain knowledge*, *inference knowledge* and *task knowledge*. In the lower level, particular domain knowledge and available information are described by specifying definitions of types and a knowledge base containing instances of these ones. In the middle level, inference knowledge establishes how the static information/knowledge structures of the domain knowledge can be used to carry out a reasoning process. Knowledge in this level specifies and structures inference steps, and it determines the semantic roles played by the domain rules and concepts in the inference structure. Finally, in the upper layer, task knowledge describes what objectives an application pursues and how these objectives can be accomplished through a decomposition into subtasks and inferences which will be performed in a particular order. That is, how some problem-solving goals are achieved by using the corresponding inference structure.

The mentioned model describes the cognitive process used by an expert in order to solve, with the appropriate domain knowledge, a given problem. One fundamental property is that the internal structures of this model are independent of the expertise domain because they are directly linked with a cognitive task (e.g., diagnosis). Consequently, CommonKADS provides a collection of predefined sets of model elements such as task templates and inference catalogues, which detail tasks and inferences typical for resolving a problem of a particular type. These templates also propose a characteristic structure for specifying the domain knowledge from the point of view of the selected type of task. Such templates, independent of domain knowledge, can be used in order to accelerate the acquisition and the modelling of the experts' knowledge.

In particular, this model is built as part of the analysis process in a way that is understandable by humans; therefore, it does not contain any implementation-specific term. Thus, this one is an important vehicle for communication with experts and users about the problem-solving aspects. Consequently, our approach uses the aforementioned model as a mean of interpreting and structuring the available knowledge.

3.2. Tetrahedron of States

The Tetrahedron of States (ToS) [15,16], formalised by Rosenberg and Karnopp [17] in the early 80's, is a framework that describes a set of generalised equations which are common to a wide variety of physical domains (electromagnetism, fluid dynamics, thermodynamics, etc.). ToS allows mapping physical variables of a specific domain into four classes of generalised variables and identifying five binary relationships between these ones. As Fig. 3 shows, the variables are effort (e), flow (f), impulse (p) and displacement (q);

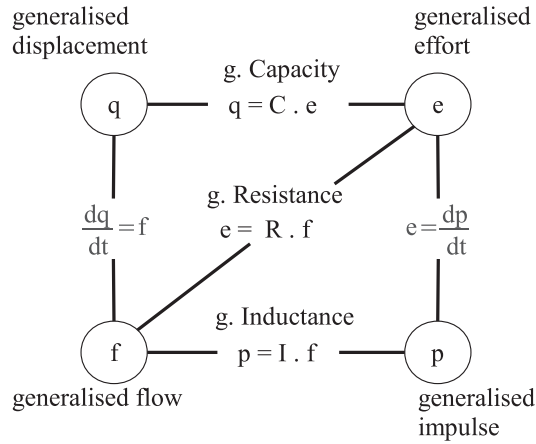


Fig. 3. Tetrahedron of States (ToS) (based on [16, p. 1728]).

and the binary relations are represented with equations defining three types of constants and one differential operator. The constants are Capacity (C), Resistance (R) and Inductance (I). The relations $q = C \cdot e$, $e = R \cdot f$ and $p = I \cdot f$ define the algebraic relations between the generalised variables and $\frac{dq}{dt} = f$ and $\frac{dp}{dt} = e$ define their temporal behaviour.

The generality of ToS allows, for example, establishing the relations between variables in the electric domain (Electric ToS). That is to say, *current* is mapped to f (i.e., generalised flow), *electric charge* to q (i.e., generalised displacement), *voltage* to e (i.e., generalised effort) and *magnetic flux* to p (i.e., generalised impulse); thus, the relations between the electric domain variables are clearly established from this framework.

Our modelling approach then resorts to ToS as framework of analysis and interpretation of knowledge about a dynamic process.

3.3. Abstraction levels

The notion of level can be found in different scientific domains and is conceived with purposes of understanding, explaining and solving problems or situations by means of classifying and representing objects, entities, events, concepts or experiences. Several works about levels of organisation, representation, complexity, abstraction, interpretation and even levels of reality give rise to broad ontological and epistemological debates. In particular, we consider that Newell's well-known work [18] and Floridi's work [19,20] contribute to an interesting analysis about the issue addressed in this work.

Newell [18] gives a characterisation of the standard computer system levels and proposes *The Knowledge Level Hypothesis* characterising a new level that verifies some properties of the first ones but that has important characteristics that make it particular. The standard hierarchy of the computer system levels presented by Newell consists of device level, circuit level, logic level (logic circuit, transfer register), symbol level (program) and configuration level (processor–memory switch). Common aspects of these levels are then the following: each level constitutes a *system* which will process a *medium* through *components* providing primitive processing, *laws of compositions* of these components and *laws of behaviour* which establish how the system behaviour depends on the behaviour of the components and their composition (i.e., the system structure). The knowledge level is above these ones where the laws of behaviour are characterised by the *principle of rationality*. Under these considerations a system at the knowledge

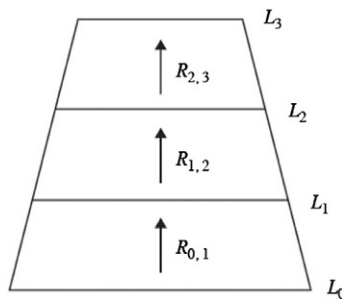


Fig. 4. Nested GoA with four levels of abstraction [19, p. 16].

level is an *agent* whose components are *goals*, *actions* and a *body*; and which processes its knowledge to determine the actions to take according to its behaviour laws (i.e., the principle of rationality). However, the aspect of having *laws of composition* in the computer levels, does not exist in the knowledge level. That is to say, there is not a structure; the relation among components is given by the mentioned principle.

In turn, Floridi [19,20] defines The Method of Levels of Abstraction, an epistemological notion of *level*, introducing concepts as *variable*, *observable*, *level of abstraction* (LoA), *behaviour*, *moderated Level of Abstraction* (mLoA) and *gradient of abstraction* (GoA). In this theory, the word *system* refers to the object of study, which may be a system in science or engineering but it may also be a domain of discourse, of analysis or a semantic system.

“A level of abstraction (LoA) is a finite but non-empty set of observables. [...]” [19, p. 10]. *“The behaviour of a system, at a given LoA, is defined to consist of a predicate whose free variables are observables at that LoA. The substitutions of values for observables that make the predicate true are called the system behaviours. A moderated LoA is defined to consist of a LoA together with a behaviour at that LoA.”* [19, p. 11].

In addition, an important concept of the Floridi's proposal is GoA. Different LoAs correspond to different representations or views of a system and a GoA allows varying the LoA in order to make observations at different granularity levels. A GoA is a set of moderated LoAs LoA_i and a family of relations $R_{ij} \subseteq LoA_i \times LoA_j$, $0 \leq i \neq j \leq n$; where for $i \neq j$, R_{ij} is the reverse of R_{ji} and the behaviour moderating each lower LoA is consistent with that specified by a higher LoA. In particular, two kinds of GoA are defined: *disjoint GoAs* and *nested GoAs*. We are interested in the nested GoAs whose views provide successively more information (Fig. 4). A nested GoA describes a system at each level and incrementally more accurately. In a nested GoA, any abstract observation has at least one concrete counterpart. *“The quantity of information in a model varies with the LoA: a lower LoA, of greater resolution of finer granularity, produces a model that contains more information than a model produced at a higher, or more abstract, LoA.”* [19, p. 18]. That is to say, the higher the level of abstraction, the fewer the information.

In particular, we believe that the Newell's ontological point of view and the Floridi's epistemological point of view can complement each other being of special interest to this work.

4. Related work

As established in [21], although creating knowledge is the central focus of both KE and KDD, knowledge creation has tended to be approached from one or the other perspective, rather than from a combined perspective. In particular, KE approaches do not address the interpretation of data models obtained through data mining techniques; and consequently, building dynamic process models from combining knowledge obtained from data with that obtained from experts is difficult.

In the KE domain, some of the well-known methodologies and tools for building KBS are CommonKADS [2,1], MIKE [4,5] and Protégé [3]. CommonKADS proposes a structured approach which consists in the creation of a collection of models that capture different aspects of the system to be developed, MIKE is an approach which puts emphasis on a formal and executable specification of the Expert Model. This methodology takes the CommonKADS Expert Model as a pattern for separating the different types of knowledge. Protégé is a methodology and an environment which provides tools for development of KBS emphasising on the utilisation of the notion of ontology.

CommonKADS is currently one of more important KE methodologies which has been extended and widely used for knowledge-based system development. Two well-known examples of the extensions are CoMoMas [22] and MAS-CommonKADS [23] for the multi-agent system development. The first one focuses on the building of conceptual agent models and their translation in an executable code. The second one, adapts the agent model and the communication model of CommonKADS to multi-agent systems. As an example, [24] applies MAS-CommonKADS to the web-based expert systems development. KEMNAD [25], in turn, is a methodology which uses CommonKADS in the development of negotiation agents and it establishes a pattern library for the design of these agents. In [26] a methodological approach for the mapping of knowledge models in production rules is presented, this approach combines CommonKADS Knowledge Models with a production rule representation language.

CommonKADS has also been applied to the health area; in particular [27,28] resort to this methodology. The former presents the development of a knowledge-based system for diagnosing breast cancer while the latter describes a knowledge-based approach to supervised classifier design emphasising in Alzheimer diagnosis.

A general model for representing temporal information through causal temporal constraint networks is presented in [29]. In this proposal, CommonKADS is used for temporally correlating the model symbolic information in order to represent temporal knowledge.

HeKatE (Hybrid Knowledge Engineering) [30], for its part, is a methodology for developing and designing complex rule-based systems for control and decision support.

Although diverse are the KE approaches, none of them addresses the processing of knowledge obtained from data by means of a KDD process; and consequently, the gap between expert models and data models is generally wide. The objective of our proposal is precisely to propose a KE approach which contributes in this sense to bring both disciplines in order to build more suitable process models.

5. Introduction to the Timed Observations Theory

The approach proposed in this work is based on the Timed Observations Theory (TOT) [31]. This theory provides a general framework for modelling dynamic processes from timed data by combining the Markov Chain Theory, the Poisson Process Theory, the Shannon's Communication Theory [32] and the Logical Theory of Diagnosis [33,34]. In order to present the TOM4D KE methodology, this section aims at introducing the main concepts of TOT.

5.1. Observed process

TOT defines a dynamic process as an arbitrarily constituted set $X(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$ of n functions $x_i(t)$ of continuous time $t \in \mathfrak{R}$. This set implicitly defines a set $X = \{x_1, x_2, \dots, x_n\}$ of n variable names x_i . According to this theory, a dynamic process $X(t)$ is monitored by an abstract observer program denoted by $\Theta(X, \Delta)$ which observes the functions $x_i(t) \in X(t)$; and then, it describes their evolution over time with a finite set $\Delta = \{\delta_j\}_{j=1, \dots, m}$ of constants δ_j (i.e. a number or a string) by registering timed messages (alarms, warnings, etc.) in a database. The program $\Theta(X, \Delta)$ is considered an abstract observer because its implementation can be carried out by either a standard computer (i.e., a monitoring program) or a human (i.e., an expert) providing and recording the timed messages in a database.

Definition 1. An *observed process* is a couple $(X(t), \Theta(X, \Delta))$ where $X(t) = \{x_i(t)\}_{i=1, \dots, n}$ is a finite set of time functions, $X = \{x_i\}_{i=1, \dots, n}$ is the corresponding finite set of variable names, $\Delta = \{\delta_j\}_{j=1, \dots, m}$ is a finite set of constant values and $\Theta(X, \Delta)$ is a program observing the evolution of the functions of $X(t)$.

TOT considers a message δ_i at time t_k in a database as a timed observation (δ_i, t_k) where $\delta_i \in \Delta$ is a constant value and t_k is the moment at which the observation is considered to occur. The timed observation is written by the abstract observer program $\Theta(X, \Delta)$ when a function $x_i(t)$ of continuous time enters a specific interval of values. That is, the specification of the program determines that

$$x_i(t_{k-1}) < \Psi_j \wedge x_i(t_k) \geq \Psi_j \Rightarrow \text{write}((\delta_i, t_k)) \quad (1)$$

In general terms, the program registers a timed observation (δ_i, t_k) in a memory whenever a particular predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ is satisfied; in this case, the mentioned predicate is $x_i(t_{k-1}) < \Psi_j \wedge x_i(t_k) \geq \Psi_j$.

Fig. 5 illustrates a function $x_i(t)$, where values above Ψ_j are interpreted as δ_i by an observer program $\Theta(\{x_i\}, \{\delta_i\})$; that is, when $x_i(t)$ enters $[\Psi_j, +\infty)$. For example, considering timed data in a database of the form “`yymmddhhmmss, message`” where `yymmddhhmmss` is a time stamp and `message` is a value determined by a monitoring program $\Theta(X, \Delta)$; a message “080313132225, high” will be considered a timed observation (δ_i, t_k) where $t_k = 080313132225$ and $\delta_i = \text{high}$; that is to say, $(\delta_i, t_k) = (\text{high}, 080313132225)$. In particular, the observer program $\Theta(\{x_i\}, \{\text{high}\})$ has written this observation indicating that the function $x_i(t)$ entered a particular interval at time 080313132225; e.g., values above Ψ_j are interpreted by the observer program as a high level of $x_i(t)$.

TOT establishes that the existence of a timed observation (δ_i, t_k) in a database allows to infer that the mentioned observation has been recorded by a program $\Theta(\{x_i\}, \{\delta_i\})$, perhaps unknown, which implements the abstract logical sentence described in Expr. (2), for a set Γ of arbitrary time instants.

$$\forall t_k \in \Gamma, \theta(x_i, \delta_i, t_k) \in \Theta \Rightarrow (\delta_i, t_k) \in \Omega \quad (2)$$

This expression associates the set Θ of all the assignments to a ternary predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ with the set Ω of all the timed observations carried out by the program $\Theta(\{x_i\}, \{\delta_i\})$ (i.e., the database). A timed observation (δ_i, t_k) is then interpreted as the

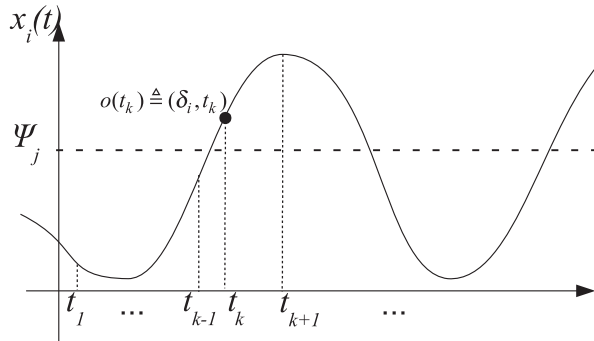


Fig. 5. Spatial segmentation of the $x_i(t)$ function of time. The symbol \triangleq in the figure denotes rewriting or “correspond to” establishing that a timed observation (δ_i, t_k) can also be written like $o(t_k)$.

logical consequence of the assignation of the values x_i , δ_i and t_k to the mentioned ternary predicate; that is, $\theta(x_i, \delta_i, t_k)$. This means that the timed observation (δ_i, t_k) was recorded when the program $\Theta(\{x_i\}, \{\delta_i\})$ assigned the values x_i , δ_i and t_k to the predicate $\theta(x_\theta, \delta_\theta, t_\theta)$.

Given the sentences (1) and (2), the general meaning “is” can be always provided to the predicate θ so that the timed observation (δ_i, t_k) is interpreted as “at time t_k , x_i is δ_i ”. Considering that x_i is associated with a function $x_i(t)$, the meaning “equal” can also be attributed to the predicate θ , which leads to the following abuse of language: $\theta(x_i, \delta_i, t_k)$ means $Equal(x_i, \delta_i, t_k)$; i.e., $x_i(t_k) = \delta_i$. Thus, this theory considers that a message contained in a database is a timed observation (δ_i, t_k) written by a program $\Theta(X, \Delta)$ which observes a time function $x_i(t)$ and implements the abstract Expr. (2). In our example, $(high, t_k)$ indicates that a program $\Theta(\{x_i\}, \{high\})$, observing a time function $x_i(t)$ and defining implicitly a predicate $\theta(x_\theta, \delta_\theta, t_\theta)$, has considered $\theta(x_i, high, t_k)$ true and then it has written the timed observation $(high, t_k)$ in the database Ω .

For the sake of generality, it is important to note that a predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ is satisfied when the corresponding time function $x_i(t)$ matches against a behavioural model [35]. Such a behavioural model can be as simple as the switch of an interrupter or, requiring very complex techniques, such as signal processing techniques for artificial vision.

5.2. Timed observation and observation class

According to the concepts previously introduced, the interpretation of a timed observation (δ_i, t_k) is precisely the assigned predicate $\theta(x_i, \delta_i, t_k)$. However, it is noteworthy that the program $\Theta(\{x_i\}, \{\delta_i\})$ might contain errors and a timed observation (δ_i, t_k) could have been written in a database although the assertion $\theta(x_i, \delta_i, t_k)$ is not really true. The notion of timed observation is established in Definition 2 along with other concepts.

Definition 2. Let $X(t) = \{x_i(t)\}_{i=1 \dots n}$ be a finite set of time functions; let $X = \{x_i\}_{i=1 \dots n}$ be the corresponding finite set of variable names; let $\Delta = \{\delta_j\}_{j=1 \dots m}$ be a finite set of constant values; let $\Theta(X, \Delta)$ be a program observing the evolution of the functions of $X(t)$; let $\Gamma = \{t_k \mid t_k \in \mathcal{R}\}$ be a set of arbitrary time instants (i.e. a stochastic clock); and let $\theta(x_\theta, \delta_\theta, t_\theta)$ be a predicate implicitly determined by $\Theta(X, \Delta)$. Then,

- A *timed observation* $(\delta_j, t_k) \in \Delta \times \Gamma$ on $x_i(t)$ is the assignation of values x_i , δ_j and t_k to the predicate $\theta(x_\theta, \delta_\theta, t_\theta)$ such that $\theta(x_i, \delta_j, t_k)$ (by definition, $o(t_k)$ also denotes a timed observation, i.e., $o(t_k) \triangleq (\delta, t_k)$) and
- A *scenario* is a set Ω of time-ordered sequences ω_i of timed observations; that is, $\omega_i: \{1, \dots, r\} \rightarrow \Delta \times \Gamma \mid r \in \mathbb{N} \wedge \forall i, j \in \{1, \dots, r\}, i \leq j, (\omega(i) = o(t_k) \wedge \omega(j) = o(t_r) \Rightarrow t_k \leq t_r)$ and $\Omega = \{\omega_i\}_{i=1 \dots q}$ constitutes the set of all timed observations; i.e., the database.

Timed observations on a particular function $x_i(t)$ implicitly determine a variable x_i which assumes discrete values δ_j and describes the evolution of $x_i(t)$ according to the interpretation implemented in the observer program. That is to say, when $\Theta(X, \Delta)$ considers $\theta(x_i, high, t_k)$ true, and then writes $(high, t_k)$ in Ω , it is implicitly defining a discrete variable x_i which assumes at least the value $high$. Consequently, a timed observation (δ_j, t_k) and the implicit existence of an associated discrete variable x_i enable to define the notion of observation class, other important concept of the TOT. An observation class is a set $C_{x_i} = \{(x_i, \delta_j) \mid \delta_j \in \Delta_{x_i}\}$ which associates a variable x_i with the constant values δ_j that the variable can assume. Owing to the fact that this concept binds a constant $\delta_j \in \Delta_{x_i}$ and a variable $x_i \in X$, a timed observation (δ_j, t_k) is an occurrence at time t_k of an observation class $C_{x_i} = \{(x_i, \delta_j) \mid \delta_j \in \Delta_{x_i}\}$. The following definition specifies this concept.

Definition 3. Let $X(t) = \{x_i(t)\}_{i=1 \dots n}$ be a finite set of time functions whose evolution is observed by an observer program; let $X = \{x_i\}_{i=1 \dots n}$ be the corresponding finite set of variable names; let $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ be such that $\Delta_{x_i} = \{\delta_j\}_{j=1 \dots p}$ is a set of constant values which can be assumed by x_i . An *observation class* C_{x_i} associated with a variable $x_i \in X$ is a set $C_{x_i} = \{(x_i, \delta_j) \mid \delta_j \in \Delta_{x_i}\}$.

One of the major consequences of the notion of observation class is the following principle.

Principle 1. Decomposition Law

Given an observed process $(X(t), \Theta(X, \Delta))$ where the program $\Theta(X, \Delta)$ is memoryless and chooses independently the constants $\delta_j \in \Delta$; then, defining a set $C = \{C_i\}_{i=1 \dots m}$ of m single observation classes $C_i = \{(x_i, \delta_j)\}$ (with $x_i \in X, \delta_j \in \Delta_{x_i}$) corresponds to the decomposition of $(X(t), \Theta(X, \Delta))$ in a set of m observed sub-processes $(X(t), \Theta_i(X_i, \Delta_i))$ so that $(X(t), \Theta(X, \Delta)) = \bigcup_{i=1}^m (X_i(t), \Theta_i(X_i, \Delta_i))$.

The memoryless and the independence conditions of the program $\Theta(X, \Delta)$ are, in practice, very easy to check. The important consequence of this principle is that the m observed sub-processes $(X_i(t), \Theta_i(X_i, \Delta_i))$ define, by construction, m stochastic clocks $\Gamma_i = \{t_k \mid t_k \in \mathcal{R}\}$, so that $\Gamma = \bigcup_{i=1}^m \Gamma_i$. The fundamental interest of this property resides in the definition of an observation class as a singleton $C_i = \{(x_i, \delta_j)\}$, $x_i \in X, \delta_j \in \Delta_{x_i}$; i.e., C_i contains one and only one couple (x_i, δ_j) . Thus, each $C_i = \{(x_i, \delta_j)\}$ determines an observed sub-process $(X_i(t), \Theta_i(X_i, \Delta_i))$ where $X_i(t) = \{x_i(t)\}$, $X_i = \{x_i\}$, $\Delta_i = \{\delta_j\}$. Consequently, m single observation classes of C decompose an observed process in m minimal processes, which produce m stochastic clocks Γ_i , where each Γ_i is the set of the time instants of the

occurrences of the corresponding observation class C_i . Thus, this decomposition facilitates the analysis of the relations between the functions $x_i(t)$ which make up the core of the process $X(t)$.

5.3. Binary temporal relations

As presented above, **Principle 1** allows to decompose a process in minimal sub-process; and then, to analyse the temporal relations between these ones through the corresponding observation classes.

Definition 4. A binary temporal relation $r(C_i, C_j, [\tau^-, \tau^+])$ is an oriented (sequential) relation between two observation classes C_i and C_j restricted by the interval of time $[\tau^-, \tau^+]$, $\tau^-, \tau^+ \in \mathbb{R}$ such that, the elapsed time between an occurrence of C_i and an occurrence of C_j is greater than or equal to τ^- and less than or equal to τ^+ .

Considering an observation class $C_i = \{(x_i, \delta_i)\}$, we denote with $C_i(t_k)$ an occurrence at time t_k of the class C_i ; that is, the timed observation (δ_i, t_k) can also be rewritten as $C_i(t_k)$ (i.e., $C_i(t_k) \triangleq (\delta_i, t_k)$). The following definition is then introduced.

Definition 5. Let $C = \{C_l\}_{l=1, \dots, m}$ be a set of m observation classes corresponding to an observed process $(X(t), \Theta(X, \Delta))$ and let $\omega = (C_{q1}(t_1), C_{q2}(t_2), \dots, C_{qn}(t_n))$, $C_{ql} \in C$, $l = 1, \dots, n$ be a sequence of timed observations of the classes of C . A binary sequential relation $r_{ij}(C_i, C_j, [\tau^-, \tau^+])$ between two observation classes $C_i, C_j \in C$ is said to be observed in ω if there exist two timed observations $C_i(t_k), C_j(t_p)$, $t_k \leq t_p$ in ω which satisfy the timed constraint $[\tau^-, \tau^+]$; that is, if $(t_p - t_k) \in [\tau^-, \tau^+]$.

Consider the temporal relation $r(C_a, C_b, [2, 5])$ and the sequence $\omega = (C_e(18), C_a(19), C_c(20), C_b(22), C_c(24))$ of timed observations of the classes C_e, C_a, C_b and C_c . As Fig. 6 depicts, the relation mentioned is observed in ω owing to the fact that there exist the observations $C_a(19)$ and $C_b(22)$, of classes C_a and C_b respectively, such that the elapsed time between them $(22 - 19 = 3)$ is in the interval $[2, 5]$.

Naturally, the confidence respect to the representativeness in a sequence ω of a temporal binary relation $r(C_i, C_j, [\tau^-, \tau^+])$ is linked with the ratio between the number of sequential occurrences between pairs of observation classes in ω and the number of sequential occurrences $(C_i(t_k), C_j(t_p))$ in ω such that $t_p - t_k \in [\tau^-, \tau^+]$. Thus, the TOT framework introduces notions of probability that lead to a KDD process called TOM4L (Timed Observation Mining for Learning) which is beyond of the scope of this article. For more details, the reader is referred to [36–39].

Definition 6. An abstract chronicle model is a set M of binary temporal relations.

Definition 7. An abstract chronicle model is a path if it is made up of a set of binary temporal relations contiguous with each other of the form $M = \{r(C_i, C_{i+1}, [\tau_1^-, \tau_1^+]), r(C_{i+1}, C_{i+2}, [\tau_2^-, \tau_2^+]), \dots, r(C_{n-1}, C_n, [\tau_n^-, \tau_n^+])\}$.

In particular, a path provides meaning to each sequence ω of timed observations which satisfies all its relations since the path establishes a sequence of binary temporal relations which links observation classes each other.

Principle 2. Induction from timed observations

Let $(X(t), \Theta(X, \Delta))$ be an observed process defining a particular set $C = \{C_l\}_{l=1, \dots, m}$ of m observation classes such that $C_i = \{(x_i, \delta_i)\}$ and $C_j = \{(x_j, \delta_j)\}$ belong to C . Let ω be a sequence of timed observations provided by $(X(t), \Theta(X, \Delta))$. Let ω_i and ω_j be the sub-sequences of ω constituted only by timed observations of the classes C_i and C_j respectively. Inducing a binary relation $r_a(\Gamma_i, \Gamma_j)$ between two stochastic clocks Γ_i and Γ_j from two sequences ω_i and ω_j corresponds to induce a binary temporal relation of the form $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ which subsumes the existence of a relation between the variables x_i and x_j respectively.

Principle 2 highlights the importance of the concept of observation class in the TOT framework. Assuming that there exists a relation between two observations classes C_i and C_j corresponds to assuming that there exists a relation between the corresponding variables x_i and x_j ; and consequently, between the two corresponding functions $x_i(t)$ and $x_j(t)$ of the dynamic process $X(t)$. Thus, this principle establishes the relations between constants δ_i , timestamps Γ_i and functions $x_i(t)$ which are organised on the notion of

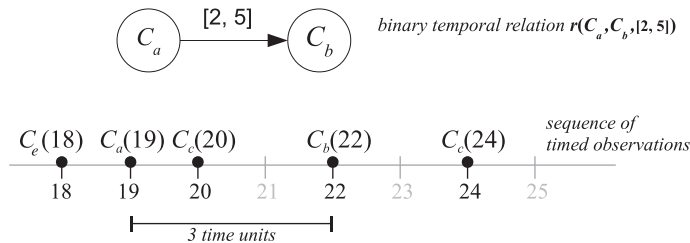


Fig. 6. Binary temporal relation observed in a sequence of timed observations.

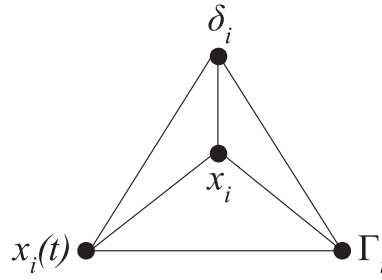


Fig. 7. Relations between the TOT basic elements.

variable whose definition determines six binary relations as Fig. 7 depicts. This provides a powerful tool to analyse the logical coherence within a knowledge corpus about a dynamic process.

Principle 3. Coherence Law

What is assumed about the relations between two variables x_i and x_j must be coherent with what is assumed about the relations between the corresponding constants, observation classes, stochastic clocks and the corresponding functions $x_i(t)$ and $x_j(t)$; and reversely.

These TOT principles are a fundamental basis of the TOM4D methodology which is presented in next section.

6. General framework of the TOM4D methodology

Behaviour assessment is based on knowledge intensive tasks of monitoring, diagnosis and prognosis. Monitoring task requires knowledge in order to infer current behaviour and to categorise it as desirable or undesirable. Diagnosis task requires knowledge in order to infer the causes of undesirable behaviour. Prognosis task requires knowledge in order to infer undesirable behaviour, potentially future, which can result from the current situation described by the monitoring and the diagnosis tasks. The quality of the knowledge corpus required by these tasks is directly linked with the behaviour assessment task. Thus, the Knowledge Engineering methodology used in order to acquire and to represent the required knowledge corpus must provide tools that guarantee an adequate level of quality. TOM4D (Timed Observations Modelling for Diagnosis), the KE modelling approach for dynamic systems, proposed in this work, pursues this aim.

6.1. Knowledge and model

Generally speaking, knowledge results from the interaction between an information flow and an arbitrary purpose. Information comes from all the possible sources such as beliefs, observations, experimentations, scientific axioms, and sensors [40–42]. The interaction is assumed by humans which define their purpose according to their own expectations [43–45] and is basically an interpretation of the information flow which traverses a thinking human [46,47]. In order to define TOM4D, we use the following operational definition of knowledge.

Definition 8. Knowledge results from an intentional interpretation of a flow of information.

This definition establishes a relation between knowledge, information and a purpose (i.e., an intention). The purpose is always defined by humans. In the TOT framework, the purpose is implemented in an observer program $\Theta(X, \Delta)$ which can be “executed” by either a human or a computer.

According to TOT, the information flow is a sequence ω of timed observations. By construction, owing to Principle 1, ω defines a set $C = \{C_i\}_{i=1, \dots, m}$ of observation classes $C_i = \{(x_i, \delta_i)\}$. The set C provides then the reading keys which allow interpreting ω as a sequence of assigned predicates $\theta(x_i, \delta_i, t_k)$. This set must then be constituted according to a particular purpose by containing each observation class $C_i \in C$ a piece of the global purpose so that a path (cf. Definition 7) represents a model of scenarios according to the mentioned purpose.

Considering that a path is a sequence of binary temporal relations $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ and taking into account that each of the observation classes C_i and C_j are made up according to a particular purpose, the notion of knowledge in Definition 8 can then be reformulated according to the TOT induction principle (Principle 2) as follows.

Definition 9. Any relation logically consistent with a binary temporal relation of the form $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ is a piece of knowledge.

In order to evaluate the behaviour of an observed process $(X(t), \Theta(X, \Delta))$, it is necessary to elicit an adequate knowledge corpus and to carry out its representation under the form of a model with the purpose of making possible the emergence of specific patterns, which are directly linked with the behaviour to be assessed. Defining this behaviour is then crucial for modelling a dynamic process;

in particular, it specifies de facto the abstraction level of the model. In particular, a model is made up of a particular arrangement of signs (i.e., an alphabet) where the meaning is given precisely by the specific disposition (i.e., the grammar) chosen by the modeller in order to share her or his knowledge. Thus, the representation of a knowledge corpus requires a set of rules which defines the authorised arrangements. This leads to the following notion of model according to TOT.

Definition 10. A model is an organised set of knowledge representations.

In TOT, being concerned with the evolutions of a process over time, the knowledge under consideration is linked with the relations between functions of time $x_i(t)$, constants δ_i and stochastic clocks Γ_i . These relations are organised around the notion of variable x_i (cf. Fig. 7) where a piece of knowledge belongs then to the following three fundamental categories.

1. *Relations between the functions $x_i(t)$ of the process $X(t)$.* This category of knowledge is called the *structural knowledge* owing to the fact that the functions are the constituents of the process $X(t)$, i.e., the components.
2. *Relations among the constants δ_i of the observer program $\Theta(X, \Delta)$.* This category is called the *functional knowledge* because the relations between the constants $\delta_i \in \Delta$ can be represented with logical rules linking together sub-sets $\Delta_i \subset \Delta$ and then, abstract mathematical functions can be specified under the form of tables of values.
3. *Relations between the stochastic clocks Γ_i .* This category is called the *behavioural knowledge* due to that these relations describe the links between the evolutions of the functions $x_i(t)$ of the process $X(t)$.

As Fig. 7 illustrates, these three categories of relations are linked together. A specific set Δ of constants δ_i will lead an observer program $\Theta(X, \Delta)$ to describe the evolution of a process $X(t)$ with a particular set Γ of timestamps. The role of the concept of variable is to provide the mean to analyse the consistency of these three categories of knowledge about a process $X(t)$. Consequently, the concept of variable defines a supplementary category of knowledge, a kind of “meta-knowledge”, which fundamentally defines the way a dynamic process $X(t)$ is perceived by humans; i.e., the modeller point of view.

Definition 11. The modelling activity of a dynamic process $X(t)$ aims to represent, according to a formalism, the elicited knowledge and to distribute the knowledge representations over a structural model, a functional model and a behavioural model according to the definition of a particular set X of variables.

By construction, X is a subset of all the variables involved in a dynamic process. The only rational way to specify this subset is precisely the modelling purpose, which determines what variables play a role significant for the established purpose. The other variables can be forgotten, at least in a first step. Thus, X defines the process according to a modelling purpose; and consequently, it determines the abstraction level of the model.

6.2. Modelling principles

The main peculiarity of the TOM4D KE methodology is to allow the modelling from both timed observations and experts' a priori knowledge. Thus, a TOM4D process model presents the following properties:

1. the model can be faced with real world timed data; i.e., sequences of timed data making up scenarios and,
2. any experts' assertion about the dynamic process can be faced with the model.

These two principal properties are fundamental when reasoning about a dynamic process. The first one allows assessing the adequateness of a model with real world data and the second one allows checking experts' knowledge with an already validated model. Furthermore, the latter constitutes a way to analyse completeness and coherence of the model under construction. These two properties come from the modelling principles which are presented below.

The TOM4D purpose is to build a model which resides at the same abstraction level as that one of experts' knowledge, it is logically coherent and it is as complete as possible. Providing a coherent model at the right abstraction level is the main modelling law of TOM4D; completeness, although desired, does not constitute a primary condition.

In order to assess the abstraction level of the model, TOM4D resorts to a knowledge interpretation framework with the purpose of introducing, in the modelling process, the semantic content provided by experts in a gradual and controlled way. This framework is based on the conceptual models of CommonKADS [2,1] and the Tetrahedron of States (ToS) [16,17], which were introduced in Section 3; the former, in order to interpret experts' knowledge, and the latter, in order to analyse the soundness of the interpretation. Hence, from this conceptual framework, a particular set X of variables constituting the core of the modelled process can be defined so that the six types of binary relations illustrated in Fig. 7 can be examined.

For the purpose of assessing the model coherence, knowledge representations are distributed over three basic models according to the three aforesaid category of knowledge. TOM4D has been designed as a primarily syntax-driven approach [48–50]. Consequently, Reiter's Logical Theory of Diagnosis [33] can be used in order to analyse properties of the produced models according to the Coherence Law (Principle 3), and in return, in order to supply tools which allow assessing experts' knowledge [34]. Thus, the TOM4D knowledge representation formalism is based on the Formal Logic; i.e. the predicate calculus.

In this methodology, any proposition formulated by an expert is, by hypothesis, an assertion about a property of the process to be modelled. These propositions are concerned with structure, functions, behaviour or role of the process. Therefore, the completeness of a TOM4D model corresponds to the property of such a model of allowing providing any proposition that an expert can formulate about the corresponding process. In this way, for assessing the completeness of a TOM4D model, it is necessary but not sufficient to

check first the completeness of the set X of variables and next, to verify that all the possible binary relations between two variables $x_i, x_j \in X$ have been examined according to the three fundamental dimensions of knowledge (i.e., structural, functional and behavioural). Thus, this syntactic modelling approach facilitates the analysis of the completeness of a model.

Consequently, the combination of a syntactic and a semantic approach in the TOM4D modelling principles allows providing knowledge engineers with tools in order to control the knowledge acquisition process and to identify the main concepts of the dynamic process (variables, constants, values, thresholds, components, states, etc.).

Five fundamental modelling principles, constituted from the variable notion, determine the proposed methodology.

1. Variable localisation

A function $x_i(t)$ is a signal provided by a sensor located at a particular place defined as a component. So a function $x_i(t)$ specifies a variable x_i , a component c_i and a binary relation which associates x_i to c_i . Hence, a variable x_i is always associated with a sensor which is physically located on a component c_i . Thus, any variable $x_i \in X$ must be associated with one and only one component c_i .

2. Multi-value variable

A variable x_i is necessarily defined over a set Δ_{x_i} of possible values containing at least two elements. This means that when the experts' knowledge defines only one value δ_i for a variable x_i , the knowledge engineer must introduce in Δ_{x_i} another constant (e.g., ϕ_i) meaning “not δ_i ” (i.e. $\Delta_{x_i} = \{\delta_i, \phi_i\}$ and $\phi_i \triangleq \neg\delta_i$). This principle is a direct consequence of the TOT *spatial segmentation principle* (Fig. 5).

3. Discernible state

The assignment of a value δ_i to a variable x_i is necessarily the result of an observable modification in the dynamic process. Such an observable modification marks a state transition which is conditioned by an occurrence of an observation class $C_i = \{(x_i, \delta_i)\}$; and besides, a temporal binary relation $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ defines a unique “discernible state”. Owing to the fact that the notion of discernible state is linked with the observer program $\Theta(X, \Delta)$, this notion is weaker than the classical notion of state of the Discrete Event System (DES) domain.

4. Separation of different types of knowledge

A process model is constituted by four models where each one represents a specific category of knowledge; that is, a structural model SM containing all the structural knowledge, a functional model FM containing all the functional knowledge, a behavioural model BM containing all the behavioural knowledge, and a perception model PM containing the perception knowledge. As Fig. 8 illustrates, this constitutes the TOM4D multi-modelling framework [16,51,52] which is described below.

5. Symbol driven modelling

The knowledge interpretation aims at identifying the minimal set of symbols denoting variables, constants or components and the minimal set of relations between them. The logical properties coming from these minimal sets are necessary and sufficient to complete the model. The introduction of a symbol that is not associated with an element of the domain knowledge is prohibited. In particular, a CommonKADS domain schema must be used to identify the concepts that play the role of variable in the knowledge corpus.

Fig. 8 illustrates the aforesaid conceptual framework of the TOM4D modelling process which is centred on the TOT notion of variable. The structural model SM associates any variables used in a function of the functional model FM with a component (or a component aggregate). A timed observation class of the behavioural model BM associates a variable to a constant. Consequently, due to the fact that a timed observation is an occurrence of an observation class, the mentioned observation corresponds to the assignment of a constant to a variable. This variable plays a role in at least one function of the functional model FM and then it is associated with one of the components of the structural model SM . The perception model PM , for its part, defines the minimal set of variables and the corresponding minimal set of constants which are required to define the goals of a dynamic process. Therefore, the perception model specifies a minimal set of constraints that the functional, the structural and the behavioural models must respect. This means that any relation defined between these models must be consistent with the constraints of the perception model PM .

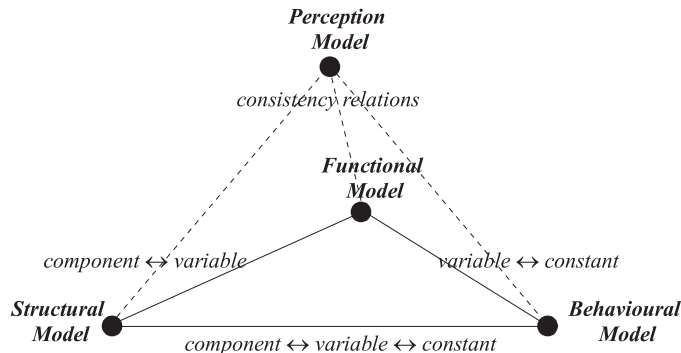


Fig. 8. Relations between TOM4D models.

The five previously introduced principles constitute a strong logical basis for the modelling work. From the identification of a variable, the possible values that this one can take over time, its corresponding component and its observation classes can be identified. By carrying out this for all the variables, the knowledge engineer defines the state space of the dynamic process. Next, all the possible and the impossible relations between two elements (variable, constant, component and observation classes) are examined through the semantic properties of these ones. Thus, the organisation of the resulting knowledge representations in the four models leads to an operational model of the dynamic system.

6.3. Modelling process

The modelling process aims to produce a generic model of a dynamic process from the available knowledge and data. The available knowledge contains, by definition, a description of the modelling goal. Three main phases constitute this process: knowledge interpretation, process definition and generic modelling. The logical dependencies of these three phases are represented in Fig. 9 like a CommonKADS inference structure where ovals represent inference steps and rectangles define concept roles. The TOM4D modelling process is then described as a reasoning process which provides a model of a dynamic process given the available knowledge. Therefore, the exploitation method of such an inference structure must be defined according to the modelling problem; this explains why TOM4D is not a method but a methodology. How the control flow of the modelling process is carried out is not part of this structure. Clearly, when a problem requires a modelling phase, any multi-modelling method is cyclical and each step can require returning back to previous phases with the objective of revising experts' knowledge or modelling decisions.

A partial description of the dynamic process is provided by Ω which allows establishing a primary set of functions, variables, constants and stochastic clocks. The identification of these elements is accomplished through the available knowledge which is interpreted with a CommonKADS template. This template describes the usual cognitive task that experts carry out when analysing the behaviour of the dynamic process. The analysis of the relations between the primary elements leads to a set of primary knowledge representations which are organised in an initial model called the scenario model $M(\Omega) = \langle SM(\Omega), FM(\Omega), BM(\Omega) \rangle$. This model relates a structural model $SM(\Omega)$, a functional model $FM(\Omega)$ and a behavioural one $BM(\Omega)$ of the process according to Ω . Clearly, this primary model $M(\Omega)$ is not complete due to that it is restricted to what appears in the sequences $w_i \in \Omega$. However, if Ω is representative of the dynamic process behaviour, such a model is sufficient to provide a first view of the process to be modelled.

The second main step of the TOM4D modelling process is the process definition which aims to provide the boundary of the process in terms of time functions $x_i(t)$, the operating goals of this one and its normal and abnormal operating modes. The actual process $P(t)$ is then restricted to a set $X(t) = \{x_i(t)\}_{i=1,\dots,n}$ of time functions $x_i(t)$. The operational goals and the operating modes of the process are represented with sets of conjunctions whose propositions are of the form $\forall t \in \mathfrak{R}, x_i(t) \geq \Psi_i$ (or $x_i(t) \leq \Psi_i$). This set describing operating modes is partitioned in two according to desired and undesired operating modes; establishing thus, the sets of normal

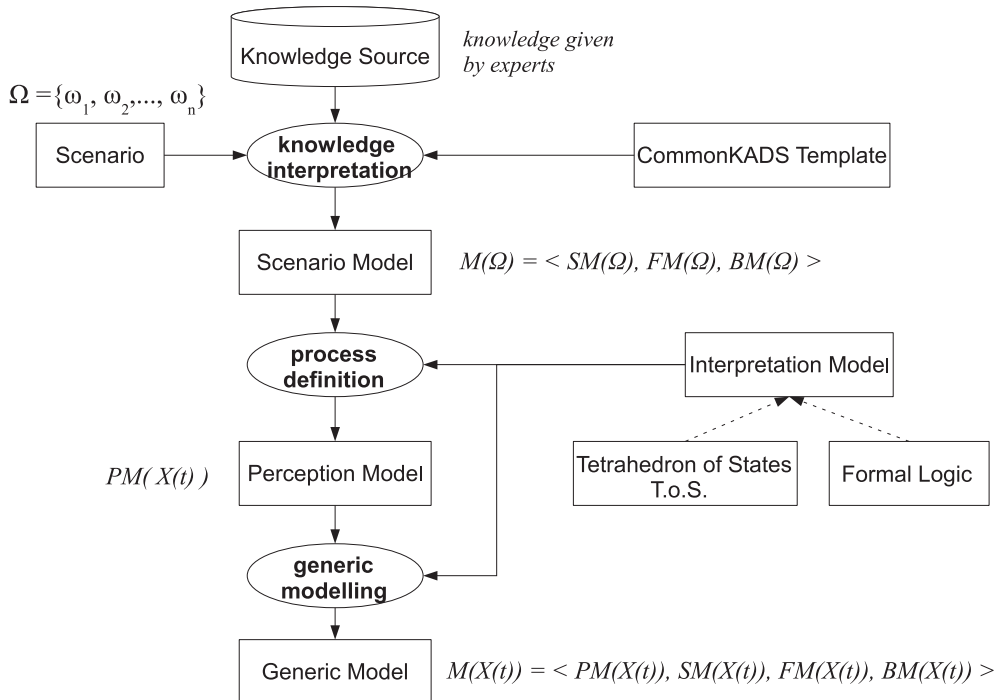


Fig. 9. General structure of the TOM4D modelling process.

behaviours and abnormal behaviours as described in Reiter's Theory of Diagnosis [33]. The scenario model $M(\Omega)$ resultant from the knowledge interpretation step and the conceptual frameworks of Formal Logic and the Tetrahedron of States (ToS) [17] of the Newton classical physics are input knowledge in this phase. The mentioned frameworks constitute the only semantic contexts which allow the logical and the physical interpretations of the symbols used to denote variables and constants defined in the scenario model $M(\Omega)$. The role of these frameworks is to provide the set of laws which make possible to control the semantic content represented in the model. The process definition step gives as result the perception model $PM(X(t))$ of the dynamic process establishing how the process is perceived by experts. Only what can be derived from this model can be taken into account in order to build the structural, the functional and the behavioural models of the process. In other words, $PM(X(t))$ defines the level of abstraction used by experts with the purpose of reasoning about the process $P(t)$.

Generic modelling is the last step. This phase defines the set $X = \{x_i\}_{i=1, \dots, n}$ of the variables and their respective domains Δ_{x_i} , it identifies the corresponding sets of components, observation classes and logical relations between the constants of $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$, and it distributes the representation of the pertinent binary relations (Fig. 7) over the three models (i.e., the structural model $SM(X(t))$, the functional model $FM(X(t))$ and the behavioural model $BM(X(t))$). The objective is then to define a model $M(X(t)) = \langle PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) \rangle$ of a type of process where $M(X(t))$ is coherent with the scenario model $M(\Omega)$ and, in turn, is a generalisation of the latter. This phase is accomplished using the perception model $PM(X(t))$ and the available knowledge according to the representation and the interpretation laws defined following the Formal Logic and the ToS frameworks. These frameworks allow the systematic exploration of the whole semantic and syntactic spaces which constitute the global modelling space: the semantic space is defined with the physical dimension of the variables x_i (typically according to the International System of Units), and the syntactic space is defined as the matrix of all the pairs (a, b) that can be made with the alphabet of the symbols used to represent the knowledge.

6.4. Semantic control

One of the main difficulties with the application of KE methodologies is the analysis of the semantics contained in a knowledge corpus provided by an expert. The knowledge engineer has no elements necessary to analyse the coherence and the scope of a new knowledge piece of a particular domain. This difficulty increases drastically when working with a dynamic process owing to the fact that propositions which seem physically reasonable, but are not coherent with the current version of the knowledge model, can be admitted. KE provides then tools in order to facilitate the semantic analysis.

TOM4D aims at producing the generic model of a dynamic process used by an expert when resorting to her or his domain knowledge in order to solve a problem; that is to say, producing the model built in the expert's mind. In particular, our proposal makes use of CommonKADS, introduced in Section 3.1, as interpretation framework when employing the template of domain scheme of the corresponding Knowledge Model.

A TOM4D model can be used to validate domain knowledge described by a CommonKADS conceptual model or to help in its construction. Inversely, the use of domain knowledge specified through a CommonKADS model will accelerate the construction of a TOM4D model of the corresponding process. Thus, CommonKADS templates constitute an important tool to control the introduction of semantics in the TOM4D knowledge interpretation phase. Nevertheless, due to the fact that these templates are domain independent, they are not sufficient to provide a correct physical interpretation of a set of variables. In order to complement the interpretative framework, TOM4D resorts then to the Tetrahedron of States (ToS) introduced in Section 3.2.

The major interest of the ToS is its applicability to any physical domains (electromagnetism, fluid dynamics, thermodynamics, etc.) of the classical Newtonian physics. Thus, any variable $x_i \in X$ of the perception model $PM(X(t))$ can receive a physical dimension and consequently, any proposition about the relations between two variables (e.g., hydraulic variables) can be analysed according to the corresponding ToS (e.g., the hydraulic ToS). Therefore, a proposition which does not satisfy one of the five relations of the mentioned tetrahedron must be rejected by the knowledge engineer. Hence, the ToS constitutes a powerful tool in order to interpret and represent experts' knowledge; that is to say, in order to control the introduction of the semantic content within the TOM4D modelling process.

Formal Logic, and more precisely the first order predicate calculus, is also used by TOM4D, complementing the ToS, in order to analyse logical consequences of a validated proposition through the corresponding tetrahedron. A new proposition can satisfy the semantic constraints of the ToS; however, it can be contradictory with the current version of the model. In this case, the knowledge engineer must solve the contradiction either rejecting the new proposition or updating adequately the current version of the model so that the model contains the new proposition. It is noteworthy that using Formal Logic allows employing Reiter's Theory of Diagnosis [33] in order to analyse the model coherence.

In conclusion, the combination of CommonKADS templates, the ToS and the first order predicate calculus constitute a powerful framework which allows controlling the semantic content introduced during the modelling process of our methodology.

7. TOM4D modelling process: application to a didactic example

In this section, the proposed modelling approach is described by means of a case study about the diagnosis of problems with a car, which has been taken from the book by Schreiber et al. [2] where it is presented by the authors in order to describe concepts and components of a CommonKADS Knowledge Model.

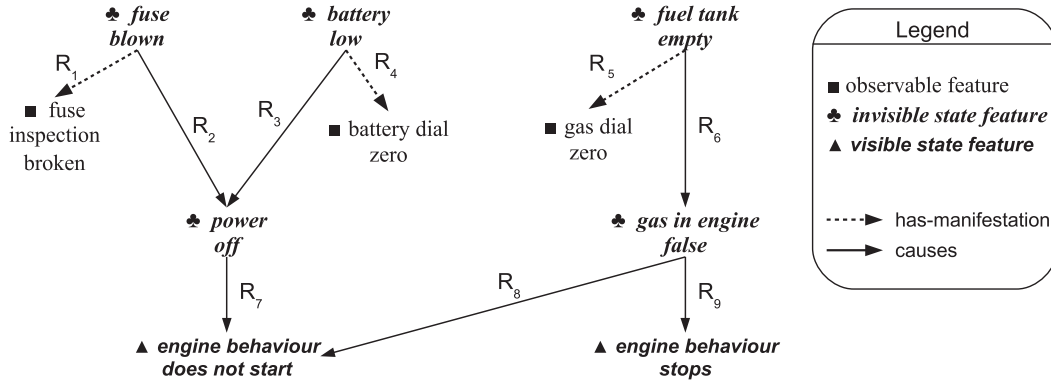


Fig. 10. Classification and organisation of knowledge pieces.

7.1. Knowledge interpretation

Fig. 10 illustrates nine rules of domain knowledge provided by an expert; e.g., the rule (R_1) establishes that *if fuse is blown, fuse inspection is broken*. From the analysis carried out through CommonKADS, the knowledge pieces are classified as observable features, and as visible and invisible aspects of an internal state. In addition, two types of relations between the knowledge elements are established as the figure depicts. More details about the mentioned analysis and the rule specification in Conceptual Modelling Language [2] can be found in Appendix A.1.

Considering the interpretation carried out through CommonKADS and the TOM4D principles introduced in Section 6, the next objective is to define a scenario model $M(\Omega) = \langle SM(\Omega), FM(\Omega), BM(\Omega) \rangle$ from the given knowledge and a set Ω of sequences of timed observations (see Definition 2) which describe certain modes of functioning of the car. In a real case, it would be desirable to have a set of timed observations describing the evolution over time of the process under study. In this case, Ω has not been provided; nevertheless, basing ourselves on deductions about the existent domain knowledge, we shall assume a scenario Ω .

From interpreting the available knowledge, the concepts *fuse*, *battery*, *fuel-tank*, *battery-dial* and *gas-dial* are considered components of the system. However, the concepts *fuse-inspection*, *power*, *gas-in-engine* and *engine-behaviour* denote physical entities which are unknown or whose information is insufficient. Consequently, abstract components (or component aggregates) such as *tools that allow fuse inspection*, *electric supply*, *gas supply* and *engine* will be defined to represent these concepts. In addition, the knowledge interpretation from CommonKADS enables to identify the variables of the system such as *fuse.status*, *gas-dial.value*, and *engine-behaviour.status* (cf. rules in Expr. (A.1)). Thus, these variables and components are defined in Expr. (3).

Variables $X = \{x_1, \dots, x_9\}$

$x_1 \triangleq \text{fuse.status}$
 $x_2 \triangleq \text{battery.status}$
 $x_3 \triangleq \text{fuel-tank.status}$
 $x_4 \triangleq \text{fuse-inspection.value}$
 $x_5 \triangleq \text{battery-dial.value}$
 $x_6 \triangleq \text{gas-dial.value}$
 $x_7 \triangleq \text{power.status}$
 $x_8 \triangleq \text{gas-in-engine.status}$
 $x_9 \triangleq \text{engine-behaviour.status}$

Components $COMPS = \{c_1, \dots, c_9\}$

$c_1 \triangleq \text{fuse}$
 $c_2 \triangleq \text{battery}$
 $c_3 \triangleq \text{fuel-tank}$
 $c_4 \triangleq \text{fuse inspection tools}$
 $c_5 \triangleq \text{battery-dial}$
 $c_6 \triangleq \text{gas-dial}$
 $c_7 \triangleq \text{electric supply}$
 $c_8 \triangleq \text{gas supply}$
 $c_9 \triangleq \text{engine}$

(3)

The rules are causal relations which implicitly define the notion of timed sequence of events; thus, from these rules, a set of sequences of timed observations can be assumed. Taking into consideration (R_1) and (R_2), if the *fuse* blows at the instant t_0 , the *fuse inspection* will result equal to broken at a subsequent moment $t_0 + \Delta t_i$ and the *electric supply* will be off at another moment $t_0 + \Delta t_j$. Affirming the order of sequence between $t_0 + \Delta t_i$ and $t_0 + \Delta t_j$ is not possible from the available information; nevertheless, we assume that all sensors properly work and quickly react, therefore, the order $t_0 + \Delta t_i < t_0 + \Delta t_j$ will be considered. In other words, first the *fuse* blows, then the *fuse inspection* result is equal to broken and, after that, the *electric supply* is switched off. Analogously, other two assumptions are: first the level of *battery* falls below the minimum, then the *battery-dial* is equal to zero and later the *electric supply* is turned off; and besides, first the *fuel-tank* is empty, then the *gas-dial* is equal to zero and after that the *gas supply* is empty.

Thus, considering the previous assumptions, a scenario Ω of timed observations is supposed such that

$$\begin{aligned}\Omega &= \{\omega_1, \omega_2, \omega_3, \omega_4\} \text{ where} \\ \omega_1 &= ((\text{blown}, t_{10}), (\text{broken}, t_{10} + \Delta t_{11}), (\text{off}, t_{10} + \Delta t_{11} + \Delta t_{12}), (\text{does_not_start}, t_{10} + \Delta t_{11} + \Delta t_{12} + \Delta t_{13})) \\ \omega_2 &= ((\text{low}, t_{20}), (\text{battery_zero}, t_{20} + \Delta t_{21}), (\text{off}, t_{20} + \Delta t_{21} + \Delta t_{22}), (\text{does_not_start}, t_{20} + \Delta t_{21} + \Delta t_{22} + \Delta t_{23})) \\ \omega_3 &= ((\text{empty}, t_{30}), (\text{gas_zero}, t_{30} + \Delta t_{31}), (\text{false}, t_{30} + \Delta t_{31} + \Delta t_{32}), (\text{does_not_start}, t_{30} + \Delta t_{31} + \Delta t_{32} + \Delta t_{33})) \\ \omega_4 &= ((\text{empty}, t_{40}), (\text{gas_zero}, t_{30} + \Delta t_{31}), (x_8, \text{false}, t_{40} + \Delta t_{41} + \Delta t_{42}), (\text{stops}, t_{40} + \Delta t_{41} + \Delta t_{42} + \Delta t_{43}))\end{aligned}\quad (4)$$

7.1.1. Structural model

A TOM4D structural model specifies the components of a system, the interconnections between them and the link of each system variable to a component. The components are related each other by means of input and output ports and each variable is associated with an output port of a component.

Definition 12. Let Ξ be an observed process or a particular scenario of this one, which governs the dynamic of a system, and let $X = \{x_i\}_{i=1\dots n}$ be a set of discrete variables associated with Ξ . Besides, let $\mathbb{N} \cup \{\lambda\}$ be a set of indices where λ is an empty string or null index and; let $in_i(Z)$ and $out_j(Z)$, $i, j \in \mathbb{N} \cup \{\lambda\}$ be predicates denoting the values of an input port i and an output port j of a component Z , respectively. In particular $i = \lambda$ or $j = \lambda$ if the input or output port is unique. Then, a structural model SM of Ξ is a tuple $SM(\Xi) = \langle COMPS, R_{port}, R_{xport} \rangle$ such that

- $COMPS$ is a finite set of constants identifying the system components,
- R_{port} is a set of equality predicates defining the interconnections between the components, where

$$R_{port} = \{out_i(Z) = in_j(Q) \mid i, j \in \mathbb{N} \cup \{\lambda\}, Z, Q \in COMPS, Z \neq Q\},$$

- R_{xport} is a set of equality predicates linking each variable $x \in X$ to only one output port of a component, then

$$\begin{aligned}R_{xport} &= \{out_i(Z) = x \mid i \in \mathbb{N} \cup \{\lambda\}, Z \in COMPS, x \in X\} \wedge \\ \forall (out_i(Z) = x) \in R_{xport}, ((out_j(Q) = x) \in R_{xport} \Rightarrow Q = Z \wedge j = i).\end{aligned}$$

From interpreting the rules provided by the expert and according to Definition 12, a first structural model $SM(\Omega)$ of the given scenario can be described. The link between a component and a variable is determined from the descriptions in Expr. (3), and the rules in the figure establish relations between the system variables; therefore, the relations between the components are implicitly defined. For example, the rule (R_1) specified as `fuse.status = blown HAS-MANIFESTATION fuse-inspection.value = broken` (see Expr. (A.1) in Appendix A.1) determines an implicit relation between `fuse` and `fuse inspection tools`; that is to say, between the components c_1 and c_4 designated in Expr. (3) whose linked variables are x_1 and x_4 , respectively. Thus, $out(c_1) = x_1$ and $out(c_4) = x_4$ will belong to R_{xport} in our specification. From (R_1) we can consider that $out(c_1)$ will affect in a some way the output value $out(c_4)$; hence, $out(c_1)$ will be connected to an input of c_4 ; that is to say, $out(c_1) = in(c_4)$ will belong to R_{port} . In a similar way, all rules can be interpreted and, consequently, the different relations between the components can be established. Thus, considering the previous

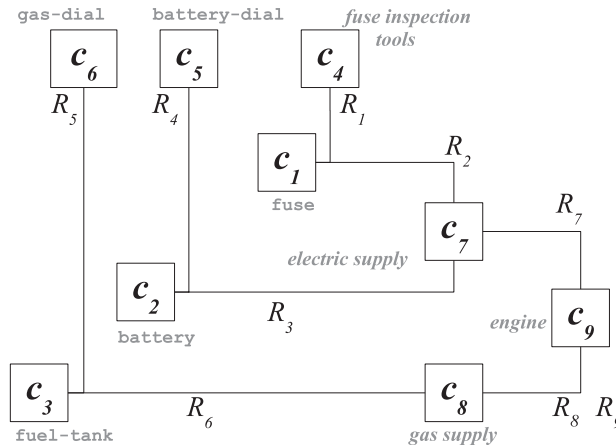


Fig. 11. Structural model $SM(\Omega)$.

analysis, the structural model $SM(\Omega)$, illustrated in Fig. 11, is defined (Expr. (A.2) in Appendix A.2 presents the $SM(\Omega)$ specification from Definition 12).

7.1.2. Functional model

A functional model, for its part, describes the relations among the values that can be assumed by the variables, establishing thus a logical implication or relation between them. This model specifies the domain values of each variable and how these ones relate each other; that is, how values of a variable are defined depending on values of other variables.

Definition 13. Let Ξ be an observed process or a particular scenario of this one, which governs the dynamic of a system, and let $X = \{x_i\}_{i=1\dots n}$ be a set of discrete variables associated with Ξ . A functional model FM of Ξ is a tuple $FM(\Xi) = \langle \Delta, F, R_f \rangle$ where

- $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ with Δ_{x_i} the set of values which can be assumed by $x_i \in X$, and $\phi_i \in \Delta_{x_i}$ an unknown value of x_i ,
- $F = \{f : \Delta_{x_1} \times \dots \times \Delta_{x_m} \rightarrow \Delta_{x_r} \mid \{x_1, \dots, x_m, x_r\} \subseteq X\}$ is a set of functions and
- R_f is a set of equality predicates defining a variable as function of the other ones:

$$R_f = \{x_r = f(x_1, \dots, x_m) \mid \{x_r, x_1, \dots, x_m\} \subseteq X, f \in F\}.$$

For the purpose of specifying $FM(\Omega)$, the values which can be assumed by $x_1, \dots, x_9 \in X$ will be determined by defining the sets $\Delta_{x_1}, \dots, \Delta_{x_9}$. According to (R_8) , (R_9) and the scenario in Expr. (4), the variable x_9 assumes the values *stops* and *does_not_start*. However, for $x_i \in X$ with $i = 1, \dots, 8$ only one value is known; therefore, by assuming that these ones are variables, one symbol ϕ_i will be added to each set Δ_{x_i} ($i = 1, \dots, 8$), denoting an unknown value that each x_i could take. In this way, $\Delta_{x_1}, \dots, \Delta_{x_9}$ are specified as follows:

$$\begin{array}{lll} \Delta_{x_1} = \{\text{blown}, \phi_1\} & \Delta_{x_4} = \{\text{broken}, \phi_4\} & \Delta_{x_7} = \{\text{off}, \phi_7\} \\ \Delta_{x_2} = \{\text{low}, \phi_2\} & \Delta_{x_5} = \{\text{battery_zero}, \phi_5\} & \Delta_{x_7} = \{\text{off}, \phi_7\} \\ \Delta_{x_3} = \{\text{empty}, \phi_3\} & \Delta_{x_6} = \{\text{gas_zero}, \phi_6\} & \Delta_{x_9} = \{\text{stops}, \text{does_not_start}\} \end{array} \quad (5)$$

In order to define F and R_f , the rules must be analysed and then the relations among values established. For example, from

(R_2) fuse.status = blown CAUSES power.status = off;

(R_3) battery.status = low CAUSES power.status = off;

or, what is the same thing,

(R_2) $x_1 = \text{blown}$ CAUSES $x_7 = \text{off}$;

(R_3) $x_2 = \text{low}$ CAUSES $x_7 = \text{off}$;

we can consider that the value $x_7 = \text{off}$ is conditioned by $x_1 = \text{blown}$ or $x_2 = \text{low}$. Therefore, in order to specify these relations a function $f_4 : \Delta_{x_1} \times \Delta_{x_2} \rightarrow \Delta_{x_7}$ is considered, which will be a member of F and whose definition is described in Expr. (6). Consequently, the conditioning factor of the values assumed by x_7 can be expressed as $x_7 = f_4(x_1, x_2)$, where the mentioned equality will belong to R_f .

$f_4 : \Delta_{x_1} \times \Delta_{x_2} \rightarrow \Delta_{x_7}$		
y_1	y_2	$f_4(y_1, y_2)$
blown	low	off
ϕ_1	low	off
blown	ϕ_2	off
ϕ_1	ϕ_2	ϕ_7

(6)

The previous analysis can be repeated with all rules until obtaining a set of functions relating the values that can be assumed by the variables. Nevertheless, there exists a case which presents some ambiguity.

According to the rules

(R_7) power.status = off CAUSES engine-behaviour.status = does-not-start;

(R_8) gas-in-engine.status = false CAUSES engine-behaviour.status = does-not-start;

(R_9) gas-in-engine.status = false CAUSES engine-behaviour.status = stops;

or

(R_7) $x_7 = \text{off}$ CAUSES $x_9 = \text{does_not_start}$;

(R_8) $x_8 = \text{false}$ CAUSES $x_9 = \text{does_not_start}$;

(R_9) $x_8 = \text{false}$ CAUSES $x_9 = \text{stops}$;

the value of x_9 is conditioned by the values of the variables x_7 and x_8 ; hence, a function $f_6 : \Delta_{x_7} \times \Delta_{x_8} \rightarrow \Delta_{x_9}$ will exist. However, defining f_6 presents two situations to deal. On one hand, since there are at least two known values associated to x_9 , an unknown value

symbol ϕ_9 would not be a priori necessary; but, the need of using ϕ_9 arises when this function must associate two unknown input values ϕ_7 and ϕ_8 with another unknown output value, as shown in Expr. (7). On the other hand, defining the function f_6 from the rules (R_7) , (R_8) and (R_9) presents a problem: for a same combination of values in $\Delta_{x_7} \times \Delta_{x_8}$, there are two different values in Δ_{x_9} . The rules (R_8) and (R_9) establish that when x_8 is *false*, x_9 is *does_not_start* and *stops*. This results in the definition shown in Expr. (7), where f_6 is not a function but a relation.

$f_6 : \Delta_{x_7} \times \Delta_{x_8} \rightarrow \Delta_{x_9}$		
y_1	y_2	$f_6(y_1, y_2)$
<i>off</i>	<i>false</i>	<i>does-not-start</i>
<i>off</i>	<i>false</i>	<i>stops</i>
<i>off</i>	ϕ_8	<i>does-not-start</i>
ϕ_7	<i>false</i>	<i>does-not-start</i>
ϕ_7	<i>false</i>	<i>stops</i>
ϕ_7	ϕ_8	ϕ_9

(7)

This situation should not be accepted because fuzzy knowledge is present. In general, the rules define variables whose values are described as adjectives (for example, “battery is low”); however, in the rules (R_7) , (R_8) and (R_9) the values of *engine behaviour* are described as verbs (for example, “the engine behaviour stops”). This knowledge provided by experts is implicitly related to states of the system; for example, the engine behaviour stops if it was running previously and runs out of gas, as Fig. 12a illustrates. Here *broken down* is a state in which the engine is stopped and does not work. In a similar way, as Fig. 12b depicts, the engine does not start if it is stopped, has run out of gas and a key is turned.

This circumstance should lead to consult to experts in order to distinguish the knowledge concerning functional aspects from that one concerning behavioural aspects; and thus, to define the function f_6 . Due to the fact that there is not an expert, an assumption will be established: the values *stops* and *does_not_start* are not part of the functional model. Instead, \neg works will be assumed as value of this variable and the rules R_7 , R_8 and R_9 will be rewritten as:

$(R_7) x_7 = \text{off CAUSES } \neg \text{works};$

$(R_8) x_8 = \text{false CAUSES } \neg \text{works};$

$(R_9) x_8 = \text{false CAUSES } \neg \text{works};.$

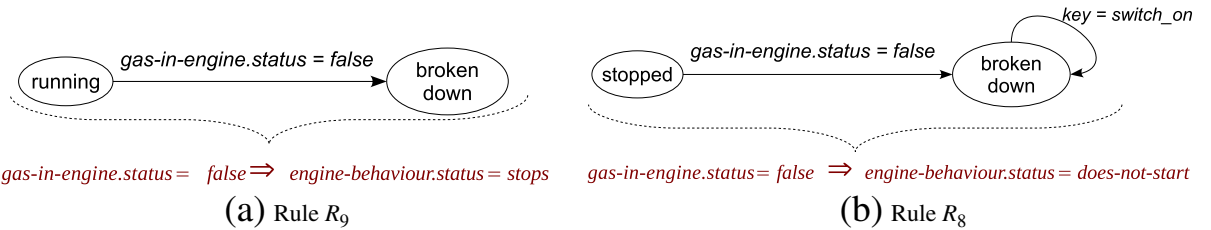


Fig. 12. Interpretation of rules R_8 and R_9 .

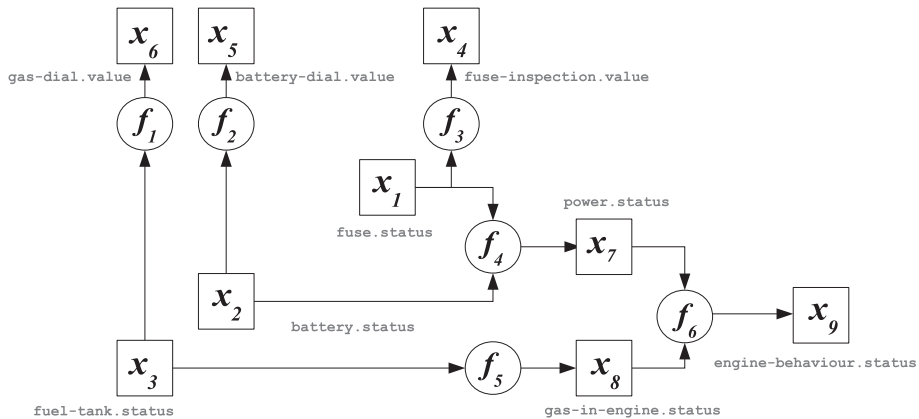


Fig. 13. Functional model $FM(\Omega)$.

Therefore, from this hypothesis, Δ_{x_9} in Expr. (5) and f_6 in Expr. (7) have to be redefined. Thus, the possible values that x_9 can take are values belonging to $\Delta_{x_9} = \{\neg works, \phi_9\}$ and f_6 is redefined in Expr. (8).

$f_6 : \Delta_{x_7} \times \Delta_{x_8} \rightarrow \Delta_{x_9}$		
y_1	y_2	$f_6(y_1, y_2)$
<i>off</i>	<i>false</i>	$\neg works$
<i>off</i>	ϕ_8	$\neg works$
ϕ_7	<i>false</i>	$\neg works$
ϕ_7	ϕ_8	ϕ_9

(8)

Following the previous reasoning, all functions can be deduced from the rules. These functions define the scenario functional model $FM(\Omega)$ whose graphical representation is illustrated in Fig. 13 (Expr. (A.3) in Appendix A.2 specifies $FM(\Omega)$ from Definition 13)

7.1.3. Behavioural model

A TOM4D behavioural model describes the possible sequences of observation classes that can occur; and therefore, the discernible states between them. Thus, as follows from Section 6, a change of state is determined for a *timed observation*, or what is the same, an *occurrence of a class observation*; that is to say, when a variable assumes a new value.

Definition 14. Let Ξ be an observed process or a particular scenario of this one, which governs the dynamic of a system, let $X = \{x_i\}_{i=1 \dots n}$ be a set of discrete variables associated with Ξ , and let $\Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ be such that Δ_{x_i} is the set of values which can be assumed by $x_i \in X$. A behavioural model BM of Ξ is a tuple $BM(\Xi) = \langle S, C, \gamma \rangle$ such that

- $S = \{s : X \rightarrow \Delta \mid s(x_i) = \delta, x_i \in X, \delta \in \Delta_{x_i}\}$ is a set of functions which characterise the system states,
- C is a set of observation classes, where an *observation class* associated with a variable $x_i \in X$ is a set $C_i = \{(x_i, \delta) \mid \delta \in \Delta_{x_i}\}$ and
- $\gamma : S \times C \rightarrow S$ is a function of state transition.

From these definitions, a state is characterised by a function $s \in S$ which returns the values assumed by the variables in the state. For example, a state characterised by the battery value *low* is specified as $s_m : X \rightarrow \Delta \mid s_m(x_2) = low$.

In this scenario, there are nine known binary variables; consequently, there exist $2^9 = 512$ possible combinations of values which will determine 512 discernible states, as Table 1 illustrates. In this table, the columns identify the variables and the rows identify the discernible states; thus, an element a_{ij} in the table is $a_{ij} = s_i(x_j)$. For example, $s_1(x_5) = battery_zero$ meaning that s_1 denotes a state in which the battery dial (x_5) shows *battery_zero*. Therefore, Table 1 illustrates the set S which will be part of the model $BM(\Omega)$ of this scenario.

For simplicity and convenience to define C , we shall consider each observation class as a singleton; that is, $C_{i,j} = \{(x_i, \delta_j)\}$, $x_i \in X, \delta_j \in \Delta_{x_i}$. From Δ the observation classes can be established. For example, $\Delta_{x_5} = \{battery_zero, \phi_5\}$ allows assuming two observation classes (or types of events): $C_{5,1} = \{(x_5, battery_zero)\}$ and $C_{5,2} = \{(x_5, \phi_5)\}$. Thus, the timed observation (*battery_zero*, $t_{20} + \Delta t_{21}$) in the scenario Ω (see Expr. (4)), is an occurrence of the observation class $C_{5,1}$ at time $t_{20} + \Delta t_{21}$. In this way, the different observation classes are specified as follows.

$$\begin{array}{lll}
 C_{1,1} = \{(x_1, blown)\} & C_{4,1} = \{(x_4, broken)\} & C_{7,1} = \{(x_7, off)\} \\
 C_{1,2} = \{(x_1, \phi_1)\} & C_{4,2} = \{(x_4, \phi_4)\} & C_{7,2} = \{(x_7, \phi_7)\} \\
 C_{2,1} = \{(x_2, low)\} & C_{5,1} = \{(x_5, \phi_5)\} & C_{8,1} = \{(x_8, false)\} \\
 C_{2,2} = \{(x_2, \phi_2)\} & C_{5,2} = \{(x_5, \phi_5)\} & C_{8,2} = \{(x_8, \phi_8)\} \\
 C_{3,1} = \{(x_3, empty)\} & C_{6,1} = \{(x_6, gas_zero)\} & C_{9,1} = \{(x_9, \neg works)\} \\
 C_{3,2} = \{(x_3, \phi_3)\} & C_{6,2} = \{(x_6, \phi_6)\} & C_{9,2} = \{(x_9, \phi_9)\}
 \end{array} \tag{9}$$

By considering these observation classes, the sequences of timed observations $\omega_1, \omega_2, \omega_3$ and ω_4 , which define Ω , determine sequences of observation classes. We recall that the x_9 values *does_not_start* and *stops* were interpreted as $\neg works$; therefore, the observations of the form (*does_not_start*, t_k) and (*stops*, t_k) are occurrences of the observation class $C_{9,1}$. For example, $\omega_1 = ((blown, t_{10}),$

Table 1
Discernible states of Ω .

$S = \{s : X \rightarrow \Delta \mid s(x_i) = \delta, x_i \in X, \delta \in \Delta_{x_i}\}$									
$s_i \in S$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
s_1	<i>blown</i>	<i>low</i>	<i>empty</i>	<i>broken</i>	<i>battery_zero</i>	<i>gas_zero</i>	<i>off</i>	<i>false</i>	<i>works</i>
s_2	ϕ_1	<i>low</i>	<i>empty</i>	<i>broken</i>	<i>battery_zero</i>	<i>gas_zero</i>	<i>off</i>	<i>false</i>	<i>works</i>
s_3	<i>blown</i>	ϕ_2	<i>empty</i>	<i>broken</i>	<i>battery_zero</i>	<i>gas_zero</i>	<i>off</i>	<i>false</i>	<i>works</i>
...
s_{512}	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9

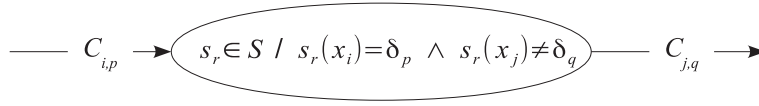


Fig. 14. Characterisation of states between two observation classes.

(*broken*, $t_{10} + \Delta t_{11}$), (*off*, $t_{10} + \Delta t_{11} + \Delta t_{12}$), (*does_not_start*, $t_{10} + \Delta t_{11} + \Delta t_{12} + \Delta t_{13}$)) indicates that the sequence of observation classes $C_{1,1}$, $C_{4,1}$, $C_{7,1}$, $C_{9,1}$ can occur (Fig. A.28 in Appendix A.6 shows the sequences established from Ω).

In order to propose a behavioural model of Ω , the states in which these observation classes can take place, must be identified. For this purpose, the following definitions are presented.

Definition 15. Let C_i be an observation class and let $s \in S$ be a discernible state, we say that

- an occurrence of C_i is *possible* in s if $\exists (x, \delta) \in C_i \mid s(x) \neq \delta$ and,
- an occurrence of C_i *has happened* in s if $\exists (x, \delta) \in C_i \mid s(x) = \delta$.

Definition 16. Let C_i, C_j be two observation classes. We say that $s \in S$ is a *possible state* between C_i and C_j if an occurrence of C_i has happened in s and an occurrence of C_j is possible in s . That is to say, $s \in S$ is possible between C_i and C_j if $\exists (x_i, \delta_i) \in C_i, (x_j, \delta_j) \in C_j \mid s(x) = \delta_i \wedge s(x_j) \neq \delta_j$.

Fig. 14 illustrates the general sense of Definition 16.

Consequently, given two observation classes, it is possible to characterise the discernible states between them. Thus, the sequences of observation classes determined by Ω , can occur through states $s_a, s_b, s_c, s_d, s_e, s_f, s_g, s_h, s_i, s_m, s_n, s_o \in S$, as Fig. 15 presents. These states are part of the behavioural model of the scenario Ω and are such that:

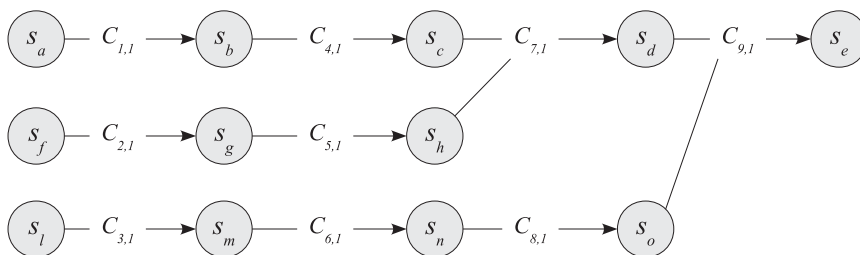
$$\begin{array}{ll}
 s_a \in S \mid s_a(x_1) \neq \text{blown} & s_g \in S \mid s_g(x_2) = \text{low} \wedge s_g(x_5) \neq \text{battery_zero} \\
 s_b \in S \mid s_b(x_1) = \text{blown} \wedge s_b(x_4) \neq \text{broken} & s_h \in S \mid s_h(x_5) = \text{battery_zero} \wedge s_h(x_7) \neq \text{off} \\
 s_c \in S \mid s_c(x_4) = \text{broken} \wedge s_b(x_7) \neq \text{off} & s_i \in S \mid s_i(x_3) \neq \text{empty} \\
 s_d \in S \mid s_c(x_7) = \text{off} \wedge s_b(x_9) \neq \neg \text{works} & s_m \in S \mid s_m(x_3) = \text{empty} \wedge s_m(x_6) \neq \text{gas_zero} \\
 s_e \in S \mid s_e(x_9) = \neg \text{works} & s_n \in S \mid s_n(x_6) = \text{gas_zero} \wedge s_n(x_8) \neq \text{false} \\
 s_f \in S \mid s_f(x_2) \neq \text{low} & s_o \in S \mid s_o(x_8) = \text{false} \wedge s_o(x_9) \neq \neg \text{works}
 \end{array} \quad (10)$$

Each one of these discernible states can represent one or more states that verify the conditions defined. That is, s_a could be both s_2 and s_{512} in Table 1 since both states verify $s_a(x_1) \neq \text{blown}$. Indeed, $s \in S$ also defines types of states since the last ones are characterised only by those variables which can be observed.

Therefore, from Expr. (10) the state transitions of $BM(\Omega)$ (see Fig. 15) can be defined. For example, a state transition from s_a to s_b through an occurrence of $C_{1,1}$ is specified by $\gamma(s_a, C_{1,1}) = s_b$ with $s_a(x_1) \neq \text{blown} \wedge s_b(x_1) = \text{blown} \wedge s_b(x_4) \neq \text{broken}$.

Accordingly, this analysis along with Table 1 and Expr. (10) lead to the specification of the behavioural model $BM(\Omega)$ according to Definition 14 (the $BM(\Omega)$ definition is presented in Expr. (A.4), Appendix A.2).

Finally, from the previous reasoning, the scenario model $M(\Omega) = \langle SM(\Omega), FM(\Omega), BM(\Omega) \rangle$ is defined. However, this model, which describes the available knowledge, is inadequate for analysing or diagnosing behaviour problems. It should be noticed that the existence of only 9 binary components determines 512 states, a number significant with respect to the small number of units. Presumably, certain defined states are irrelevant for the pursued objectives or, they are meaningless since they are impossible physically. Then, the two following stages aim at dealing with these aspects.

Fig. 15. Behavioural model $BM(\Omega)$.

7.2. Process definition

The result of the first phase was a scenario model made up of three models, which are a representation of the knowledge given by experts. The objective of this new phase is to obtain a *perception model* that defines the boundaries and operating constraints of the studied process.

Definition 17. Let Ξ be an observed process or a particular scenario of this one, which governs the dynamic of a system. A perception model PM of Ξ is a tuple $PM(\Xi) = \langle X, \Psi, R_q \rangle$ where

- $X = \{x_i\}_{i=1 \dots n}$ a set of discrete variables associated with Ξ ,
- Ψ is a finite set of constant values, typically corresponding to thresholds of the variables and
- $R_q = R_{goal} \cup R_n \cup R_{ab}$ is a set of predicates that relate elements of X with elements of Ψ in order to determine constraints on variables. R_{goal} is a set that describes the operating goals and, R_n and R_{ab} are sets that specify the normal and abnormal operating modes, respectively.

Since, in this example, the dynamic process to be modelled is described by means of the binary variables, formal logic will be used to give an interpretation and a more abstract representation of the scenario. Particularly, we shall aim to specify the system as a set of logical gates, which will allow visualising more clearly goals and operating modes, and to reason on these ones.

First, the variables of the process will be considered logic; that is to say, they will take values *true* (1) or *false* (0). Therefore, x_1, \dots, x_9 will be substituted for the binary variables $\bar{x}_1, \dots, \bar{x}_9$, respectively; and each value $\delta_i \in \Delta_{x_i}$ will be represented by a truth value belonging to a set $\Delta_{\bar{x}_i}$. Consequently, for each Δ_{x_i} its analogous $\Delta_{\bar{x}_i}$ is established; for example, from $\Delta_{x_1} = \{blown, \phi_1\}$, $\Delta_{\bar{x}_1} = \{0, 1\}$ is defined. In a similar way, for each $f_i \in F$ of the previous functional model, a binary function f_{B_i} exists. That is to say, for f_4 defined in Expr. (6), the function f_{B4} established in Expr. (11) is defined, and so on.

$f_{B4} : \Delta_{\bar{x}_1} \times \Delta_{\bar{x}_2} \rightarrow \Delta_{\bar{x}_7}$		
y_1	y_2	$f_{B4}(y_1, y_2)$
0	0	0
1	0	0
0	1	0
1	1	1

(11)

The definitions of f_{B4} and f_{B6} correspond to the logical function *and*, and the definitions f_{B1}, f_{B2}, f_{B3} , and f_{B5} can be written as logical functions *and* and *or* (see Appendix A.4). That is, for example, $f_{B4}(y_1, y_2) = \text{and}(y_1, y_2)$, and $f_{B1}(y) = \text{and}(y, y)$ or $f_{B1}(y) = (0, y)$. For simplicity, the definitions of $f_{B1}(y), f_{B2}(y), f_{B3}(y), f_{B5}(y)$ will be considered equal to $\text{and}(y, y)$. Accordingly, $f_{B1}, f_{B2}, f_{B3}, f_{B4}, f_{B5}, f_{B6}$ are functions *and* which allow establishing the relations among the values that can be taken by the variables; thus, Expr. (12) defines the dependencies among these last ones.

$$\begin{array}{lll} \bar{x}_6 = f_{B1}(\bar{x}_3) & \bar{x}_4 = f_{B3}(\bar{x}_1) & \bar{x}_8 = f_{B5}(\bar{x}_3) \\ \bar{x}_5 = f_{B2}(\bar{x}_2) & \bar{x}_7 = f_{B4}(\bar{x}_1, \bar{x}_2) & \bar{x}_9 = f_{B6}(\bar{x}_7, \bar{x}_8) \end{array} \quad (12)$$

Each one of the logical variables $\bar{x}_1, \dots, \bar{x}_9$ must be associated with the output of a logical component c_{B1}, \dots, c_{B9} ; thus, a structural model of logical gates can be inferred from the definitions of $f_{B1}, f_{B2}, f_{B3}, f_{B4}, f_{B5}, f_{B6}$ and Expr. (12). For example, $f_{B4}(y_1, y_2) = \text{and}(y_1, y_2)$ and $\bar{x}_7 = f_{B4}(\bar{x}_1, \bar{x}_2)$, then $\bar{x}_7 = \text{and}(\bar{x}_1, \bar{x}_2)$; hence, \bar{x}_7 is associated with the output of a component c_{B7} , which is a logical gate *AND*. The inputs of this gate are the outputs of the components c_{B1} and c_{B2} associated with \bar{x}_1 and \bar{x}_2 , respectively. In particular, the previously defined relations indicate that the values of \bar{x}_1 and \bar{x}_2 are not conditioned by values of other variables; then, the components c_{B1} and c_{B2} are not logical gates but independent logical components. Thus, the same reasoning can be followed in order to define the structural model.

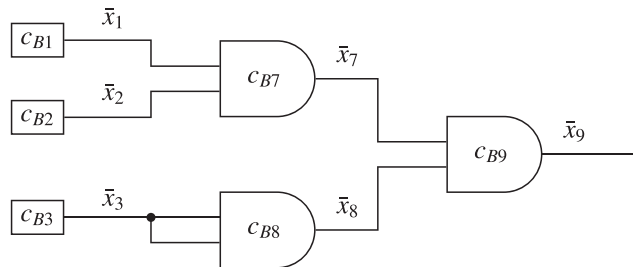


Fig. 16. Logical model of the process.

In particular, the components c_{B4} , c_{B5} , c_{B6} simply replicates the outputs of c_{B1} , c_{B2} , c_{B3} ; e.g., c_{B4} has associated as output $\bar{x}_4 = f_{B3}(\bar{x}_1)$ (or $\bar{x}_4 = \text{and}(\bar{x}_1, \bar{x}_1)$) then this component only replicates the value of \bar{x}_1 provided by c_{B1} . Therefore, for reducing complexity, the behaviour of c_{B4} , c_{B5} , c_{B6} will be assumed as correct. Consequently, the process model will be limited to the variables $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_7, \bar{x}_8, \bar{x}_9$ and the components c_{B1} , c_{B2} , c_{B3} , c_{B7} , c_{B8} , c_{B9} ; where the structural model is that one shown in Fig. 16.

This logical representation allows a description with the first order predicate logic where the Reiter's diagnosis theory [33] can be applied. Considering that the unary predicate $AB(.)$ means *abnormal behaviour*, the description of the system under this theory consists of a set of components $\{c_{B1}, c_{B2}, c_{B3}, c_{B7}, c_{B8}, c_{B9}\}$ whose normal behaviour, type and connections are described by logic sentences (Exprs. (A.12), (A.13) and (A.14) in Appendix A.5).

The logical description allows determining more clearly operating goals and normal and abnormal behaviours, which are required in the perception model. As the equalities in Expr. (13) describe, each variable is associated with the output of a component; hence, operating goals and modes will be defined in terms of these variables.

$$\begin{array}{lll} \text{out}(c_{B1}) = \bar{x}_1 & \text{out}(c_{B3}) = \bar{x}_3 & \text{out}(c_{B8}) = \bar{x}_8 \\ \text{out}(c_{B2}) = \bar{x}_2 & \text{out}(c_{B7}) = \bar{x}_7 & \text{out}(c_{B9}) = \bar{x}_9 \end{array} \quad (13)$$

The perception model $PM(X(t))$ can be then defined from the logical representation where operating goals, normal behaviour and abnormal behaviour can be easily established. Clearly, the operating goal is $\bar{x}_9 = 1$; that is, *engine behaviour works*. R_n describes, like in Expr. (14), the normal behaviour conditions according to the Reiter' theory, which verify consistency conditions in the model (see Fig. 16). In turn, R_{ab} specifies, like in Expr. (15), the abnormal behaviour conditions; that is, logic inconsistency among values.

$$\begin{array}{l} R_n \text{ describes the conditions of the abnormal operating mode :} \\ (\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_3 = 1 \wedge \bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 1) \vee \\ (((\bar{x}_1 = 0 \vee \bar{x}_2 = 0) \wedge \bar{x}_7 = 0) \vee (\bar{x}_3 = 0 \wedge \bar{x}_8 = 0)) \wedge \bar{x}_9 = 0) \end{array} \quad (14)$$

$$\begin{array}{l} R_{ab} \text{ describes the conditions of the abnormal operating mode :} \\ (\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_7 = 0) \vee ((\bar{x}_1 = 0 \vee \bar{x}_2 = 0) \wedge \bar{x}_7 = 1) \vee (\bar{x}_3 = 0 \wedge \bar{x}_8 = 1) \vee \\ (\bar{x}_3 = 1 \wedge \bar{x}_8 = 0) \vee (\bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 0) \vee ((\bar{x}_7 = 0 \vee \bar{x}_8 = 0) \wedge \bar{x}_9 = 1) \end{array} \quad (15)$$

In this manner, from the logic interpretation of the variables and the perception model $PM(X(t))$, the process model boundaries are established. It is noteworthy that the number of variables has been reduced; and therefore, a new definition of the set S of states is considered, where the number of these ones has been reduced from 512 to $2^6 = 64$.

7.3. Generic modelling

The previous logical interpretation leads to build a logical model, according to the Reiter's theory of the consistency-based diagnosis [33], as a set of first order predicate formulas. Thus, from this theory, the perception model allows to discriminate between those value combinations that will characterise consistent (or normal behaviour) states, and those ones that do not (i.e.; abnormal behaviour). Nevertheless, the Reiter's theory tacitly assumes that logically *consistent* states correspond to *normal and desired* behaviour, and the *inconsistent* states, denoting a problem with at least one component, coincide with *abnormal and undesired* behaviour. The problem is that this correspondence sometimes is not compatible with a physical interpretation of the variables; thus, a logical model is a strong tool for reasoning but is not sufficient.

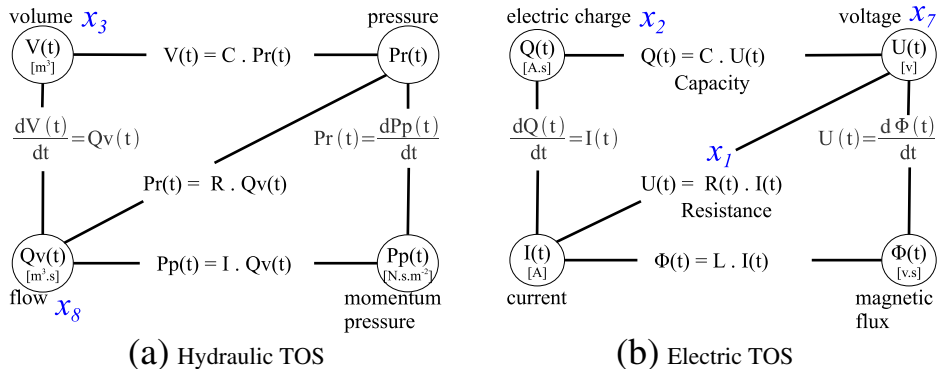


Fig. 17. Physical interpretation of variables.

For example, when observing Fig. 16, a state s_{64} such that $s_{64}(\bar{x}_i) = 1$ with $i = 1, 2, 3, 7, 8, 9$, is clearly a consistent state associated with normal and desired behaviour; that is, all components work and the engine works. However, s_{15} , such that $s_{15}(\bar{x}_i) = 0$ for $i = 1, 2, 7, 9$ and $s_{15}(x_i) = 1$ for $i = 3, 8$, is a consistent state of normal but undesired behaviour; that is, there is gas in engine ($s_{15}(\bar{x}_3) = 1, s_{15}(\bar{x}_8) = 1$) but power is off ($s_{15}(\bar{x}_1) = 0, s_{15}(\bar{x}_2) = 0, s_{15}(\bar{x}_7) = 0$) then engine does not work ($s_{15}(\bar{x}_9) = 0$). A state s_4 ($s_4(\bar{x}_3) = 1 \wedge s_4(\bar{x}_i) = 0$ for $i = 1, 2, 7, 8, 9$) in which the tank has fuel ($s_4(\bar{x}_3) = 1$) but there is not gas in engine ($s_4(\bar{x}_8) = 0$) is a inconsistent state which corresponds to a problem with the gas supply (component c_{88} does not work). In contrast, s_6 ($s_6(\bar{x}_8) = 1 \wedge s_6(\bar{x}_i) = 0$ for $i = 1, 2, 3, 7, 9$) is an inconsistent state, in which the fuel tank is empty ($s_6(\bar{x}_3) = 0$) and there is gas in the engine ($s_6(\bar{x}_8) = 1$). Therefore, s_6 cannot be associated with the problem of a component. On the contrary, this state is transient and corresponds to normal behaviour; however, in the logical model, it is identified as a state of abnormal behaviour.

These examples show that the logical interpretation of variables required by the Reiter's theory must be completed with a physical interpretation. For this purpose, [16] proposes to utilise the Tetrahedron of States (ToS), introduced in Section 3.2, where the given variables can be mapped into physical variables of ToS, thus establishing the relations among them. In this way, the introduction of semantic content in the physical interpretation of variables is controlled through the ToS framework. In particular, the ToS of hydraulic domain and that one of electric domain, shown respectively in Fig. 17a and b, will be used in this example.

Each given variable $x_i \in X$, which has already been matched with a logical variable \bar{x}_i , is mapped with a physical variable of the corresponding ToS. For example, using the Hydraulic ToS, the variable x_3 (fuel tank status) is associated with the gas volume $V(t)$ in the tank (see Fig. 17a). Thus, $x_3 = \text{empty}$, which is logically interpreted as $\bar{x}_3 = 0$ (false), is physically interpreted through ToS as $V(t) = 0$; and, $x_3 = \text{empty}$ (or $x_3 = \phi_3$), related to $\bar{x}_3 = 1$ (true), is physically interpreted as $V(t) \neq 0$. The variable x_8 (gas supply status), for its part, is associated with the gas flow $Qv(t)$ in the gas supply, so that $x_8 = \text{false}$, related to $\bar{x}_8 = 0$, is physically interpreted as $Qv(t) = 0$; and, $x_8 = \text{false}$ (or $x_8 = \phi_8$), related to $\bar{x}_8 = 1$, is interpreted as $Qv(t) \neq 0$. Similarly, the electric ToS allows the following associations (see Fig. 17b): x_2 (battery status) corresponds to the electric charge $Q(t)$ in the battery, x_1 (fuse status) is associated with the system resistance $R(t)$, x_7 (electric supply status) is linked to the voltage $U(t)$. For the purpose of interpreting the process behaviour and the corresponding states, we assume that the current $I(t)$, the voltage $U(t)$ and the resistance $R(t)$ are piecewise constant over time: $I(t) = i_c$ or $I(t) = 0$ (no current), $U(t) = u_c$ or $U(t) = 0$ (no voltage) and $R(t) = r_c$ or $R(t) = \infty$ (the fuse is blown). Thus, $x_1 = \text{blown}$ ($\bar{x}_1 = 0$) is considered as $R(t) = \infty$, and $x_1 = \text{blown}$ ($\bar{x}_1 = 1$) is interpreted as $R(t) = r_c$; and, in a similar way, $x_7 = \text{off}$ ($\bar{x}_7 = 0$) is understood as $U(t) = 0$ and in the opposite case $U(t) = u_c \neq 0$. Since $I(t) = \frac{dQ(t)}{dt}$, when $I(t)$ is zero, the electric charge of the battery is a constant; thus, $x_2 = \text{low}$ ($\bar{x}_2 = 0$) is interpreted as $Q(t) = q$ and otherwise $Q(t)$ evolves over time.

Consequently, the preceding associations result in the physical model illustrated in Fig. 18 where all variables of the process have been matched with the ToS physical variables, except x_9 associated with the engine behaviour. It is important to note that x_9 was defined from concepts of the domain knowledge which were fuzzy (see functional model, Section 7.1). In particular, we shall assume that an occurrence of the type (x_9 , works) can be observed when starting the engine; and then, $x_9 = \text{works}$ means that the engine is working. Besides, we shall suppose that $x_9 = \text{works}$ means that the engine is stopped, owing to the fact that it is off or because there is a problem. Thus, x_9 will be considered in the physical model as the result of the voltage $U(t)$ by the gas flow $Qv(t)$ by $\alpha \in \{0, 1\}$; where the last one models the car key (off/on). That is, x_9 is interpreted as a physical variable $x_9^p = \alpha \cdot U(t) \cdot Qv(t)$.

The physical interpretation through ToS allows providing semantic to each state and to identify the states useful for our purpose, discarding thus those ones that are meaningless. For example, because $Qv(t) = dV(t)/dt$, all the states where $\bar{x}_3 = 0$ and $\bar{x}_8 = 1$ correspond to a transient situation; that is, the fuel tank is empty ($V(t) = 0$) but there is still gas flow in the engine ($Qv(t) \neq 0$). Thus, we can assume that $\exists t_k \in \mathcal{T}$ such that $\forall t \geq t_k, V(t) = 0 \Rightarrow Qv(t) = 0$ and then we can discard these transient states. This interpretation allows to consider that $x_3 = \text{empty} \Rightarrow x_8 = \text{false}$ and hence, $\bar{x}_3 = 0 \Rightarrow \bar{x}_8 = 0$. Consequently, each state containing $\bar{x}_3 = 0$ and $\bar{x}_8 = 1$ is a circumstance without interest for this case and can be removed from the logical model. In an analogous way, the physical and the

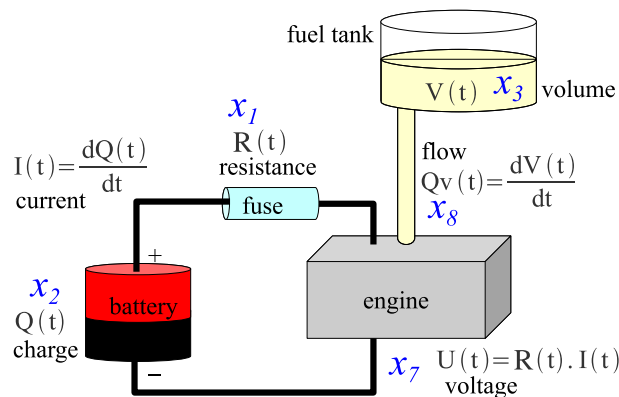


Fig. 18. Physical model of the process.

Table 2

Logical and physical functional relations.

Logical relations	Physical relations ^a	
$\bar{x}_7 = \bar{x}_1 \wedge \bar{x}_2$	$x_7^P = x_1^P \cdot \frac{dx_2^P}{dt}$	$(U(t) = R(t) \cdot \frac{dQ(t)}{dt}, (R(t) = \infty \vee Q(t) = 0) \Rightarrow U(t) = 0)$
$\bar{x}_8 = \bar{x}_3$	$x_8^P = \frac{dx_3^P}{dt}$	$(Qv(t) = \frac{dV(t)}{dt}, V(t) = 0 \Rightarrow Qv(t) = 0)$
$\bar{x}_9 = \bar{x}_7 \wedge \bar{x}_8$	$x_9^P = \alpha \cdot x_7^P \cdot x_8^P$	$(U(t) = 0 \vee Qv(t) = 0) \Rightarrow \alpha \cdot U(t) \cdot Qv(t) = 0)$

^a $R(t)$, $Q(t)$, $V(t)$, $U(t)$ and $Qv(t)$ are also noted as x_1^P , x_2^P , x_3^P , x_7^P and x_8^P , respectively.

logical relations among the other variables can be analysed and established, as in Table 2, in order to eliminate those transitory states which can be removed from the model to be built. Table 3 describes conditions, on the variables, characterising the states to remove.

From considering the previous logical and physical interpretations, we shall define a generic model of the process $M(X(t)) = \langle PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) \rangle$.

Starting with the behavioural model, the set of observation classes of $BM(X(t))$ is defined from the logical interpretation; that is to say, $C = \{C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{7,1}, C_{7,2}, C_{8,1}, C_{8,2}, C_{9,1}, C_{9,2}\}$ where

$$\begin{array}{llll} C_{1,1} = \{(\bar{x}_1, 0)\} & C_{2,2} = \{(\bar{x}_2, 1)\} & C_{7,1} = \{(\bar{x}_7, 0)\} & C_{8,2} = \{(\bar{x}_8, 1)\} \\ C_{1,2} = \{(\bar{x}_1, 1)\} & C_{3,1} = \{(\bar{x}_3, 0)\} & C_{7,2} = \{(\bar{x}_7, 1)\} & C_{9,1} = \{(\bar{x}_9, 0)\} \\ C_{2,1} = \{(\bar{x}_2, 0)\} & C_{3,2} = \{(\bar{x}_3, 1)\} & C_{8,1} = \{(\bar{x}_8, 0)\} & C_{9,2} = \{(\bar{x}_9, 1)\} \end{array} \quad (16)$$

Moreover, in order to describe state transitions, a matrix $E = (e_{ij})_{n \times n}$ which defines possible successions between two observation classes is constructed. Rows and columns of E refer to observation classes, n is the number of these ones, and $e_{ij} \in \{0, 1\}$; where, $e_{ij} = 1$ if it is possible to observe an occurrence of the class i , and after that, an occurrence of the class j , otherwise $e_{ij} = 0$ (or null). For this case study, the matrix $E = (e_{ij})_{12 \times 12}$ (Table A.5 in Appendix A.6) is defined from the physical interpretation of Fig. 18; and besides, the following assumptions are assumed: all occurrences of observation classes can be observed at the moment in which they occur, the fuel tank does not have leaks, the battery is not broken and, as already mentioned, when x_9 is works means that the engine stops. Thus, under these considerations, E is specified and its graphical representation is shown in Fig. 19. Each sequence $(C_{i,q}, C_{j,p})$, $i, j \in \{1, 2, 3, 7, 8, 9\}$, $p, q \in \{1, 2\}$ can be associated with one or more states from Definition 16, which allows characterising states between two observation classes.

Firstly, we evaluate the states in which the car can start; i.e., states where $C_{9,2} = \{(\bar{x}_9, 1)\}$ can occur. Fig. 19 indicates that we must analyse the states between $C_{7,2}$ and $C_{9,2}$; and those ones between $C_{8,2}$ and $C_{9,2}$. That is to say, $s \in S \mid s(\bar{x}_9) = 0 \wedge (s(\bar{x}_7) = 1 \vee s(\bar{x}_8) = 1)$. The physical interpretation illustrated in Fig. 18 and Table 2 permits to establish that to start the machine and to observe $C_{9,2}$, gas and voltage must be present; that is, $\bar{x}_7 = 1 \wedge \bar{x}_8 = 1$. Then, only the states in which $\bar{x}_9 = 0 \wedge \bar{x}_7 = 1 \wedge \bar{x}_8 = 1$ will be considered, which allows taking into account a more reduced set of these ones. Besides, from Table 3, we shall not consider the possible transitory states between these observation classes. Consequently, as Fig. 20 depicts, a state s_{32} , such that $s_{32}(\bar{x}_i) = 1$ for $i = 1, 2, 3, 7, 8$ and $s_{32}(x_9) = 0$, is the only state between $C_{7,2}$ and $C_{9,2}$, and between $C_{8,2}$ and $C_{9,2}$; where s_{32} is a state such that all work but the machine is stopped.

From s_{32} and after an occurrence of the observation class $C_{9,2}$, the target state will be s_{64} , where the engine works. The observation classes subsequent to $C_{9,2}$ (see Fig. 19) that can occur in s_{64} are analysed from the physical model in order to determinate new possible transitions between states. Thus, this kind of analysis is carried out with all sequences of observation classes depicted in Fig. 19 and the set of states resultant from removing the physically impossible states.

From the previous reasoning, the set S of states, the set C of observation classes and the transition function γ can be defined in order to specify the behavioural model $MB(X(t))$ (Expr. (A.8), Appendix A.3). In particular, the perception model (Section 7.2) must be modified since the values associated with transitory states do not correspond with states of abnormal behaviour.

Consequently, in this phase, a generic model $M(X(t)) = \langle PM(X(t)), SM(X(t)), FM(X(t)), BM(X(t)) \rangle$ of the process is defined. The complete specification of $M(X(t))$ can be found in Appendix A.3.

As a result, we consider that the construction of a generic model of the process requires interpretations of the expert's knowledge both in logical and physical terms (Table A.6 in Appendix A.6). These interpretations along with modelling decisions allow a reduction from 512 to only 16 states physically possible and of interest for diagnosing behaviour problems (see Expr. (A.8) in Appendix A.3). The

Table 3

Conditions on variables in order to identify transient states which can be discarded from the process model.

Logical relations	Physical relations
Transient states	Transient states
$(\bar{x}_1 = 0 \vee \bar{x}_2 = 0) \wedge \bar{x}_7 = 1$	$(R(t) = R_\infty \vee C(t) = 0) \wedge U(t) \neq 0$
$\bar{x}_3 = 0 \wedge \bar{x}_8 = 1$	$V(t) = 0 \wedge Qv(t) \neq 0$
$(\bar{x}_7 = 0 \vee \bar{x}_8 = 0) \wedge \bar{x}_9 = 1$	$(U(t) = 0 \vee Qv(t) = 0) \wedge \alpha \cdot U(t) \cdot Qv(t) \neq 0$

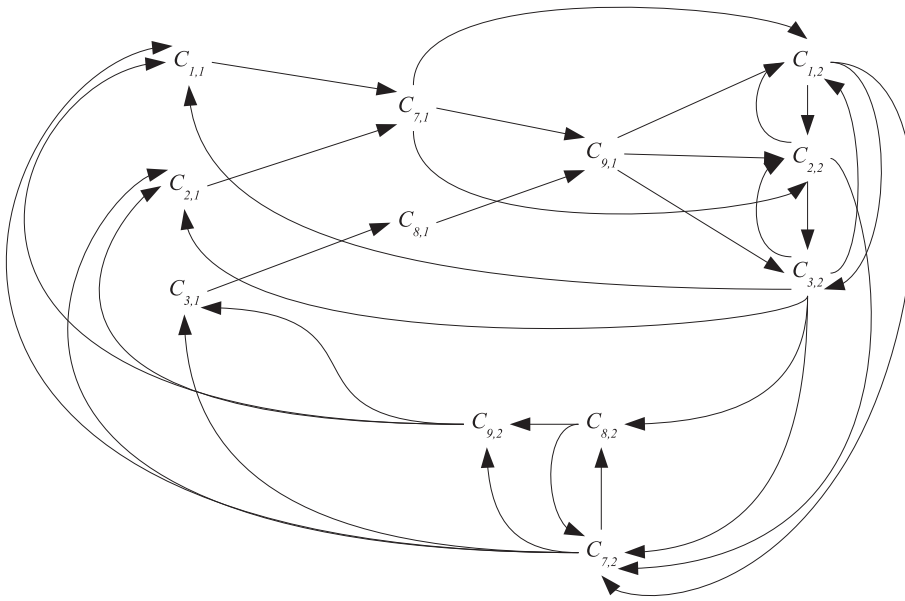


Fig. 19. Possible sequences of observation classes.

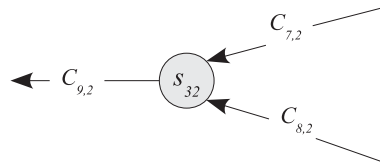


Fig. 20. State between two sequences of observation classes.

logical model of Fig. 16 describes the structure of the expert's diagnosis reasoning and the physical model of Fig. 18 provides the diagnosis knowledge required for this reasoning. Thus, both logical and physical models are necessary and complement each other. We believe that these models are, ultimately, those ones "constructed" by experts where, in practice, the combination of these ones simplifies the diagnosis task.

Moreover, the resultant model $M(X(t))$ admits the application of model-based diagnosis techniques; and simultaneously, introduces the dimension of time allowing to model the dynamic of the process in a behavioural model.

8. Discussion

This section discusses three main points about this paper: i) TOT as a key theory to combine an Expert Model M_e and a Data Model M_d as shown in Fig. 1, ii) a method of abstraction, based on the TOT concept of timed observation, can be defined; and thus, logical relations between an Expert Model M_e and a Data Model M_d residing at different levels of abstraction can be established (cf. Fig. 2) and, iii) the position of the TOM4D methodology in the KE domain.

8.1. Combining KE and KDD

In the beginnings of KE, the systems were described in terms of representation techniques assuming that experts' knowledge could be translated almost directly into computer implementations from observations or experts' verbalised expressions. However, this view resulted unsuitable for real world problems owing to the fact that it is based on the hypothesis that a symbol "captures" the semantics of a concept. In our opinion, most KE approaches are not sufficient to deal with big amounts of timed data of the real world because, precisely, a particular symbol cannot "capture" all the semantic of a real-world concept. Our proposal, based on TOT, reverses this point of view by starting from the fact that a symbol, if possible, can only denote a very small part of a real-world concept. Consequently, the semantics of a concept is not captured by a symbol but that it will emerge from the binary relations that can be



Fig. 21. The ω sequence of 100 timed observations (top–down: $C_{1,1}$, $C_{2,1}$, $C_{3,1}$, $C_{7,1}$, $C_{8,1}$ and $C_{9,1}$).

logically constructed between this symbol and the others. In other words, the semantics associated with a symbol is what emerging from the set of all binary relations that involve this symbol.

TOT provides a mathematical basis of a KDD process called TOM4L (Timed Observation Mining For Learning) [53,38] which aims to automatically find the mentioned binary relations from a set Ω of timed observations sequences; and consequently, automatically to produce a functional model $FM(\Omega)$ and a behavioural model $BM(\Omega)$. TOM4L implements the main concepts and the main algorithms of the mentioned theory in an unsupervised learning process from data; and thus, establishes a purely syntactic computing process which contains no semantics but algebra; i.e., symbol-manipulation rules. The TOM4L models describe part of an observed process $\Theta(X, \Delta)$ which originates timed observation sequences with the same stochastic properties as those of the set Ω .

Owing to the fact that both TOM4D and TOM4L are based on the same theory, TOM4D functional and behavioural models and TOM4L functional and behavioural models share the same formalism; and consequently, they can be compared to each other. In particular, these models complement together [54] where TOM4D models provide semantic content to an observed set of phenomena and TOM4L models describe an observed process $\Theta(X, \Delta)$ under the form of a timed stochastic process that produces sequences of timed observations.

The TOM4L KDD process is based on representing the set Ω under the form of a structure $K(\Omega)$ called *K-Representation* which is made up of 12 matrices. One of the most important matrices measures the interlacing of the occurrences of two classes C_i and C_j in Ω through a function $\alpha_{ij} = \alpha((\omega_i, \omega_j)|\Omega)$ where ω_i and ω_j are sequences of Ω made up of timed observations of the classes C_i and C_j , respectively. This function satisfies the three axioms of Cox–James [55] so that α_{ij} can be used as a plausibility measure within an induction reasoning. In particular, this property results from the *BJ-Measure* [36] which is based on the Kullback–Leibler divergence [56] and a generalisation of Shannon's communication channel [32]. In particular, α_{ij} provides a quantitative core for Principle 2 in Section 5.3 due to the fact that a relation $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ can be induced from Ω if and only if $\alpha((\omega_i, \omega_j)|\Omega) \geq 0.5$ [36].

This property allows to implement a safe induction reasoning to discover a minimal set of binary temporal relations $R = \{r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])\}$ from a K-Representation $K(\Omega)$. From this minimal set, the Tom4BN (Timed Observation Mining for Bayesian Networks) algorithm [38,36] and the BJT4S (BJ-Measure for Signatures) algorithm [37] produce a functional model $FM(\Omega)$ (cf. Fig. 22a) and a behavioural model $BM(\Omega)$ (cf. Fig. 23), respectively.

In order to illustrate the first point of this discussion, as Fig. 21 depicts, a sequence ω of 100 timed observations of the classes $C_{1,1}$, $C_{2,1}$, $C_{3,1}$, $C_{7,1}$, $C_{8,1}$ and $C_{9,1}$ was built from the a priori probabilities in Table 4, according to the time stochastic distribution method described in [39].

From these timed observations, the Tom4BN algorithm computes the functional model $FM(\omega)$ under the form of the naive Bayesian networks illustrated in Fig. 22a. This model allows defining the functional model shown in Fig. 22b whose functions correspond to those ones of the TOM4D functional model (see Section 7). Hence, the TOM4L functional model obtained from the sequence ω can be easily compared with the TOM4D functional model defined from the experts' knowledge. The main difference between them comes from the probabilities that are associated by the Tom4BN algorithm to the relations between the values of the variables. For example $f_4(x_1, x_2)$ in Fig. 22b, describing the rules in Expr. (17) which explain $x_7 = \text{off}$, is equivalent to $f_4(x_1, x_2)$ of the TOM4D functional model. These rules have no meaning in a TOM4L functional model; however, the corresponding TOM4D model (Fig. 16) provides semantic content to them establishing that (R_1) corresponds to a double fault and (R_2) and (R_3) correspond to a single fault in the observed process. The probability distribution provided by the TOM4L estimates that the frequency of the double fault (68%) is twice more frequent than the single faults (30%), which is not very realistic in a real world car but is representative of the probabilities in Table 4.

$$\begin{aligned}
 (x_1 = \text{blown} \wedge x_2 = \text{low}) &\Rightarrow x_7 = \text{off} (68\%) & (R_1) \\
 (x_1 = \neg \text{blown} \wedge x_2 = \text{low}) &\Rightarrow x_7 = \text{off} (17\%) & (R_2) \\
 (x_1 = \text{blown} \wedge x_2 = \neg \text{low}) &\Rightarrow x_7 = \text{off} (13\%) & (R_3)
 \end{aligned} \tag{17}$$

Table 4
Prior probabilities of the car example [38,36].

$C_{1,1} = \{(x_1, \text{blown})\}$	$C_{2,1} = \{(x_2, \text{low})\}$	$C_{3,1} = \{(x_3, \text{empty})\}$	$C_{7,1} = \{(x_7, \text{off})\}$	$C_{8,1} = \{(x_8, \text{false})\}$	$C_{9,1} = \{(x_9, \neg \text{works})\}$
$P(C_{1,1})$	$P(C_{2,1})$	$P(C_{3,1})$	$P(C_{7,1})$	$P(C_{8,1})$	$P(C_{9,1})$
0.05	0.15	0.3	0.2	0.2	0.1

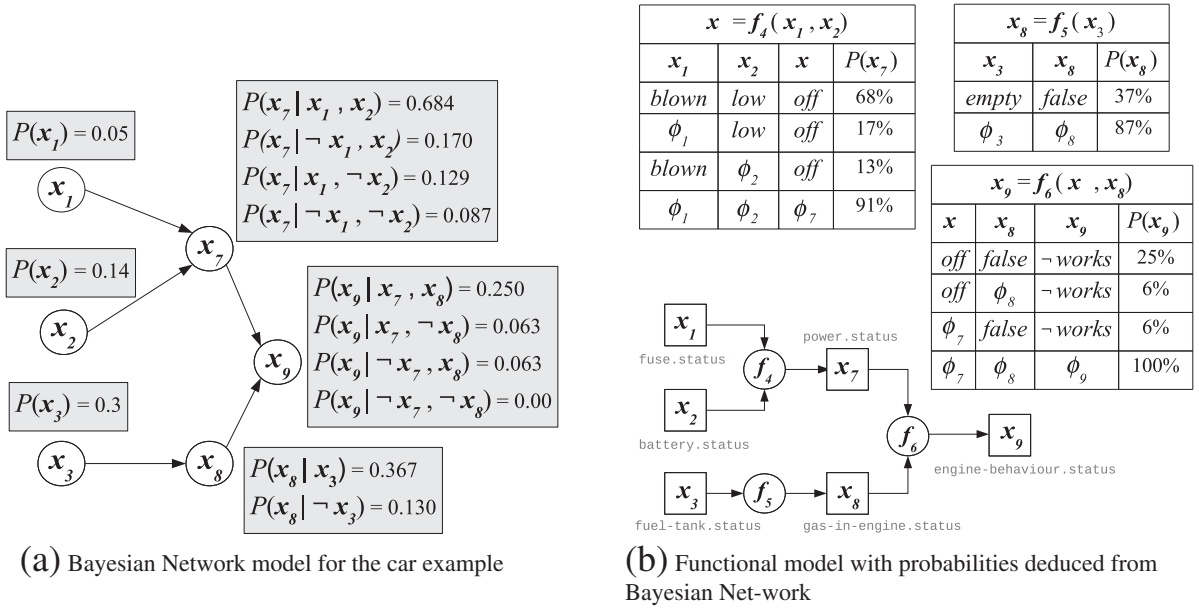
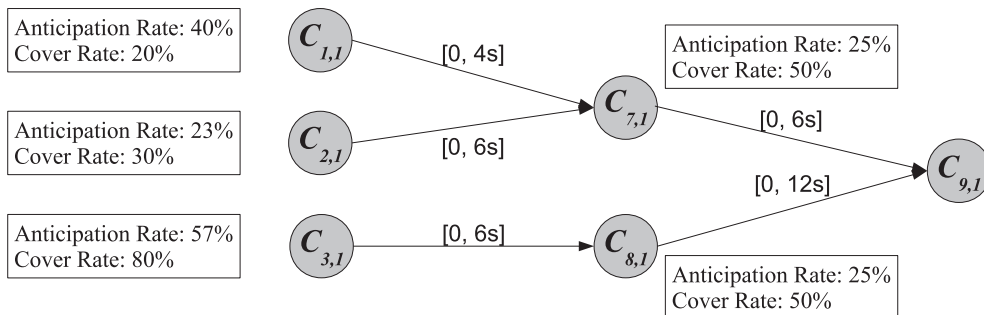


Fig. 22. Functional model from TOM4L [38].

The BJT4S algorithm produces from ω the behavioural model illustrated in Fig. 23 under the form of an abstract chronicle model according to Definition 6. The timed constraints are computed from observations timestamps according to the stochastic properties (i.e., Poisson distribution) of the stochastic clocks Γ_i corresponding to the occurrences of the classes C_i . For example $\{r_{17}(C_{1,1}, C_{7,1}, [0, 4 \text{ s}]), r_{79}(C_{7,1}, C_{9,1}, [0, 6 \text{ s}])\}$ in the figure is a path (cf. Definition 7) made up of two timed binary relations. The *Anticipation Rate* indicates that in 40% of the cases, when an occurrence of $C_{1,1}$ is followed by an occurrence of $C_{7,1}$ in at most 4 s, then an occurrence of $C_{9,1}$ takes place in at most 6 s. The *Cover Rate*, for its part, means that in 20% of the cases in which an occurrence of $C_{9,1}$ is observed, the path $\{r_{17}(C_{1,1}, C_{7,1}, [0, 4 \text{ s}]), r_{79}(C_{7,1}, C_{9,1}, [0, 6 \text{ s}])\}$ was observed.

Clearly, the TOM4L behavioural model of Fig. 23 is a sub-model of the TOM4D behavioural model of Fig. 19. Once again, the latter provides semantic content to the former; that is, the corresponding TOM4D sub-model describes the propagation of a fault ($x_1 = \text{blown}$ or $x_2 = \text{low}$ or $x_3 = \text{empty}$) within the car engine, leading undesired states (Fig. A.27, Appendix A.6) which satisfy the condition of abnormal operating mode defined in the TOM4D Perception Model. Although incomplete, the advantage of the TOM4L behavioural model is to provide a quantification of the dynamic properties within the observed process; i.e. temporal constraints. This is of major importance in monitoring, diagnosis and prognosis tasks owing to the fact that assigning timestamps to observations might reveal surprises about the causal relation orientations within real-world dynamic processes. Besides, it is extremely difficult for an expert to determine realistic timed constraints from the graphical representations of a set of continuous time function $x_i(t)$ [35]; therefore, the possibility of estimating automatically these constraints is of great interest.

Consequently, an expert's a priori knowledge about the behaviour of a dynamic process can be made explicit through the TOM4D methodology and it can be confronted with the process data measured over time by means of the TOM4L KDD process. Inversely,

Fig. 23. Behavioural model from TOM4L. Signature tree of the observation class $C_{9,1}$ [38].

when only a small amount of a priori knowledge about the behaviour of a dynamic process is present, the TOM4L KDD process can be applied in order to start the modelling phase from the available data. Thus, these models belonging to different disciplines, such as KE and KDD, can be very easily related and compared to each other allowing cross-validation between experts' models and measured data.

8.2. Abstraction levels in TOT

The car example is simple enough to illustrate the complementarity between TOM4D models and TOM4L models. The possible cross-validation through these models strongly increases both the confidence of the resulting models and the efficiency of the modelling process. However, the simplicity of this example masks another reality which opposes to an approach combining a KE methodology with a KDD process. As Fig. 2 illustrates, in complex processes there exists a huge conceptual distance between the abstraction level of the cognitive tasks carried out by an expert and the abstraction level of raw data. A clear example of this is the Sachem system of blast furnaces. Sachem [35] is a large-scale real time knowledge-based system designed to monitor and diagnose blast furnaces. In particular, 2000 pages make up the linguistic model formulated from experts' knowledge about 175 blast furnace phenomena, where the occurrences of these phenomena are detected through applying the spatial segmentation principle depicted in Fig. 5. The real time observation of 500 high-level time functions $f_i(t)$ accounts for the behaviour of the blast furnace. The values of these functions are computed from the values of 2850 intermediate-level time functions where these latter are calculated, in turn, from real time values obtained from 1200 concrete sensors [57]. The sensors provide raw data measured directly on the blast furnace or manually entered in the plant information system. Sachem System has been designed according to the CommonKADS methodology. The template of the Sachem problem solving method, partially given in [35], is based on three levels of abstraction, which are absolutely necessary in order to analyse the behaviour observed on the 500 high-level time functions $f_i(t)$ and then to produce the occurrences of the blast furnace phenomena. This constitutes a concrete example of what Floridi [19] calls a *nested gradient of abstraction* presented in Section 3.3. In particular, the Sachem *abstraction method* is based on a model matching technique [35].

There does not exist, up to our knowledge, a mathematical theory which includes a nested gradient of abstraction and its corresponding method of abstraction. The concept of timed observation has been built in order to design a *timed observation algebra* which allows to add and to subtract timed observations, by computing abstract timed observations from observing binary temporal relations $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$ (Definition 5). In this algebra, finding two timed observations $C_i(t_{k-n}) = (\delta_i, t_{k-n})$ and $C_j(t_k) = (\delta_j, t_k)$ in a sequence ω , which satisfy the constraints of a binary temporal relation $r_{ij}(C_i, C_j, [\tau_{ij}^-, \tau_{ij}^+])$, corresponds to compute the temporal addition $C_i(t_{k-n}) \overset{[\tau_{ij}^-, \tau_{ij}^+]}{+} C_j(t_k) = C_{ij}(t_k)$. The result of this addition is a timed observation $C_{ij}(t_k) = (\delta_{ij}, t_k)$ of the abstract class C_{ij} where δ_{ij} does not belong to the initial set Δ of the observed process $\Theta(X, \Delta)$ but it is a new constant identifying the abstract observation class. Consequently, by construction, δ_{ij} belongs to an abstraction level immediately superior to the abstraction level of δ_i and δ_j . In particular, [35] demonstrates that observing a path of n binary temporal relations corresponds to compute a temporal addition of $n + 1$ timed observations whose result is a timed observation. Thus, the concept of timed observation is a paradigm which can be used to provide a mathematical definition of a nested gradient of abstraction, based on the mentioned temporal addition.

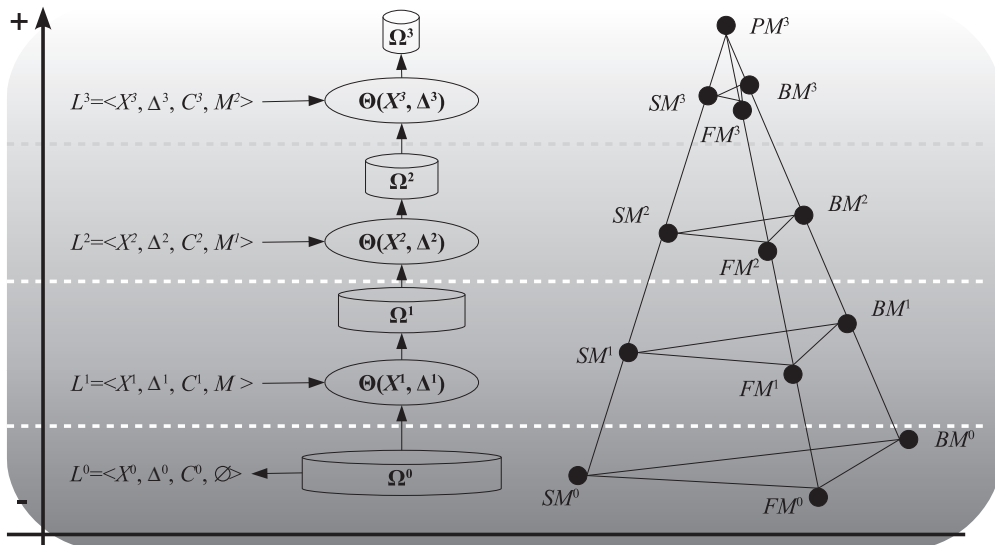


Fig. 24. Nested gradient of abstraction according to the TOT.

As an example, in smart environments which recognise human behaviour from sensor data, the definition of abstraction levels results fundamental. This real time recognition task is particularly complex due to the fact that multiple activities (e.g., cooking, washing, and watching TV) can be executed at the same time, either by one or by several people; and besides, the same detected action can be associated to several activities depending on the context in which it is carried out. In particular, we have defined a nested gradient of abstraction which has been applied to modelling of elderly behaviour at home in order to recognise the mentioned behaviour and to prevent accidents [58,59].

This theoretical framework of abstraction levels, formalised in [59] and illustrated in Fig. 24, defines an abstraction level $L^i = \langle X^i, \Delta^i, C^i, M^{i-1} \rangle$ (with $i \geq 1$) as a structure which links a set X^i of variable names, a set Δ^i of constants, a set C^i of observation classes and a set M^{i-1} of behavioural models, these last ones defined from the set C^{i-1} of observation classes in the level L^{i-1} . Particularly, M^{i-1} specifies the timed observation additions which allow creating timed observations at the abstraction level L^i . The lowest abstraction level $L^0 = \langle X^0, \Delta^0, C^0, \Phi \rangle$ is the concrete level (i.e., raw data). The abstraction method is implemented by a program $\Theta(X^i, \Delta^i)$ which computes the n -ary timed observation additions specified by the models M^{i-1} . This program is unique; thus, the abstraction method is only a concrete parametrisation at each level. In particular, a general version of the mentioned program is currently developed in Java integrating a software framework called TOM4K (Timed Observation Management for Knowledge) which implements the TOM4L KDD process.

As Fig. 24 depicts, a dynamic process $X(t)$ is modelled through a nested gradient of abstraction where each level defines a TOM4D model $M^i(X^i(t)) = \langle PM^i(X^i(t)), SM^i(X^i(t)), FM^i(X^i(t)), BM^i(X^i(t)) \rangle$. Consequently, n levels of abstraction define a network of n TOM4D models organised like a pile of n tetrahedrons; where all the models are linked to each other by means of the nested gradient of abstraction.

We are currently formalising these relations in order to add the adequate modelling principles to the TOM4D methodology. However, this theoretical work is based on the following hypotheses.

- The introduction of the concept of nested gradient of abstraction is required to handle the complexity of real world dynamic processes (e.g., Sacher).
- The mentioned concept needs a mathematical basis and Timed Observation Theory provides the theoretical framework required.
- The relations between two TOM4D models $M^{i-1}(X^{i-1}(t))$ and $M^i(X^i(t))$ must respect the properties identified by Newell [18] described in Section 3.3.
- TOM4D enables a multi-scale modelling process owing to the fact that the modelling principles are adequate.
- The complexity of a multi-scale modelling process requires a strong cooperation between a KE methodology and a KDD process in order to validate the models defined at each level of abstraction. The natural complementarity between the TOM4D KE methodology and the TOM4L KDD process is particularly efficient for this purpose. In particular, a set of modelling tools must be integrated within the TOM4K framework in order to facilitate the management of a network of n TOM4D models $M^i(X^i(t))$.

In this way, a nested gradient of abstraction provides conceptual tools which allow managing the representation of a knowledge corpus where the complexity is mainly conceptual. This kind of complexity has been called the *intrinsic complexity* by C. H. Bennet in order to mark the difference with the well-known *Kolmogorov–Chaitin complexity*. The latter evaluates the complexity according to the dimension of the model to be described; that is to say, according to the number of variables $x_i \in X$ and the number of constants δ_j

$\in \Delta = \bigcup_{x_i \in X} \Delta_{x_i}$ where each Δ_{x_i} is the domain of the corresponding variable x_i .

The Kolmogorov–Chaitin complexity illustrates one of the main limitations of the TOM4D methodology. The potential number of timed observation classes defined in a TOM4D behavioural model $BM(X(t))$ is directly proportional to the cardinality of $\Delta \times \Delta$. In other words, considering Δ of n elements, the number of discernible states is n^2 and the number of state transitions is then $n^2 \cdot (n^2 - 1)$. Thus, 10 binary variables determine 400 discernible states and 159,600 state transitions to be examined. Such number is too large regarding the small amount of variables. Fortunately, owing to the numerous state transitions physically impossible, the number of significant state transitions decreases strongly.

Nevertheless, the problem remains when modelling real world processes. Consequently, TOM4D has then been recently extended in order to define a process as a network of sub-processes where each one is modelled with an autonomous TOM4D model. The relations between sub-processes are established through shared variables. As a consequence, the global model of a network of sub-processes results from the combination of TOM4D sub-models. Thus a local validation of each sub-model can be carried out simplifying strongly the global validation of the model. Although this simplifies the modelling process, the algorithmic usage of these models becomes complex owing to the fact that the state transitions have to be propagated to the other models through the network structure.

In particular, the extended TOM4D methodology has been applied to the risk assessment of a hydraulic dam failure [60,61,49,50], the Sapins French dam, in operation since November 1978. In October 1988, for safety reasons, the reservoir was completely emptied in order to improve its structural safety and put back into service. The extended TOM4D methodology was applied to this dam producing a network of six TOM4D models which has been validated by experts of the French government. Each TOM4D model corresponds to one of the six sub-processes which constitute a dam, and each TOM4D model is made up of three variables having three possible values. Thus, each TOM4D model defines a set Δ of 9 constants, 81 discernible states and 6480 potential state transitions. Taking into account the physically impossible states this number is reduced to only 3 discernible states in one of the sub-processes and to a maximum of 12 discernible states in the other five most complex sub-processes.

As a consequence, the global state space defined by the extended TOM4D methodology for the Sapins dam contains $12^5 \cdot 3 = 746,496$ discernible states, resulting the number of potential transitions (superior to $557 \cdot 10^9$) out of any possibility of modelling

and validation. This partially explains why, up to now, there was no complete models of the dynamic of a hydraulic dam. Hence, a specific diagnosis algorithm has been designed in order to manage the relations between the six processes which hinder the building of the complete behavioural model [62,63]. It is noteworthy that the TOM4D extension to networks of sub-processes is mainly concerned with the algorithmic aspects of the model utilisation; therefore, the added modelling principles are quite natural and obvious.

In this application, all the sub-processes of the network belong to the same abstraction level. The complexity described does not require the definition of a nested gradient of abstraction levels; however, adequate composition laws which allow propagating timed observations through a network of TOM4D models are necessary. Furthermore, this extended methodology allows defining generic models; e.g., the five TOM4D models corresponding to the five sub-processes of the dam are structurally the same differing only in the set of variables.

Ultimately, networks of models deal with dimensional complexity and abstraction levels address conceptual complexity. The future introduction of the notion of nested gradient of abstraction levels in the TOM4D methodology will provide tools which allow a multi-scale and a multi-grain modelling process. This one is required when dealing with highly complex systems which are currently under consideration; e.g., smart towns or multi-modal distribution networks.

8.3. Position of TOM4D

The first point to take into account in order to place our approach respect to KE is to consider that this methodology is still evolving. However, the methodology core is consolidated and is precisely what we present in this paper along with future evolutions. In particular, our current theoretical work has the purpose of introducing modelling principles which allow the notion of a nested gradient of abstraction in order that TOM4D is extended in that direction. The second element to take into account is that, according to our knowledge, there is no other KE methodology whose objective is to build the dynamic process model that an expert has in mind when formulating her or his knowledge. Most of KE approaches, if not all, aim at modelling the expert's knowledge in a similar way to the CommonKADS methodology.

Finally, the last element to take into account is the fact that TOM4D is a KE “symbol driven modelling” approach which allows controlling the introduction of semantic content within the models. Owing to the fact that the latest developments in KE mainly propose tools which capture semantic content, we considered more suitable to select an adequate tool, e.g. CommonKADS, than to develop new ones. Therefore, TOM4D leans on available contributions in the KE domain but does not try to replace some of them.

Nevertheless, we argue that despite of the latest progress, the KE approaches still pose two fundamental problems: i) the cost of the knowledge management process and ii) the intrinsic difficulty of validating an expert's knowledge model. These two problems are linked to each other since finishing the knowledge acquisition phase entails to validate the expert's model. This requires either manual validation or to wait for the resulting KBS which implement the model in order to verify that the system provides the same results which could be obtained from an expert [64,65]. Therefore, we claim that combining a KDD process with a KE methodology is an efficient way to simultaneously reduce costs in the knowledge management process and to facilitate the expert's knowledge model validation with the available data. However, we affirm also that such a combination requires the following.

- To elicit the dynamic process model used by an expert when formulating her or his knowledge.
- The formalism which represents the models must be adapted for use within a KDD process. This requirement justifies the TOM4D principle of “symbol driven modelling”.
- A nested gradient of abstraction is necessary in order to reduce the gap between the abstraction level of the expert's model and that one of raw data, where these ones could be used to validate the mentioned model. Such a concept is required when the complexity of the dynamic process under consideration is large, this is essentially the case in which a KE methodology is required in order to build an decision system.

Therefore, up to our knowledge, there is no other KE methodology which i) allows to model not only the expert's knowledge but also the process under consideration and ii) is designed in order to be combined with a KDD process. This positions the TOM4D methodology in a particular place within the scope of the KE domain and makes room for further developments. In particular, we believe that including the Floridi's and Newell's analysis about abstraction levels within KE approaches promises interesting developments in both the KE domain and the KDD domain. Besides, we think that TOT can advantageously contribute to these developments.

9. Conclusions

Dynamic process modelling is generally accomplished from experts' knowledge, this could be complemented from analysing and mining data which describe the process evolution. Nevertheless, although Knowledge Engineering (KE) and Knowledge Discovery in Database (KDD) are disciplines which pursue knowledge creation, they are seldom or never considered in an integral way.

In this work we presented a KE methodology called Timed Observation Modelling For Diagnosis (TOM4D), which proposes a primarily syntax-driven modelling approach where semantic content is introduced in a gradual and controlled way through the CommonKADS conceptual approach, Formal Logic and the Tetrahedron of States framework.

The principles and fundamentals of TOM4D were presented along with a didactic example that illustrates the modelling approach which, we believe, makes explicit the specialists' knowledge in a way that this one can be compared with the real process. Thus, this methodology allows to build a process model from experts' knowledge and data where this model, by construction, can be directly

associated with the experts' Knowledge Model and, at the same time, can be collated with real process models obtained from data, as discussed in Section 8. In this sense, this approach allows reducing the distance between specialists' knowledge and data, by binding both universes, and it proposes to lead both fields of study towards a holistic view.

Real world problems have been addressed through TOM4D. In particular, the security of the dam of Cubblize (France) where the resultant TOM4D models have been validated by the hydraulic dam experts of the French governmental organisation (Cemagref) which controls the security of hydraulic civil engineering structures in the corresponding country [49]. Moreover, nowadays we are utilising this methodology and the TOM4L process in order to model human behaviour in a smart environment from gerontologists' knowledge and data, in the context of the GerHome Project of the Centre Scientifique et Technique du Bâtiment (CSTB) of Sophia Antipolis (France).

Appendix A. Elements of the application of TOM4D to the didactic example

A.1. Interpretation from CommonKADS

Fig. A.25 depicts the domain knowledge of a case study about the diagnosis of problems with a car, where nine rules constitute the knowledge provided by an expert [2]. These rules can be interpreted as (R_1) if the fuse is blown then the result of the fuse inspection is broken, (R_2) if the fuse is blown then the power is off, (R_7) if the power is off, the engine does not start, and so on.

In particular, we shall limit ourselves to introduce a concise description of the interpretation from CommonKADS, carried out by the authors in [2], of the introduced knowledge pieces (Fig. A.25), and to present the domain rules specified in the Conceptual Modelling Language (CML) [2].

The diagnosis template presents a typical domain schema in which each system being diagnosed can be characterised in terms of two types of features: those ones that can be observed and those ones that can represent an internal state of the system. Consequently, the concepts *fuse inspection*, *battery dial* and *gas dial* are considered observable features; and *fuse*, *battery*, *fuel tank*, *power*, *gas in engine* and *engine behaviour* are considered concepts that allow representing the states of the car. In particular, *engine behaviour* refers to a state which can be perceived in some way; therefore, the last concepts associated with car states can in turn be classified as visible or invisible.

For their part, the arrows in Fig. A.25 show dependences between the knowledge pieces. These dependences are rules which indicate relations between domain concepts; for example, “if there is not gas in engine, engine stops” establishes a causal relation between the concepts (that is, `gas-in-engine.status = false => engine-behaviour.status = stops`). In this case study, two types of dependences can be observed: rules that indicate that a value assumed by an entity *causes* a certain value in other entity; and rules which establish that a value assumed by an entity *has a particular manifestation* in another entity. The mentioned dependences are specified in language CML in Expr. (A.1) as nine rules of domain knowledge.

```

fuse.status = blown HAS-MANIFESTATION fuse-inspection.value = broken;      (R1)
fuse.status = blown CAUSES power.status = off;                          (R2)
battery.status = low CAUSES power.status = off;                         (R3)
battery.status = low HAS-MANIFESTATION battery-dial.value = zero;        (R4)
fuel-tank.status = empty HAS-MANIFESTATION gas-dial.value = zero;        (R5)
fuel-tank.status = empty CAUSES gas-in-engine.status = false;           (R6)
power.status = off CAUSES engine-behaviour.status = does-not-start;      (R7)
gas-in-engine.status = false CAUSES engine-behaviour.status = does-not-start; (R8)
gas-in-engine.status = false CAUSES engine-behaviour.status = stops;     (R9)

```

(A.1)

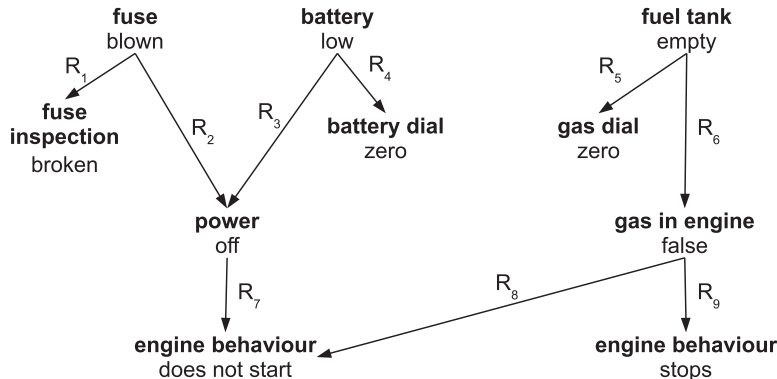


Fig. A.25. Knowledge pieces in the car-diagnosis domain (Taken from [2, p. 92]).

A.2. Definition of scenario model $M(\Omega) = \langle SM(\Omega), FM(\Omega), BM(\Omega) \rangle$

$M(\Omega) = \langle SM(\Omega), FM(\Omega), BM(\Omega) \rangle$ such that,

$SM(\Omega) = \langle COMPS, R_{port}, R_{xport} \rangle$ where

$$\begin{aligned}
 COMPS &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\} \\
 R_{port} &= \{out(c_1) = in(c_4), out(c_1) - in_1(c_7), out(c_2) = in(c_5), out(c_2) = in_2(c_7), \\
 &\quad out(c_3) = in(c_6), out(c_3) = in(c_8), out(c_7) = in_1(c_9), out(c_8) - in_2(c_9)\} \\
 R_{xport} &= \{out(c_1) = x_1, out(c_2) = x_2, out(c_3) = x_3, out(c_4) = x_4, out(c_5) = x_5, \\
 &\quad out(c_6) = x_6, out(c_7) = x_7, out(c_8) = x_8, out(c_9) = x_9\}
 \end{aligned} \tag{A.2}$$

$FM(\Omega) = \langle \Delta, F, R_f \rangle$ where

$\Delta = \Delta_{x_1} \cup \Delta_{x_2} \cup \Delta_{x_3} \cup \Delta_{x_4} \cup \Delta_{x_5} \cup \Delta_{x_6} \cup \Delta_{x_7} \cup \Delta_{x_8} \cup \Delta_{x_9}$ with

$$\begin{aligned}
 \Delta_{x_1} &= \{blown, \phi_1\} & \Delta_{x_4} &= \{broken, \phi_4\} & \Delta_{x_7} &= \{off, \phi_7\} \\
 \Delta_{x_2} &= \{low, \phi_2\} & \Delta_{x_5} &= \{battery_zero, \phi_5\} & \Delta_{x_7} &= \{off, \phi_7\} \\
 \Delta_{x_3} &= \{empty, \phi_3\} & \Delta_{x_6} &= \{gas_zero, \phi_6\} & \Delta_{x_9} &= \{\neg works, \phi_9\}
 \end{aligned}$$

$F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$ with

$$\begin{aligned}
 f_1 &: \Delta_{x_3} \rightarrow \Delta_{x_6}, \\
 f_2 &: \Delta_{x_2} \rightarrow \Delta_{x_5}, \\
 f_3 &: \Delta_{x_1} \rightarrow \Delta_{x_4}, \\
 f_4 &: \Delta_{x_1} \times \Delta_{x_2} \rightarrow \Delta_{x_7}, \\
 f_5 &: \Delta_{x_3} \rightarrow \Delta_{x_8}, \\
 f_6 &: \Delta_{x_7} \times \Delta_{x_8} \rightarrow \Delta_{x_9} \text{ and such that,}
 \end{aligned}$$

$f_1 : \Delta_{x_3} \rightarrow \Delta_{x_6}$	
y	$f_1(y)$
empty	gas_zero
ϕ_3	ϕ_6

$f_2 : \Delta_{x_2} \rightarrow \Delta_{x_5}$	
y	$f_2(y)$
low	battery_zero
ϕ_2	ϕ_5

$f_3 : \Delta_{x_1} \rightarrow \Delta_{x_4}$	
y	$f_3(y)$
blown	broken
ϕ_1	ϕ_4

(A.3)

$f_4 : \Delta_{x_1} \times \Delta_{x_2} \rightarrow \Delta_{x_7}$		
y_1	y_2	$f_4(y_1, y_2)$
<i>blown</i>	<i>low</i>	<i>off</i>
ϕ_1	<i>low</i>	<i>off</i>
<i>blown</i>	ϕ_2	<i>off</i>
ϕ_1	ϕ_2	ϕ_7

$f_5 : \Delta_{x_3} \rightarrow \Delta_{x_8}$	
y	$f_5(y)$
empty	false
ϕ_3	ϕ_8

$f_6 : \Delta_{x_7} \times \Delta_{x_8} \rightarrow \Delta_{x_9}$		
y_1	y_2	$f_6(y_1, y_2)$
<i>off</i>	<i>false</i>	$\neg works$
<i>off</i>	ϕ_8	$\neg works$
ϕ_7	<i>false</i>	$\neg works$
ϕ_7	ϕ_8	ϕ_9

$$R_f = \{x_6 = f_1(x_3), x_5 = f_2(x_2), x_4 = f_3(x_1), x_7 = f_4(x_1, x_2), x_8 = f_5(x_3), x_9 = f_6(x_7, x_8)\}$$

$BM(\Omega) = \langle S, C, \gamma \rangle$ where

$S = \{s_1 \dots s_{512}\}$, ($2^9 = 512$ characterised states, S is partially presented due to its length)

$S = \{s : X \rightarrow \Delta \mid s(x_i) = \delta, x_i \in X, \delta \in \Delta_{x_i}\}$									
S	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
s_1	blown	low	empty	broken	battery_zero	gas_zero	off	false	$\neg works$
s_2	ϕ_1	low	empty	broken	battery_zero	gas_zero	off	false	$\neg works$
s_3	blown	ϕ_2	empty	broken	battery_zero	gas_zero	off	false	$\neg works$
...
s_{512}	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9

(A.4)

$$C = \{C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{4,1}, C_{4,2}, C_{5,1}, C_{5,2}, C_{6,1}, C_{6,2}, C_{7,1}, C_{7,2}, C_{8,1}, C_{8,2}, C_{9,1}, C_{9,2}\}$$

$$\begin{aligned}
 C_{1,1} &= \{(x_1, blown)\} & C_{4,1} &= \{(x_4, broken)\} & C_{7,1} &= \{(x_7, off)\} \\
 C_{1,2} &= \{(x_1, \phi_1)\} & C_{4,2} &= \{(x_4, \phi_4)\} & C_{7,2} &= \{(x_7, \phi_7)\} \\
 C_{2,1} &= \{(x_2, low)\} & C_{5,1} &= \{(x_5, battery_zero)\} & C_{8,1} &= \{(x_8, false)\} \\
 C_{2,2} &= \{(x_2, \phi_2)\} & C_{5,2} &= \{(x_5, \phi_5)\} & C_{8,2} &= \{(x_8, \phi_8)\} \\
 C_{3,1} &= \{(x_3, empty)\} & C_{6,1} &= \{(x_6, gas_zero)\} & C_{9,1} &= \{(x_9, \neg works)\} \\
 C_{3,2} &= \{(x_3, \phi_3)\} & C_{6,2} &= \{(x_6, \phi_6)\} & C_{9,2} &= \{(x_9, \phi_9)\}
 \end{aligned}$$

$\gamma: S \times C \rightarrow S$, such that

$$\begin{aligned}
 \gamma(s_d, C_{1,1}) &= s_b, & s_d(x_1) &\neq \text{blown} \wedge s_b(x_1) = \text{blown} \wedge s_b(x_4) \neq \text{broken} \\
 \gamma(s_b, C_{4,1}) &= s_c, & s_b(x_1) &= \text{blown} \wedge s_b(x_4) \neq \text{broken} \wedge \\
 & & s_c(x_4) &= \text{broken} \wedge s_b(x_7) \neq \text{off} \\
 \gamma(s_c, C_{7,1}) &= s_d, & s_c(x_4) &= \text{broken} \wedge s_b(x_7) \neq \text{off} \wedge \\
 & & s_d(x_7) &= \text{off} \wedge s_d(x_9) \neq \neg \text{works} \\
 \gamma(s_d, C_{9,1}) &= s_e, & s_d(x_7) &= \text{off} \wedge s_d(x_9) \neq \neg \text{works} \wedge s_e(x_9) = \neg \text{works} \\
 \gamma(s_f, C_{2,1}) &= s_g, & s_f(x_2) &\neq \text{low} \wedge s_g(x_2) = \text{low} \wedge s_g(x_5) \neq \text{battery_zero} \\
 \gamma(s_g, C_{5,1}) &= s_h, & s_g(x_2) &= \text{low} \wedge s_g(x_5) \neq \text{battery_zero} \wedge \\
 & & s_h(x_5) &= \text{battery_zero} \wedge s_h(x_7) \neq \text{off} \\
 \gamma(s_h, C_{7,1}) &= s_d, & s_h(x_5) &= \text{battery_zero} \wedge s_h(x_7) \neq \text{off} \wedge \\
 & & s_d(x_7) &= \text{off} \wedge s_d(x_9) \neq \neg \text{works} \\
 \gamma(s_l, C_{3,1}) &= s_m, & s_l(x_3) &\neq \text{empty} \wedge s_m(x_3) = \text{empty} \wedge s_m(x_6) \neq \text{gas_zero} \\
 \gamma(s_m, C_{6,1}) &= s_n, & s_m(x_3) &= \text{empty} \wedge s_m(x_6) \neq \text{gas_zero} \wedge \\
 & & s_n(x_6) &= \text{gas_zero} \wedge s_n(x_8) \neq \text{false} \\
 \gamma(s_n, C_{8,1}) &= s_o, & s_n(x_6) &= \text{gas_zero} \wedge s_n(x_8) \neq \text{false} \wedge \\
 & & s_o(x_8) &= \text{false} \wedge s_o(x_9) \neq \neg \text{works} \\
 \gamma(s_o, C_{9,1}) &= s_e, & s_o(x_8) &= \text{false} \wedge s_o(x_9) \neq \neg \text{works} \wedge s_e(x_9) = \neg \text{works}
 \end{aligned}$$

A.3. Definition of the process model $M(X(t)) = \langle PM(X(t)), SM(t), FM(t), BM(t) \rangle$

$PM(X(t)) = \langle X, \Psi, R_q \rangle$ where

$$\begin{aligned}
 X &= \{\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_7, \bar{x}_8, \bar{x}_9\}, \Delta_{\bar{x}_i} = \{0, 1\}, i = 1, 2, 3, 7, 8, 9 \\
 \Psi &= \{\Psi_i\}_{i=1,2,3,7,8,9} \text{ (threshold values of the time functions which we ignore)} \\
 R_q &= R_{goal} \cup R_n \cup R_{ab} \text{ such that} \\
 R_{goal} &\text{ describes the process operating goals } \bar{x}_9 = 1, \\
 R_n &\text{ describes the conditions of the normal operating mode :} \\
 &(\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_3 = 1 \wedge \bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 1) \vee \\
 &(((\bar{x}_1 = 0 \vee \bar{x}_2 = 0) \wedge \bar{x}_7 = 0) \vee (\bar{x}_3 = 0 \wedge \bar{x}_8 = 0)) \wedge \bar{x}_9 = 0 \\
 R_{ab} &\text{ describes the conditions of the abnormal operating mode :} \\
 &(\bar{x}_1 = 1 \wedge \bar{x}_2 = 1 \wedge \bar{x}_7 = 0) \vee \\
 &(\bar{x}_3 = 1 \wedge \bar{x}_8 = 0) \vee \\
 &(\bar{x}_7 = 1 \wedge \bar{x}_8 = 1 \wedge \bar{x}_9 = 0)
 \end{aligned} \tag{A.5}$$

$SM(X(t)) = \langle COMPS, R_{port}, R_{xport} \rangle$ where

$$\begin{aligned}
 COMPS &= \{c_{B1}, c_{B2}, c_{B3}, c_{B7}, c_{B8}, c_{B9}\} \\
 R_{port} &= \{out(c_{B1}) = in_1(c_{B7}), out(c_{B2}) = in_2(c_{B7}), out(c_{B7}) = in_1(c_{B9}), \\
 &\quad out(c_{B3}) = in_1(c_{B8}), out(c_{B3}) = in_2(c_{B8}), out(c_{B8}) = in_2(c_{B9})\} \\
 R_{xport} &= \{out(c_{B1}) = \bar{x}_1, out(c_{B2}) = \bar{x}_2, out(c_{B3}) = \bar{x}_3, \\
 &\quad out(c_{B7}) = \bar{x}_7, out(c_{B8}) = \bar{x}_8, out(c_{B9}) = \bar{x}_9\}
 \end{aligned} \tag{A.6}$$

$FM(X(t)) = \langle \Delta, F, R_f \rangle$ where

$$\begin{aligned}
 \Delta &= \Delta_{\bar{x}_1} \cup \Delta_{\bar{x}_2} \cup \Delta_{\bar{x}_3} \cup \Delta_{\bar{x}_7} \cup \Delta_{\bar{x}_8} \cup \Delta_{\bar{x}_9} \text{ with} \\
 \Delta_{\bar{x}_i} &= \{0, 1\}, i = 1, 2, 3, 7, 8, 9 \\
 F &= \{f_{B4}, f_{B5}, f_{B6}\} \text{ with} \\
 f_{B4} &: \Delta_{\bar{x}_1} \times \Delta_{\bar{x}_2} \rightarrow \Delta_{\bar{x}_7}, \\
 f_{B5} &: \Delta_{\bar{x}_3} \rightarrow \Delta_{\bar{x}_8}, \\
 f_{B6} &: \Delta_{\bar{x}_7} \times \Delta_{\bar{x}_8} \rightarrow \Delta_{\bar{x}_9}, \text{ and such that} \\
 f_{B4}(y_1, y_2) &= \text{and}(y_1, y_2) \\
 f_{B5}(y) &= \text{and}(y, y) \\
 f_{B6}(y_1, y_2) &= \text{and}(y_1, y_2) \\
 R_f &= \{\bar{x}_7 = f_{B4}(\bar{x}_1, \bar{x}_2), \bar{x}_8 = f_{B5}(\bar{x}_3), \bar{x}_9 = f_{B6}(\bar{x}_7, \bar{x}_8)\}
 \end{aligned} \tag{A.7}$$

$BM(X(t)) = \langle S, C, \gamma \rangle$ where

$S = \{s_8, s_{11}, s_{17}, s_{18}, s_{20}, s_{21}, s_{23}, s_{24}, s_{27}, s_{28}, s_{29}, s_{31}, s_{32}, s_{50}, s_{53}, s_{56}, s_{61}, s_{62}, s_{63}, s_{64}\}$ such that

$S = \{s : VAR \rightarrow VALUE \mid$ $s(x) = \delta, x \in X \subseteq VAR,$ $\delta \in \Delta \subseteq VALUE\}$						
S	\bar{x}_1	\bar{x}_2	\bar{x}_3	\bar{x}_7	\bar{x}_8	\bar{x}_9
s_8	1	0	1	0	0	0
s_{11}	0	1	1	0	0	0
s_{17}	1	1	1	0	0	0
s_{18}	1	1	0	1	0	0
s_{20}	1	0	1	1	0	0
s_{21}	1	0	1	0	1	0
s_{23}	0	1	1	1	0	0
s_{24}	0	1	1	0	1	0
s_{27}	1	1	1	1	0	0
s_{28}	1	1	1	0	1	0
s_{29}	1	1	0	1	1	0

$S = \{s : VAR \rightarrow VALUE \mid$ $s(x) = \delta, x \in X \subseteq VAR,$ $\delta \in \Delta \subseteq VALUE\}$						
S	\bar{x}_1	\bar{x}_2	\bar{x}_3	\bar{x}_7	\bar{x}_8	\bar{x}_9
s_{30}	1	0	1	1	1	0
s_{31}	0	1	1	1	1	0
s_{32}	1	1	1	1	1	0
s_{50}	1	1	0	1	0	1
s_{53}	1	0	1	0	1	1
s_{56}	0	1	1	0	1	1
s_{61}	1	1	0	1	1	1
s_{62}	1	0	1	1	1	1
s_{63}	0	1	1	1	1	1
s_{64}	1	1	1	1	1	1

(A.8)

$C = \{C_{1,1}, C_{1,2}, C_{2,1}, C_{2,2}, C_{3,1}, C_{3,2}, C_{7,1}, C_{7,2}, C_{8,1}, C_{8,2}, C_{9,1}, C_{9,2}\}$ where

$$\begin{aligned} C_{1,1} &= \{(\bar{x}_1, 0)\}, & C_{3,1} &= \{(\bar{x}_3, 0)\}, & C_{8,1} &= \{(\bar{x}_8, 0)\}, \\ C_{1,2} &= \{(\bar{x}_1, 1)\}, & C_{3,2} &= \{(\bar{x}_3, 1)\}, & C_{8,2} &= \{(\bar{x}_8, 1)\}, \\ C_{2,1} &= \{(\bar{x}_2, 0)\}, & C_{7,1} &= \{(\bar{x}_7, 0)\}, & C_{9,1} &= \{(\bar{x}_9, 0)\}, \\ C_{2,2} &= \{(\bar{x}_2, 1)\}, & C_{7,2} &= \{(\bar{x}_7, 1)\}, & C_{9,2} &= \{(\bar{x}_9, 1)\} \end{aligned}$$

$\gamma: S \times C \rightarrow S$ such that

$$\begin{aligned} \gamma(s_8, C_{2,1}) &= s_{17}, & \gamma(s_{31}, C_{1,1}) &= s_{17}, & \gamma(s_{32}, C_{1,1}) &= s_{31}, \\ \gamma(s_{17}, C_{7,2}) &= s_{27}, & \gamma(s_{32}, C_{9,2}) &= s_{64}, & \gamma(s_{18}, C_{3,2}) &= s_{27}, & \gamma(s_{32}, C_{3,1}) &= s_{29}, \\ \gamma(s_{20}, C_{7,1}) &= s_8, & \gamma(s_{32}, C_{2,1}) &= s_{30}, & \gamma(s_{21}, C_{2,2}) &= s_{28}, & \gamma(s_{50}, C_{9,1}) &= s_{18}, \\ \gamma(s_{23}, C_{7,1}) &= s_{11}, & \gamma(s_{53}, C_{9,1}) &= s_{21}, & \gamma(s_{24}, C_{1,2}) &= s_{28}, & \gamma(s_{56}, C_{9,1}) &= s_{24}, \\ \gamma(s_{27}, C_{1,1}) &= s_{23}, & \gamma(s_{61}, C_{8,1}) &= s_{50}, & \gamma(s_{27}, C_{2,1}) &= s_{20}, & \gamma(s_{62}, C_{7,1}) &= s_{53}, \\ \gamma(s_{27}, C_{8,2}) &= s_{32}, & \gamma(s_{63}, C_{7,1}) &= s_{56}, & \gamma(s_{28}, C_{7,2}) &= s_{32}, & \gamma(s_{64}, C_{1,1}) &= s_{63}, \\ \gamma(s_{29}, C_{8,1}) &= s_{18}, & \gamma(s_{64}, C_{2,1}) &= s_{62}, & \gamma(s_{30}, C_{7,1}) &= s_{21}, & \gamma(s_{64}, C_{3,1}) &= s_{61}, \end{aligned}$$

A.4. Definition of Boolean functions

$$\begin{aligned} \Delta_{x_1} &= \{\text{blown}, \phi_1\} & \Delta_{\bar{x}_1} &= \{0, 1\} \\ \Delta_{x_2} &= \{\text{low}, \phi_2\} & \Delta_{\bar{x}_2} &= \{0, 1\} \\ \Delta_{x_3} &= \{\text{empty}, \phi_3\} & \Delta_{\bar{x}_3} &= \{0, 1\} \\ \Delta_{x_4} &= \{\text{broken}, \phi_4\} & \Delta_{\bar{x}_4} &= \{0, 1\} \\ \Delta_{x_5} &= \{\text{battery_zero}, \phi_5\} & \Delta_{\bar{x}_5} &= \{0, 1\} \\ \Delta_{x_6} &= \{\text{gas_zero}, \phi_6\} & \Delta_{\bar{x}_6} &= \{0, 1\} \\ \Delta_{x_7} &= \{\text{off}, \phi_7\} & \Delta_{\bar{x}_7} &= \{0, 1\} \\ \Delta_{x_8} &= \{\text{false}, \phi_8\} & \Delta_{\bar{x}_8} &= \{0, 1\} \\ \Delta_{x_9} &= \{\neg \text{works}, \phi_9\} & \Delta_{\bar{x}_9} &= \{0, 1\} \end{aligned}$$

(A.9)

$f_{B1} : \Delta_{\bar{x}_3} \rightarrow \Delta_{\bar{x}_6}$		
y	$f_{B1}(y)$	
0	0	
1	1	

$f_{B2} : \Delta_{\bar{x}_2} \rightarrow \Delta_{\bar{x}_5}$		
y	$f_{B2}(y)$	
0	0	
1	1	

$f_{B3} : \Delta_{\bar{x}_1} \rightarrow \Delta_{\bar{x}_4}$		
y	$f_{B3}(y)$	
0	0	
1	1	

$f_{B4} : \Delta_{\bar{x}_1} \times \Delta_{\bar{x}_2} \rightarrow \Delta_{\bar{x}_7}$			
y_1	y_2	$f_{B4}(y_1, y_2)$	
0	0	0	
1	0	0	
0	1	0	
1	1	1	

$f_{B5} : \Delta_{\bar{x}_3} \rightarrow \Delta_{\bar{x}_8}$		
y	$f_{B5}(y)$	
0	0	
1	1	

$f_{B6} : \Delta_{\bar{x}_7} \times \Delta_{\bar{x}_8} \rightarrow \Delta_{\bar{x}_9}$			
y_1	y_2	$f_{B6}(y_1, y_2)$	
0	0	0	
0	1	0	
1	0	0	
1	1	1	

(A.10)

$$\begin{aligned} f_{B1}(y) &= \text{and}(y, y) & f_{B3}(y) &= \text{and}(y, y) & f_{B5}(y) &= \text{and}(y, y) \\ f_{B2}(y) &= \text{and}(y, y) & f_{B4}(y_1, y_2) &= \text{and}(y_1, y_2) & f_{B6}(y_1, y_2) &= \text{and}(y_1, y_2) \end{aligned}$$

(A.11)

A.5 Specification of structural model of logical gates

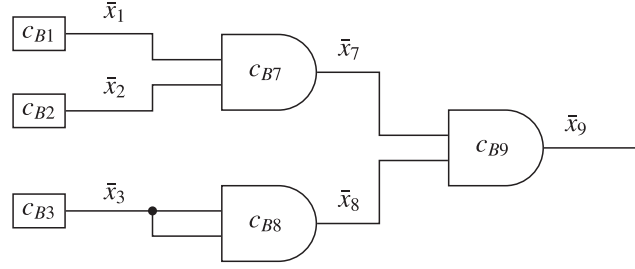


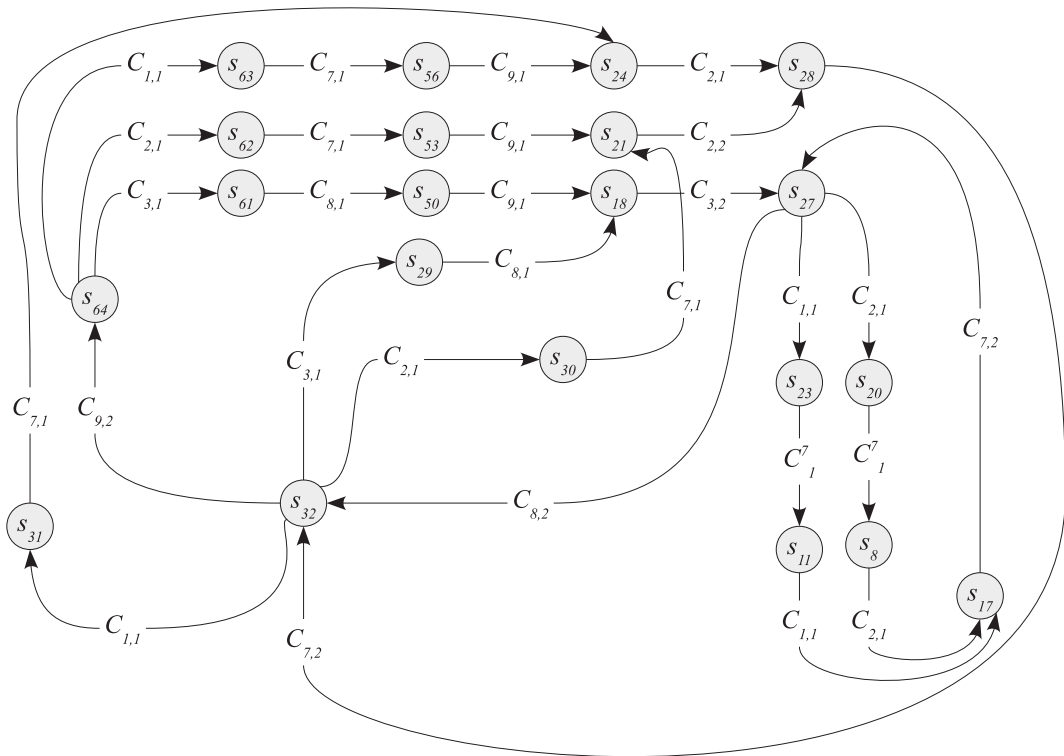
Fig. A.26. Logical model of the process.

$$\begin{aligned}
 & \text{Fuse}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = 1 \\
 & \text{Battery}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = 1 \\
 & \text{FuelTank}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = 1 \\
 & \text{GasSupply}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = \text{and}(\text{in}_1(Y), \text{in}_2(Y)) \\
 & \text{ElectricSupply}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = \text{and}(\text{in}_1(Y), \text{in}_2(Y)) \\
 & \text{Engine}(Y) \wedge \neg AB(Y) \Rightarrow \text{out}(Y) = \text{and}(\text{in}_1(Y), \text{in}_2(Y))
 \end{aligned} \tag{A.12}$$

$$\begin{aligned}
 & \text{Fuse}(c_{B1}), \quad \text{GasSupply}(c_{B7}) \\
 & \text{Battery}(c_{B2}), \quad \text{ElectricSupply}(c_{B8}) \\
 & \text{FuelTank}(c_{B3}), \quad \text{Engine}(c_{B9})
 \end{aligned} \tag{A.13}$$

$$\begin{aligned}
 & \text{out}(c_{B1}) = \text{in}_1(c_{B7}) \quad \text{out}(c_{B3}) = \text{in}_1(c_{B8}) \quad \text{out}(c_{B7}) = \text{in}_1(c_{B9}) \\
 & \text{out}(c_{B2}) = \text{in}_1(c_{B7}) \quad \text{out}(c_{B3}) = \text{in}_2(c_{B8}) \quad \text{out}(c_{B8}) = \text{in}_1(c_{B9})
 \end{aligned} \tag{A.14}$$

A.6. Figures and tables

Fig. A.27. Behavioural model of $X(t)$.

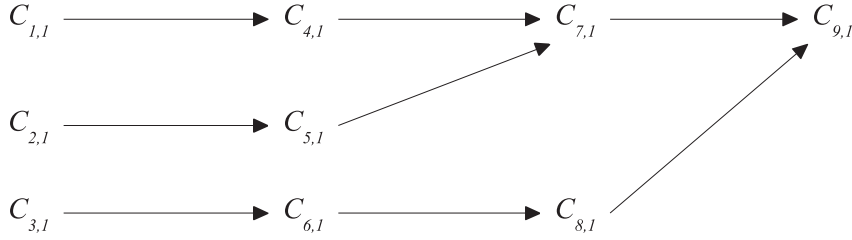
Fig. A.28. Sequences of observation classes defined from Ω .

Table A.5

Matrix E.

ij	$C_{1,1}$	$C_{1,2}$	$C_{2,1}$	$C_{2,2}$	$C_{3,1}$	$C_{3,2}$	$C_{7,1}$	$C_{7,2}$	$C_{8,1}$	$C_{8,2}$	$C_{9,1}$	$C_{9,2}$
$C_{1,1}$							1					
$C_{1,2}$				1		1		1				
$C_{2,1}$							1					
$C_{2,2}$		1				1		1				
$C_{3,1}$									1			
$C_{3,2}$	1	1	1	1				1		1		
$C_{7,1}$		1		1							1	
$C_{7,2}$	1		1		1					1		1
$C_{8,1}$											1	
$C_{8,2}$								1				1
$C_{9,1}$		1		1		1						
$C_{9,2}$	1		1		1							

Table A.6

Logical and physical interpretations.

Knowledge	Logical interpretation	Physical interpretation	x_i^{p1}
$x_1 = \text{blown}$	$\bar{x}_1 = 0$	$R(t) = \infty$	$(x_1^p = \infty)$
$x_1 = \text{blown}$	$\bar{x}_1 = 1$	$R(t) = c_r$	$(x_1^p = c_r)$
$x_2 = \text{low}$	$\bar{x}_2 = 0$	$Q(t) = 0$	$(x_2^p = 0)$
$x_2 = \text{low}$	$\bar{x}_2 = 1$	$Q(t) \neq 0$	$(x_2^p \neq 0)$
$x_3 = \text{empty}$	$\bar{x}_3 = 0$	$V(t) = 0$	$(x_3^p = 0)$
$x_3 = \text{empty}$	$\bar{x}_3 = 1$	$V(t) \neq 0$	$(x_3^p \neq 0)$
$x_7 = \text{off}$	$\bar{x}_7 = 0$	$U(t) = 0$	$(x_7^p = 0)$
$x_7 = \text{off}$	$\bar{x}_7 = 1$	$U(t) \neq 0$	$(x_7^p \neq 0)$
$x_8 = \text{false}$	$\bar{x}_8 = 0$	$Qv(t) = 0$	$(x_8^p = 0)$
$x_8 = \text{false}$	$\bar{x}_8 = 1$	$Qv(t) \neq 0$	$(x_8^p \neq 0)$
$x_9 = \text{works}$	$\bar{x}_9 = 0$	$\alpha \cdot U(t) \cdot Qv(t) = 0^2$	$(x_9^p = 0)$
$x_9 = \text{works}$	$\bar{x}_9 = 1$	$\alpha \cdot U(t) \cdot Qv(t) \neq 0$	$(x_9^p \neq 0)$

¹ $\alpha \in \{0, 1\}$ models the car's key (off/on) allowing to interpret $x_9 = \text{works}$ as the car is stopped (owing to the fact that it is off, there is no voltage or there is no gas). However, we do not have information about α , so we assume that it cannot be observed.

² $R(t)$, $Q(t)$, $V(t)$, $U(t)$ and $Qv(t)$ are also noted as x_1^p , x_2^p , x_3^p , x_7^p and x_8^p , respectively.

References

- [1] J. Breuker, W. Van de Velde, CommonKADS Library for Expertise Modelling, IOS Press, 1994.
- [2] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga, Knowledge Engineering and Management: The CommonKADS Methodology, MIT Press, 2000.
- [3] J.H. Gennari, M.A. Musen, R.W. Ferguson, W.E. Grosso, M. Crubzy, H. Erickson, N.F. Noy, S.W. Tu, The evolution of protégé: an environment for knowledge-based systems development, Int. J. Hum. Comput. Stud. 58 (2003) 89–123.
- [4] J. Angele, D. Fensel, R. Studer, Domain and task modeling in MIKE, Proceedings of the IFIP WG8.1/13.2 Joint Working Conference on Domain Knowledge for Interactive System Design, 1996, pp. 8–10.
- [5] J. Angele, D. Fensel, D. Landes, R. Studer, Developing knowledge based-systems with MIKE, Autom. Softw. Eng. 5 (4) (1998) 389–418.
- [6] O. Cairó, J.C. Alvarez, The KAMET II methodology: a modern approach for building diagnosis-specialized knowledge-based systems, ISMIS, Lecture Notes in Computer Science, vol. 2871, Springer, 2003, pp. 652–656.

- [7] O. Cairó, J.C. Alvarez, KAMET II: an extended knowledge-acquisition methodology, in: V. Palade, R.J. Howlett, L. Jain (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 2773, Springer, 2003, pp. 61–67.
- [8] E. Motta, A. Stutt, K. O'Hara, J. Kuusela, H. Toivonen, H. Reichgelt, S. Watt, S. Aitken, F. Verbeck, VITAL knowledge representation language specification, Tech. Rep. 81 (V 5), Human Cognition Research Laboratory of The Open University, 1991.
- [9] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., 1993.
- [10] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE* 77 (2) (1989) 257–286.
- [11] R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Machine learning: an artificial intelligence approach, No. 1 in *Machine Learning: An Artificial Intelligence Approach*, Tioga, 1983.
- [12] J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning Bayesian networks from data: an information-theory based approach, *Artif. Intell.* 137 (1–2) (2002) 43–90.
- [13] D. Defays, An efficient algorithm for a complete link method, *Comput. J.* 20 (4) (1977) 364–366.
- [14] T. Mitchell, *Machine Learning*, McGraw Hill, 1977.
- [15] L. Chittaro, R. Ranon, Augmenting the diagnostic power of flow-based approaches to functional reasoning, *AAAI-96 Proceedings*, 1996, pp. 1010–1015.
- [16] L. Chittaro, G. Guida, C. Tasso, E. Toppino, Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: a case study in diagnosis, *IEEE Trans. Syst. Man Cybern.* 23 (6) (1993) 1718–1751.
- [17] R.C. Rosenberg, D.C. Karnopp, *Introduction to Physical System Dynamics*, McGraw-Hill, 1983.
- [18] A. Newell, The knowledge level, *AI Mag.* 2 (2) (1981) 1–20.
- [19] L. Floridi, The method of levels of abstraction, *Mind. Mach.* 18 (2008) 303–329.
- [20] L. Floridi, Levels of abstraction and the Turing test, *Keybenetes* 39 (3) (2010) 423–440.
- [21] N. Wickramasinghe, *Encyclopedia of knowledge management*, Ch. Knowledge Creation, Idea Group Inc., 2006. 326–335.
- [22] N. Glaser, The CoMoMAS methodology and environment for multi-agent system development, in: C. Zhang, D. Lukose (Eds.), *Multi-Agent Systems Methodologies and Applications*, Lecture Notes in Computer Science, vol. 1286, Springer, Berlin Heidelberg, 1997, pp. 1–16.
- [23] C.A. Iglesias, M. Garijo, J.C. González, J.R. Velasco, A Methodological Proposal for Multiagent Systems Development Extending CommonKADS, 1996.
- [24] S.S. Hasan, R.K. Isaac, An integrated approach of MAS-CommonKADS, model-view-controller and web application optimization strategies for web-based expert system development, *Expert Syst. Appl.* 38 (1) (2011) 417–428.
- [25] X. Luo, C. Miao, N.R. Jennings, M. He, Z. Shen, M. Zhang, KEMNAD: a knowledge engineering methodology for negotiating agent development, *Comput. Intell.* 28 (1) (2012) 51–105.
- [26] N. Prat, J. Akoka, I. Comyn-Wattiau, An {MDA} approach to knowledge engineering, *Expert Syst. Appl.* 39 (12) (2012) 10420–10437.
- [27] D. Sutton, V. Patkar, CommonKADS analysis and description of a knowledge based system for the assessment of breast cancer, *Expert Syst. Appl.* 36 (2, Part 1) (2009) 2411–2423.
- [28] J. Sigut, J. Piñeiro, E. González, J. Torres, An expert system for supervised classifier design: application to Alzheimer diagnosis, *Expert Syst. Appl.* 32 (3) (2007) 927–938.
- [29] A. Fernández-Leal, V. Moret-Bonillo, E. Mosqueira-Rey, Causal temporal constraint networks for representing temporal knowledge, *Expert Syst. Appl.* 36 (1) (2009) 27–42.
- [30] G. Nalepa, A. Lig za, The HeKatE methodology. Hybrid engineering of intelligent systems, *Int. J. Appl. Math. Comput. Sci.* 20 (1) (2010) 35–53.
- [31] M. Le Goc, Notion d'observation pour le diagnostic des processus dynamiques: Application à SACHEM et à la découverte de connaissances temporelles, *Habilitation à Diriger des Recherches*, Université de Droit d'Economie et des Sciences d'Aix-Marseille, 2006.
- [32] C.E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.* 27 (1948) 379–423 (623–656).
- [33] R. Reiter, A theory of diagnosis from first principles, *Artif. Intell.* 32 (1) (1987) 57–95.
- [34] P. Dagues, *Diagnostic, Intelligence Artificielle, et Reconnaissance des Formes*, Ch. Théorie logique du diagnostic à base de modèles, Hermes Science Publications, 2001. 17–105.
- [35] M. Le Goc, SACHEM, a real-time intelligent diagnosis system based on the discrete event paradigm, *Simulation* 80 (11) (2004) 591–617.
- [36] M. Le Goc, A. Ahdab, *Learning Bayesian Networks from Timed Observations*, LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012.
- [37] N. Benayadi, M. Le Goc, Mining timed sequences with TOM4L framework, *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010)*, vol. 2, 2010, pp. 111–120.
- [38] A. Ahdab, M. Le Goc, Learning dynamic Bayesian networks with the TOM4L process, *Proceedings of the Fifth International Conference on Software and Data Technologies (ICSoft 2010)*, vol. 2, 2010, pp. 353–363.
- [39] P. Bouché, M. Le Goc, J. Coinu, A global model of sequences of discrete event class occurrences, *Proceedings of the Tenth International Conference on Enterprise Information Systems (ICEIS 2008)*, vol. AIDSS, 2008, pp. 173–180.
- [40] M. Polanyi, *The Tacit Dimension*, Doubleday & Company, Inc., 1966.
- [41] I. Nonaka, N. Konno, The concept of “Ba” building a foundation for knowledge creation, *Calif. Manag. Rev.* 40 (3) (1998) 40–54.
- [42] R. Studer, R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, *Data Knowl. Eng.* 25 (1–2) (1998) 161–198.
- [43] I. Nonaka, Dynamic theory of organizational knowledge creation, *Organ. Sci.* 5 (1) (1994) 14–37.
- [44] I. Nonaka, The knowledge-creating company, *Harv. Bus. Rev.* (1991) 96–104.
- [45] M. Alavi, D.E. Leidner, Review: knowledge management and knowledge management systems: conceptual foundations and research issues, *MIS Q.* 25 (1) (2001) 107–136.
- [46] A.R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*, Penguin Books, New York, 2005.
- [47] A.R. Damasio, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Inc., New York, 2000.
- [48] L. Pomponio, M. Le Goc, Timed observations modelling for diagnosis methodology: a case study, *Proceedings of the 5th International Conference on Software and Data Technologies (ICSoft 2010)*, vol. 2, 2010, pp. 504–507.
- [49] M. Le Goc, E. Masse, C. Curt, Modeling processes from timed observations, *Proceedings of the 3rd International Conference on Software and Data Technologies (ICSoft'08)*, 2008, pp. 249–256.
- [50] M. Le Goc, E. Masse, Towards a multimodeling approach of dynamic systems for diagnosis, *Proceedings of the 2nd International Conference on Software and Data Technologies (ICSoft'07)*, 2007, pp. 277–282.
- [51] L. Chittaro, R. Ranon, Diagnosis of multiple faults with flow-based functional models: the functional diagnosis with efforts and flows approach, *Reliab. Eng. Syst. Saf.* 64 (2) (1999) 137–150.
- [52] C. Zanni, M. Le Goc, C. Frydman, A conceptual framework for the analysis, classification and choice of knowledge-based diagnosis systems, *KES, Int. J. Knowl. Based Intell. Eng. Syst.* 10 (2) (2006) 113–138.
- [53] N. Benayadi, Contribution à la découverte de connaissances à partir de données datées, (Ph.D. thesis) Université Paul Cézanne Aix-Marseille III, 2010.
- [54] L. Pomponio, M. Le Goc, Integrating knowledge engineering with knowledge discovery in database: TOM4D and TOM4L, in: F. Colette, L. C. J. (Eds.), *Innovations in Intelligent Machines-4 – Recent Advances in Knowledge Engineering*, Studies in Computational Intelligence, vol. 514, Springer, 2014, pp. 189–231.
- [55] E. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press Cambridge, 2003.
- [56] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Stat.* 22 (1) (1951) 79–86.
- [57] M. Le Goc, M. Gaeta, Modeling structures in generic space, a condition for adaptiveness of monitoring cognitive agent, *J. Intell. Robot. Syst.* 41 (2–3) (2004) 113–140.
- [58] L. Pomponio, M. Le Goc, A. Anfosso, E. Pascual, Levels of abstraction for behavior modeling in the GerHome project, *Int. J. E-Health Med. Commun.* 3 (3) (2012) 12–28.
- [59] L. Pomponio, M. Le Goc, E. Pascual, A. Anfosso, Discovering models of human's behavior from sensor's data, *Workshop Proceedings of the 7th International Conference on Intelligent Environments*, Nottingham, UK. 25–26th of July 2011, *Ambient Intelligence and Smart Environments*, vol. 10, IOS Press, 2011, pp. 17–28.
- [60] M. Le Goc, I. Fakhfakh, C. Curt, L. Torres, Diagnosis of the hydraulic dam safety based on multi-modelling approach and the timed observations theory, *Risk and Cognition*, Springer-Verlag, Germany, 2014. (forthcoming).

- [61] I. Fakhfakh, M. Le Goc, L. Torres, C. Curt, Modeling and diagnosis processes from timed observations, Proceedings of the 23rd International Workshop on Principles of Diagnosis (DX-2012), Great Malvern, United Kingdom, 2012.
- [62] I. Fakhfakh, M. Le Goc, L. Torres, C. Curt, Modeling and diagnosis dynamic system from timed observation, in: 20th European Conference on Artificial Intelligence (Ed.), Proceedings of Diagnostic REASONing: Model Analysis and Performance (DREAMAP), Montpellier, France, 2012, pp. 5–10.
- [63] I. Fakhfakh, M. Le Goc, L. Torres, C. Curt, Towards a decompositional and incremental approach of diagnosis dynamic system from timed observations, Proceedings of the 23rd International Workshop on Principles of Diagnosis (DX-2012), Great Malvern, United Kingdom, 2012.
- [64] M. Le Goc, Ontological models as shared model to validate a knowledge-based system, KAW'99, Twelfth Workshop on Knowledge Acquisition, Modeling and Management, Voyager Inn, Banff, Alberta, Canada, 1999, pp. 16–21.
- [65] M. Le Goc, C. Frydman, L. Torres, Verification and validation of the Sachem conceptual model, Int. J. Hum. Comput. Stud. 56 (2) (2002) 199–223.



Laura Pomponio received a PhD degree in Computer Science from Aix-Marseille University (France) in 2012; by carrying out research activities in France, in the *Laboratoire des Sciences de l'Information et Systèmes* (LSIS) and in the *Centre Scientifique et Technique du Bâtiment* (CSTB). She received an MSc degree in Computer Science from the National University of Rosario, Argentina, in 2009. In previous years, she also carried out research and technological development activities in different institutions of Argentina and Portugal. Currently, she is part of the research group of Software Engineering of CIFASIS-CONICET, Argentina.



Marc Le Goc is a full Professor at Aix-Marseille University (France) and a Researcher in Artificial Intelligence, Machine Learning, and Knowledge Engineering at LSIS, the Laboratory for Sciences of Information and Systems at Marseille (UMR CNRS 7296, France). He developed the Timed Observations Theory (TOT, 2006), a mathematical framework that integrates and extends Shannon's Theory of Communication, Poisson's and Markov's Theories, and the Logical Theory of Model Based Diagnosis. TOT is the base of both a Knowledge Engineering Methodology (Tom4D, Timed Observation Modelling For Diagnosis) and a Timed Data Mining process (Tom4L, Timed Observation Mining for Learning). He is currently extending the TOT framework to financial analysis and brain modelling.